

# **Industry Project Report On Security Operation Center (SOC)**

**Developed By: -**

Vraj Patel (20162171034)

**Guided By:-**

Prof. Sonam Singh (Internal)  
Janvi Sharma (External –  
Heritage Cyberworld)

**Submitted to  
Faculty of Engineering and Technology  
Institute of Computer Technology  
Ganpat University**



**Ganpat  
University**

॥ विद्या समाजोत्कर्षः ॥

**Institute of  
Computer  
Technology**



**Year – 2024**



## CERTIFICATE

This is to certify that the **Industry** Project work entitled “**SOC**” by Vraj Patel (EnrolmentNo. 20162171034) and of Ganpat University, towards the partial fulfillment of requirements of the degree of Bachelor of Technology – Computer Science and Engineering, carried out by them in the CSE(CS) Department at Heritage Cyberworld LLP The results/findings contained in this Project have not been submitted in part or full to any other University / Institute for award of any other Degree/Diploma.

Name & Signature of Internal Guide

Name & Signature of Head

**Place: ICT - GUNI**

**Date**

**Registered Office**  
504-Santorini Square,  
Opp. Star Bazaar, Jodhpur,  
Ahmedabad-380015,  
Gujarat, INDIA.

**Corporate Office**  
504, 505 Aarohi Verve,  
Near Vakil Bridge, Bopal,  
Ahmedabad-380058,  
Gujarat, INDIA.

**Integrated Cyber Security Command Control Center**  
605-606 Solaris Business Hub,  
Opp. Parshwanath Jain Mandir BRTS,  
Bhuyangdev, Ahmedabad-380013,  
Gujarat, INDIA.



### INTERNSHIP COMPLETION CERTIFICATE

DATE: 27<sup>th</sup> April 2024

This is to certify that Vraj Patel has successfully completed a four month SOC analyst internship with Heritage Cyberworld.

During this internship, he has demonstrated exceptional dedication, enthusiasm, and a strong willingness to learn. Vraj Patel actively participated in gaining knowledge about SOC infrastructure, log analysis, implementation, and integration of different SIEM tools such as Wazuh and Splunk, as well as expertise in threat hunting, threat intelligence, and incident response also in Final Project deployed virtual SOC with AD created demo users to stimulate a working office and consistently contributed valuable insights to our team.

Date of Starting: 1<sup>st</sup> January 2024

Date of Completion: 30<sup>th</sup> April 2024

This internship has provided him with valuable hands-on experience in SIEM Tools, Analytical skills, problem solving skills, Knowledge of security tools and Incident response experience. enhancing his knowledge and skills in Cyber Security & IT Industry.

We commend Vraj Patel for his outstanding performance and commitment throughout the duration of the internship. We believe that he has a bright future ahead and wish him continued success in his academic and professional endeavour's.



*Vraj Patel*  
AUTHORIZED SIGNATURE

*Vraj Patel*

INTERN SIGNATURE

## **ACKNOWLEDGEMENT**

IBM/Industry Internship project is a golden opportunity for learning and self-development. I consider myself very lucky and honored to have so many wonderful people lead me through in completion of this project. First and foremost, I would like to thank Dr. Rohit Patel, Principal, ICT, and Prof. Dharmesh Darji , Head, ICT who gave us an opportunity to undertake this project. My grateful thanks to Prof. Sonam Singh & Mr. Anoj Dixit (Internal & External Guides) for their guidance in project work, who despite being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where we would have been without his/her help. CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

**VRAJ PATEL (Enrollment No:20162171034)**

## **ABSTRACT**

Security Operations Center (SOC) serves as a pivotal entity within modern cybersecurity infrastructures, tasked with the continuous monitoring, detection, analysis, and response to potential security threats and incidents. This abstract provides an overview of SOC's fundamental components, functions, and methodologies aimed at safeguarding organizational assets and data integrity. It explores the role of SOC in proactively identifying and mitigating security risks through advanced technologies such as intrusion detection systems (IDS), security information and event management (SIEM) platforms, threat intelligence, and incident response frameworks. Additionally, the abstract highlights the significance of collaboration between SOC analysts, threat hunters, and incident responders to ensure swift and effective remediation of security incidents. Emphasis is placed on the evolving landscape of cyber threats and the imperative for SOC to adapt and innovate in order to maintain resilience against sophisticated adversaries. Through constant refinement of processes, adoption of best practices, and integration of automation and artificial intelligence (AI), SOC endeavors to fortify organizational defenses and uphold the integrity of digital assets in an increasingly complex threat landscape.

## **INDEX**

<b>Title</b>	<b>Page No</b>
<b><u>CHAPTER 1: INTRODUCTION</u></b>	<b>01-02</b>
<b><u>CHAPTER 2: SCOPE</u></b>	<b>03-04</b>
<b><u>CHAPTER 3: HARDWARE REQUIREMENT</u></b>	<b>05-06</b>
<b><u>CHAPTER 4: SOFTWARE REQUIREMENT</u></b>	<b>07-08</b>
<b><u>CHAPTER 5: WEEKLY TASK</u></b>	<b>09-24</b>
<b>5.1 WEEK – 1</b>	10
<b>5.2 WEEK – 2</b>	12
<b>5.3 WEEK – 3</b>	13
<b>5.4 WEEK – 4</b>	16
<b>5.5 WEEK – 5</b>	19
<b>5.6 WEEK – 6</b>	23
<b><u>CHAPTER 6: WAZUH DEPLOYMENT</u></b>	<b>25-44</b>
<b><u>CHAPTER 7: VIRUSTOTAL INTEGRATION</u></b>	<b>45-50</b>
<b><u>CHAPTER 8: WAZUH POC</u></b>	<b>51-64</b>
<b><u>CHAPTER 9: WAZUH AND SOAR INTEGRATION</u></b>	<b>65-70</b>
<b><u>CHAPTER 10: WINDOWS AD CREATION</u></b>	<b>71-87</b>
<b><u>CHAPTER 11: SPLUNK INSTALLTION</u></b>	<b>88-99</b>
<b><u>CHAPTER 12: CONCLUSION</u></b>	<b>100-101</b>
<b><u>CHAPTER 13: REFRENCES</u></b>	<b>102-103</b>

## **CHAPTER: 1 INTRODUCTION**

## **CHAPTER 1 INTRODUCTION**

In an era marked by ubiquitous digital connectivity and pervasive cyber threats, the Security Operations Center (SOC) emerges as a cornerstone in the defense against malicious actors and cyber attacks. With organizations facing an ever-expanding array of sophisticated threats, ranging from malware and phishing scams to advanced persistent threats (APTs) and ransomware, the role of SOC has become increasingly vital in safeguarding sensitive data, preserving operational continuity, and mitigating financial and reputational risks.

The introduction of this discourse sets the stage for understanding the multifaceted role of SOC within the broader context of cybersecurity operations. It delineates the escalating nature of cyber threats and the imperative for organizations across industries to fortify their defenses and bolster their incident response capabilities. As technology evolves and threat vectors become more diverse and elusive, the SOC stands as a bastion of vigilance, equipped with the requisite tools, technologies, and expertise to detect, analyze, and mitigate security incidents in real-time.

This introduction provides a glimpse into the foundational principles and operational paradigms that underpin the SOC's mission, emphasizing the symbiotic relationship between proactive threat detection, rapid incident response, and continuous improvement. As organizations contend with an ever-shifting threat landscape characterized by novel attack vectors and sophisticated adversaries, the efficacy of the SOC hinges on its ability to adapt, innovate, and collaborate across organizational silos to confront emerging challenges head-on.

In the subsequent sections, we delve deeper into the core components, operational workflows, and technological innovations that define contemporary SOC operations. Through a comprehensive exploration of SOC's functions, methodologies, and best practices, this discourse seeks to illuminate the pivotal role that SOC plays in fortifying organizational resilience and safeguarding critical assets against the relentless tide of cyber threats.

## **CHAPTER: 2 PROJECT SCOPE**

## **CHAPTER 2 PROJECT SCOPE**

The scope of this discussion encompasses the multifaceted dimensions of Security Operations Center (SOC) operations, focusing on its primary functions, methodologies, technologies, and best practices within the realm of cybersecurity.

The scope also acknowledges the dynamic nature of the cybersecurity landscape and the evolving threat landscape, which necessitates ongoing adaptation, innovation, and knowledge sharing within SOC environments. While the focus is primarily on traditional SOC models, consideration may also be given to emerging trends such as managed SOC services, cloud-based SOC solutions, and the integration of artificial intelligence (AI) and machine learning (ML) technologies within SOC operations.

Overall, the scope aims to provide a comprehensive understanding of the role, functions, and operational dynamics of SOC within modern cybersecurity ecosystems, with a view towards fostering resilience, agility, and proactive threat mitigation strategies in the face of evolving cyber threats.

## **CHAPTER: 3 HARDWARE REQUIREMENTS**

## **CHAPTER: 3 HARDWARE REQUIREMENTS**

CPU: Multi-core processors (e.g., Intel Xeon, AMD EPYC) with sufficient processing power to handle concurrent tasks and data analysis.

RAM: Depending on the size of the organization and the volume of data being processed, servers may require anywhere from 32GB to 256GB or more of RAM.

Storage: High-speed SSDs or SAS drives for storing security logs, event data, and application databases. Storage capacity requirements will vary based on data retention policies and the volume of data generated.

Network Appliances:

Firewalls, routers, switches, and other network appliances typically have their own hardware specifications provided by the manufacturer. These specifications should be aligned with the organization's network throughput requirements and security policies.

Endpoint Devices:

Workstations or laptops used by SOC analysts should have modern CPUs (e.g., Intel Core i7 or equivalent), sufficient RAM (16GB or more), and fast SSD storage for responsiveness during analysis tasks.

## **CHAPTER: 4 SOFTWARE REQUIREMENTS**

## **CHAPTER: 4 SOFTWARE REQUIREMENTS**

Security Information and Event Management (SIEM) Platform:

Examples: Splunk Enterprise Security, IBM QRadar, LogRhythm, Elastic SIEM.

Description: SIEM platforms collect, analyze, and correlate security event data from various sources to provide real-time threat detection, incident response, and compliance reporting.

Intrusion Detection System/Intrusion Prevention System (IDS/IPS):

Examples: Snort, Suricata, Cisco Firepower, Palo Alto Networks IDS/IPS.

Description: IDS/IPS solutions monitor network traffic for signs of malicious activity or policy violations and can either alert SOC analysts or take automated action to block suspicious traffic.

Endpoint Detection and Response (EDR) Software:

Examples: CrowdStrike Falcon, Carbon Black, Microsoft Defender for Endpoint, Symantec Endpoint Detection and Response.

Description: EDR solutions provide visibility into endpoint activities, detect and respond to threats in real-time, and offer capabilities for investigation and remediation of security incidents on endpoints.

Threat Intelligence Platforms (TIPs):

Examples: ThreatConnect, Anomali ThreatStream, Recorded Future, ThreatQ

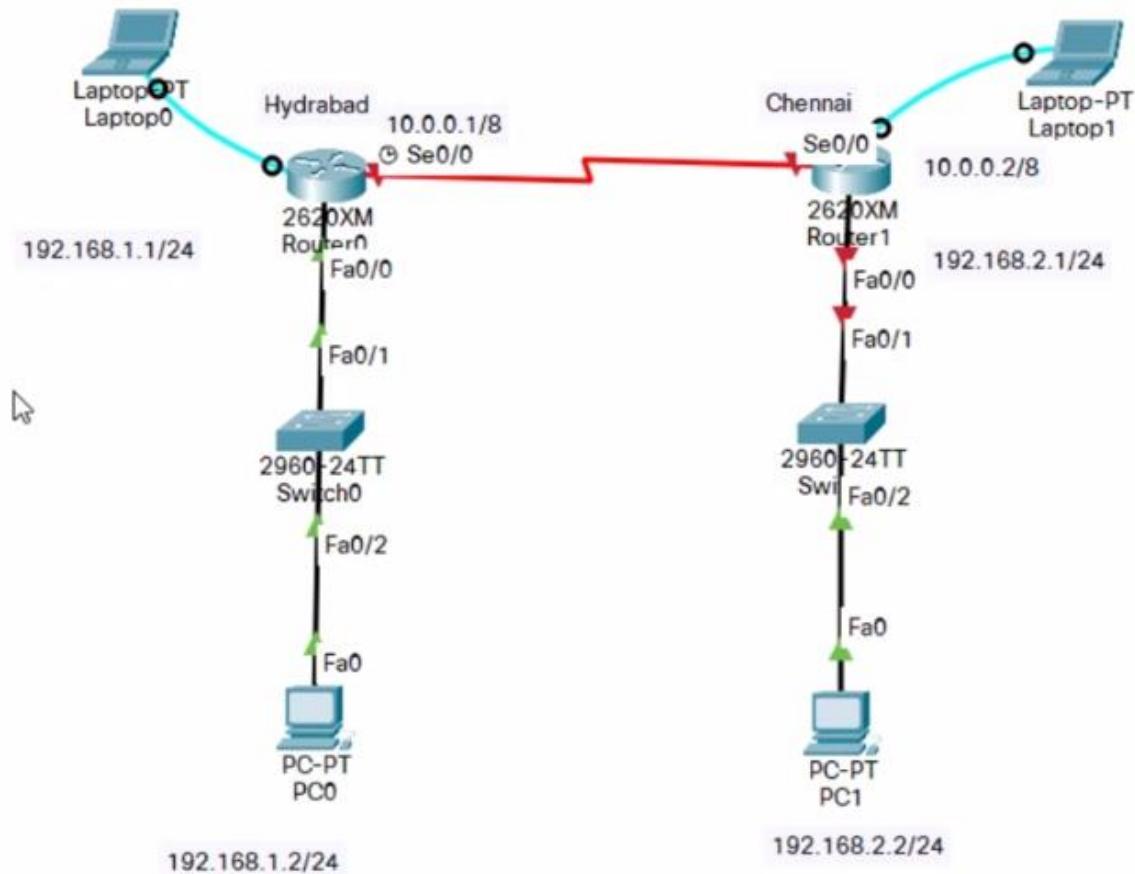
## **CHAPTER: 5 WEEKLY TASKS**

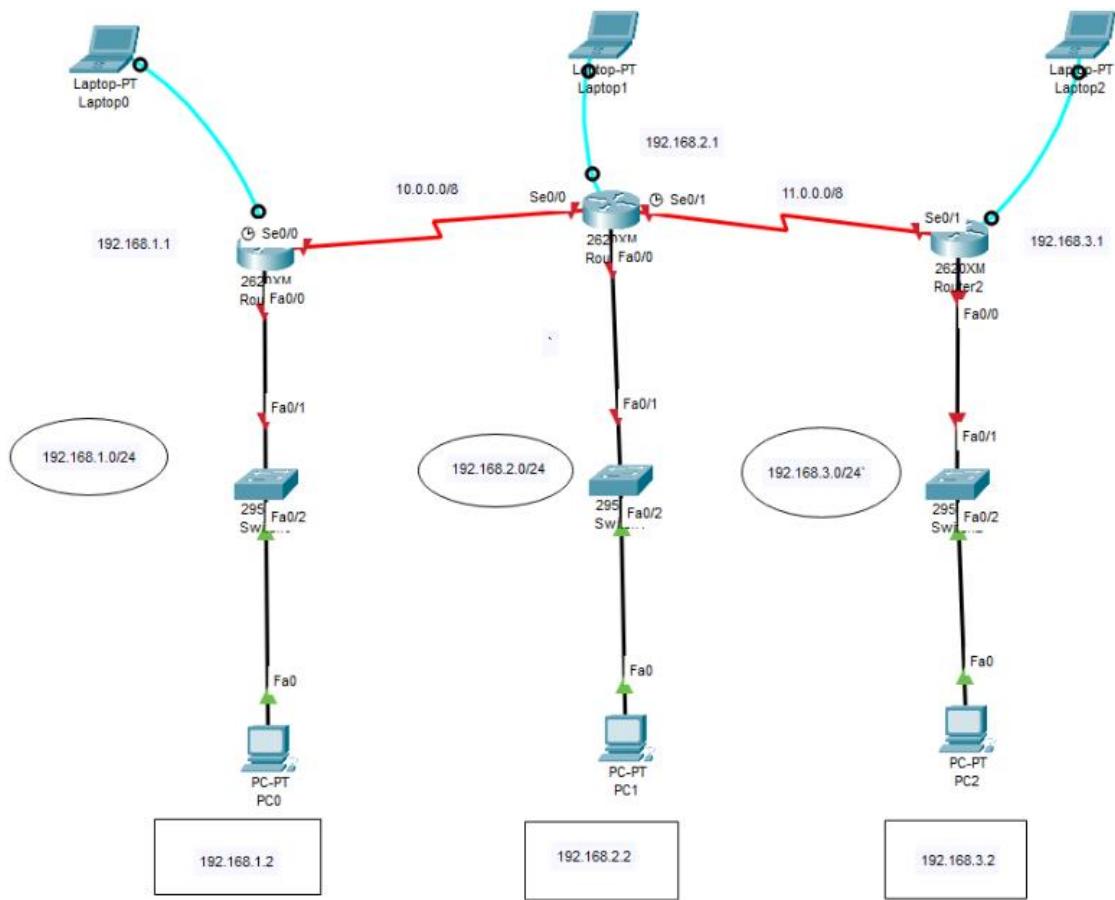
## CHAPTER: 5 WEEKLY TASKS

### 5.1 WEEK – 1

- Basics of Computer Networks
- Practical on Cisco Packet Tracer

Routing protocols –





- Practical on Wireshark

## 5.2 WEEK – 2

- Certified course of EC-Council



- Certified course of Splunk



## 5.3 WEEK – 3

- Practical on Splunk
- Analyzing Logs on Splunk

The screenshot shows the Splunk web interface with a search bar containing the query: sourcetype="firewall logs" | stats count by Src\_ip. The results table has the following columns: Src\_ip and count. The data shows approximately 82,974 events from various IP addresses, with the top entry being 84.252.143.78 with a count of 33,696.

Src_ip	count
84.252.143.78	33696
178.159.37.10	21546
198.51.100.75	3082
172.16.0.15	3078
172.16.0.5	3078
198.51.100.45	3078
198.51.100.60	3078
203.0.113.88	3078
203.0.113.90	3078
84.252.143.780	3078
84.252.143.785	3078
198.51.100.90	5
203.0.113.85	5
203.0.113.95	5
192.0.2.58	1
192.168.3.48	1
192.168.3.50	1
192.168.3.68	1

Src\_ip



24 Values, 99.999% of events

Selected

### Reports

[Top values](#)

[Top values by time](#)

[Rare values](#)

[Events with this field](#)

#### Top 10 Values

Count

%

84.252.143.78	33,696	40.611%	
178.159.37.10	21,546	25.967%	
198.51.100.75	3,082	3.714%	
172.16.0.15	3,078	3.71%	
172.16.0.5	3,078	3.71%	
198.51.100.45	3,078	3.71%	
198.51.100.60	3,078	3.71%	
203.0.113.88	3,078	3.71%	
203.0.113.90	3,078	3.71%	
84.252.143.780	3,078	3.71%	

## Destination\_ip



9 Values, 99.999% of events

Selected

Yes

No

### Reports

[Top values](#)

[Top values by time](#)

[Rare values](#)

[Events with this field](#)

Values	Count	%
8.8.8.8	18,474	22.265%
192.168.1.200	12,318	14.846%
10.0.0.1	12,156	14.65%
10.0.0.2	9,240	11.136%
10.10.0.5	9,239	11.135%
192.168.2.200	9,234	11.129%
192.168.2.100	6,156	7.419%
10.0.0.5	3,078	3.71%
10.10.0.10	3,078	3.71%

## Destination\_port



5 Values, 99.999% of events

Selected

Yes

No

### Reports

[Average over time](#)

[Maximum value over time](#)

[Minimum value over time](#)

[Top values](#)

[Top values by time](#)

[Rare values](#)

[Events with this field](#)

**Avg:** 130.88087691176648 **Min:** 0 **Max:** 443 **Std Dev:** 169.4878643130313

Values	Count	%
53	21,557	25.981%
0	21,547	25.969%
443	18,474	22.265%
80	18,317	22.076%
22	3,078	3.71%

- Report & Presentation on Qradar – Vraj

## CONTENT

- ABOUT
- TOP 10 FEATURES OF IBM QRADAR
- DEPLOYMENT OPTIONS
- COMPLIANCE
- SYSTEM REQUIREMENTS
- USE CASES
- PRICE COMPARISON WITH OTHER SIEM TOOLS
- DEMO

## PRICE COMPARISON WITH OTHER SIEM TOOLS

IBM QRADAR	SPLUNK
QRadar which is priced based on the events per second.	Splunk is priced based on the amount of data ingested on daily basis or the number of Splunk Virtual Compute (SVCs) units consumed (Workload Pricing), which can be more expensive than QRadar
\$740-\$1050 per month for 500 employees in organization(Software Based)  \$2300-\$3100 per month for 500 employees in organization(SaaS)	Pricing starts at \$150 per ingested GB of data per month.

5.4 WEEK – 4

- Email logs & Header Analysis

**ANY.RUN**  
INTERACTIVE MALWARE ANALYSIS

General Behavior MalConf Static information Video Screenshots System events Network  

---

## General Info

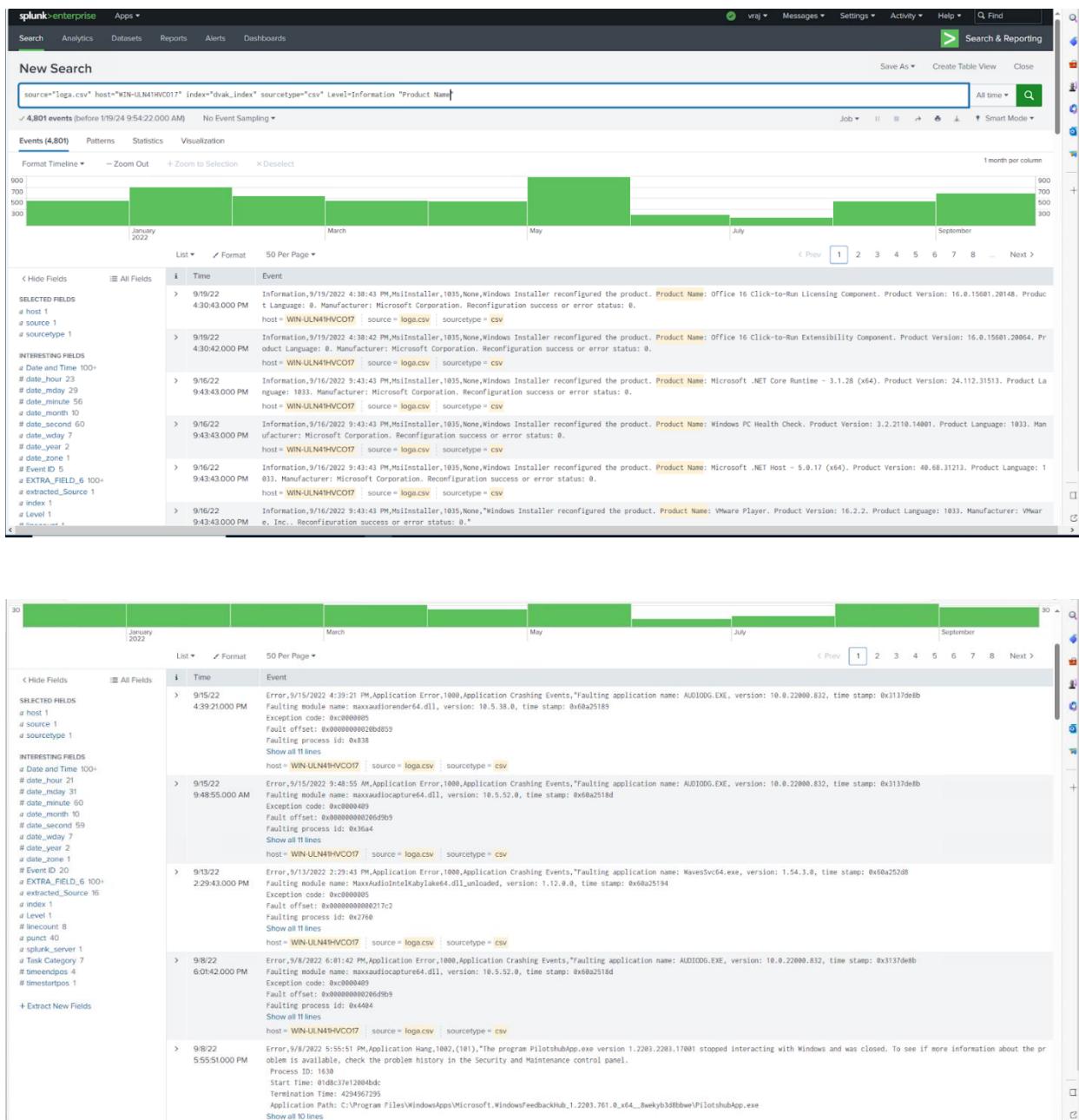
Add for printing 

File name:	What are the top 5 cloud security risks this year_.eml
Full analysis:	<a href="https://app.any.run/tasks/3c86f1d0-3e33-4472-9f80-7628ae346ee">https://app.any.run/tasks/3c86f1d0-3e33-4472-9f80-7628ae346ee</a>
Verdict:	<b>No threats detected</b>
Analysis date:	January 23, 2024 at 11:45:18
OS:	Windows 7 Professional Service Pack 1 (build: 7601, 32 bit)
Indicators:	 
MIME:	message/rfc822
File info:	RFC 822 mail, ASCII text, with very long lines, with CRLF line terminators
MD5:	400485FEE08D17FF4DFA6A6F567FF476
SHA1:	6492AA70CE8282147994C565C8B7FE8A77B66D9
SHA256:	E444A573DC8EF3FCA6C633CB17BFD85AA9C100FDDDEF3E86B8A340DB52060C42
SSDeep:	384:hoY1OyUoAsPtuOQEYUMaxcWg4uIb9mm+wRh5kPomZxdQbQuUMsg41Py

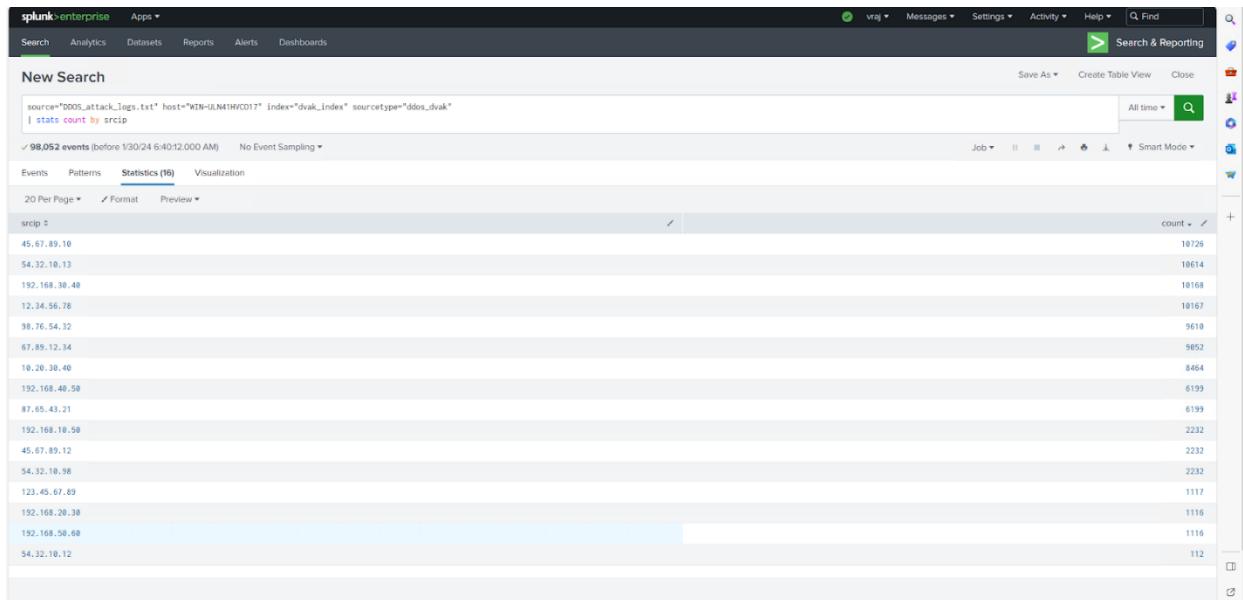
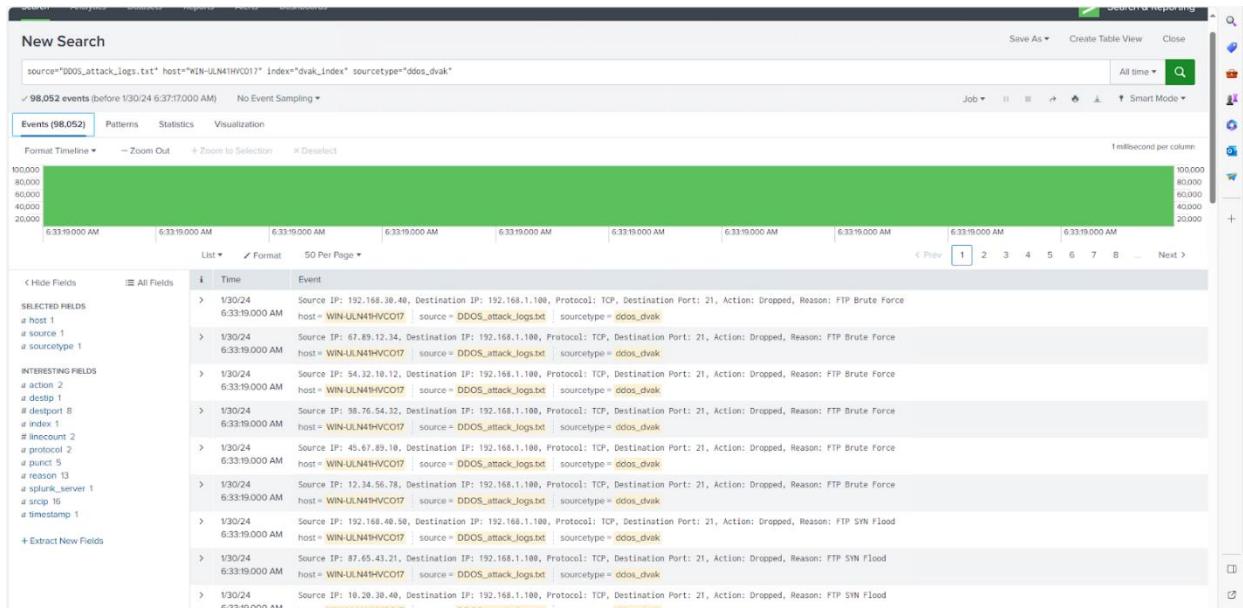
 [ANY.RUN](#) is an interactive service which provides full access to the guest system. Information in this report could be distorted by user actions and is provided for user acknowledgement as it is.  
[ANY.RUN](#) does not guarantee maliciousness or safety of the content.

Software environment set and analysis options 

## • Firewall Logs

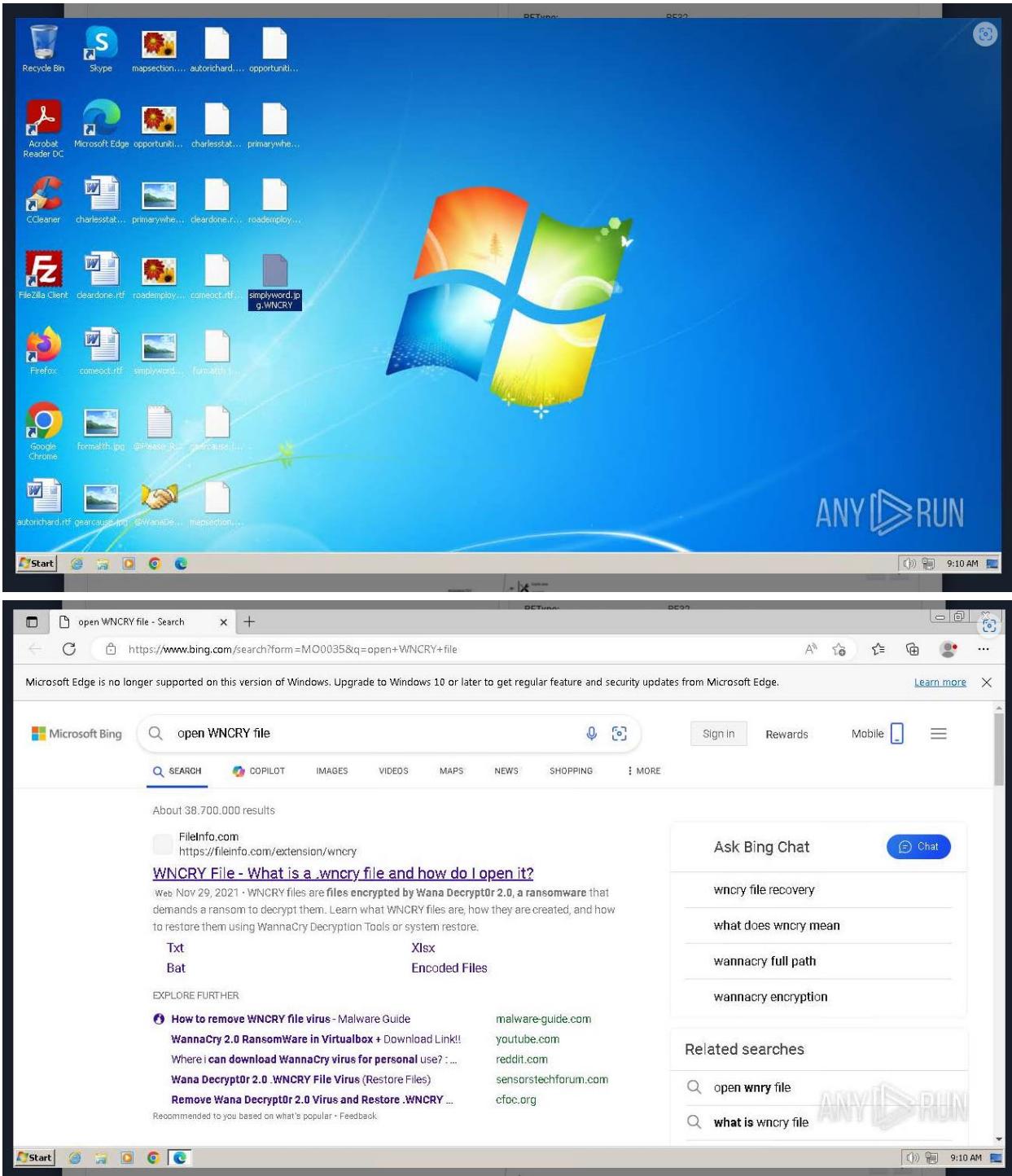


## ● DDoS Logs



## 5.5 WEEK – 5

- Malware Analysis





- Report on OEM
- Solarwinds – Vraj

**Dashboard**

KPI  
(last update: a few seconds ago)

Logs free storage 124 GiB	Used RAM 73 %	Used Memory 73 %	Used Storage 2.89 GiB	Oldest event occurred 599 days ago	EPS Last Hour 0 EPS
------------------------------	------------------	---------------------	--------------------------	---------------------------------------	------------------------

SEM Server Status  
The health and status of the SEM appliance. (last update: a few seconds ago)

EPS (last hour) 0 EPS	Used Memory 73 %	Free Memory 796 MiB
--------------------------	---------------------	------------------------

Monitor Time Series  
User Logons by Severity in Last hour (last update: a few seconds ago)

SEM Log Storage  
Health and Status of the SEM ap...

Oldest Event 599 days ago	Used Storage 45 %
------------------------------	----------------------

Node Health  
Nodes by last activity (last update: a few seconds ago)

Name	Last Event	Type	Upgrade status
2020-05-04 02:06:04	wpar-aix-test2	outdated	
2020-05-04 02:06:04	wpar-aix-test2	outdated	

Logon Failures by User  
Failed Logons by DestinationAccount (last update: a few seconds ago)

## Features

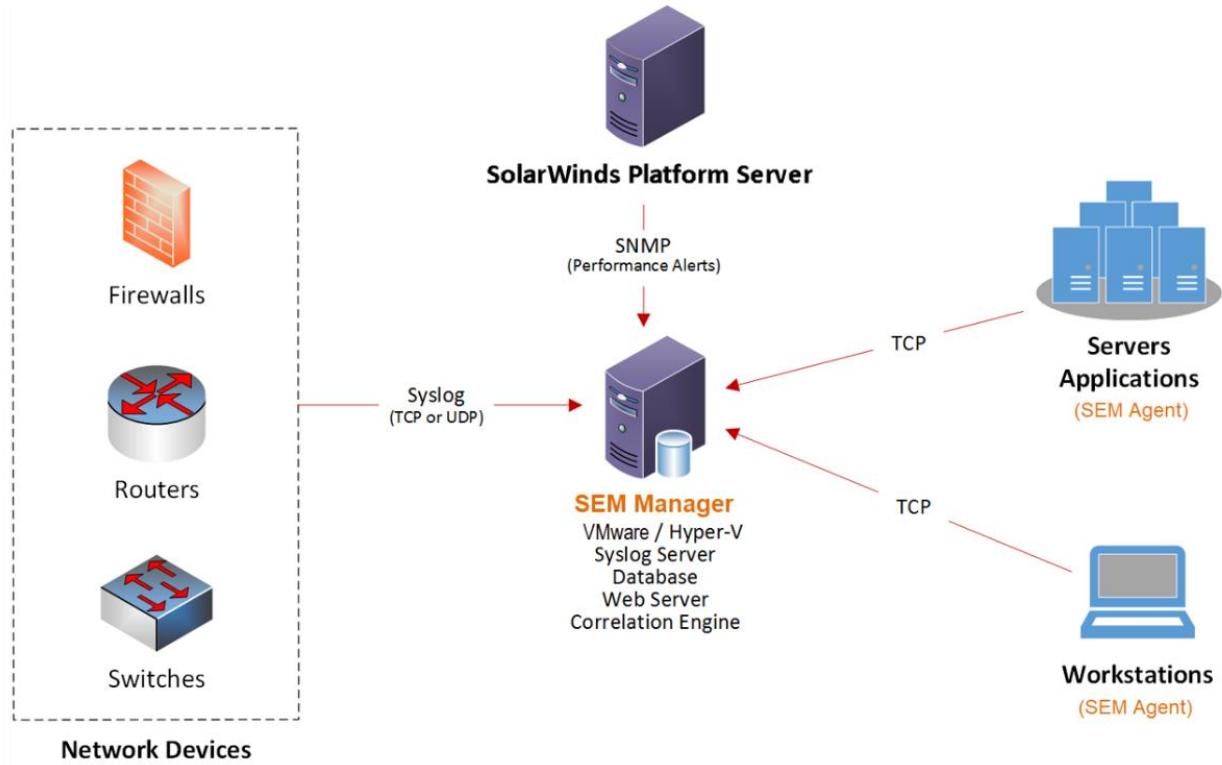
**COMPLIANCE REPORTING**

**CYBERTHREAT INTELLIGENCE**

**AUTOMATED INCIDENT RESPONSE**

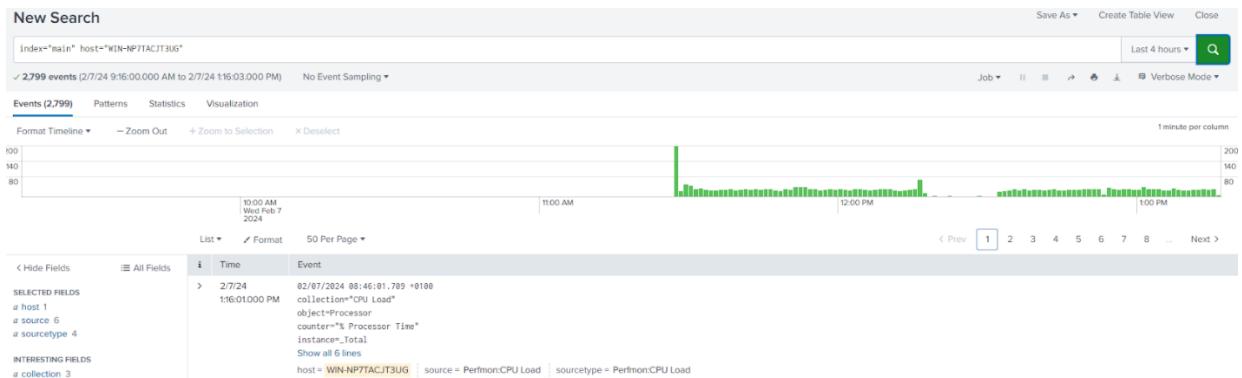
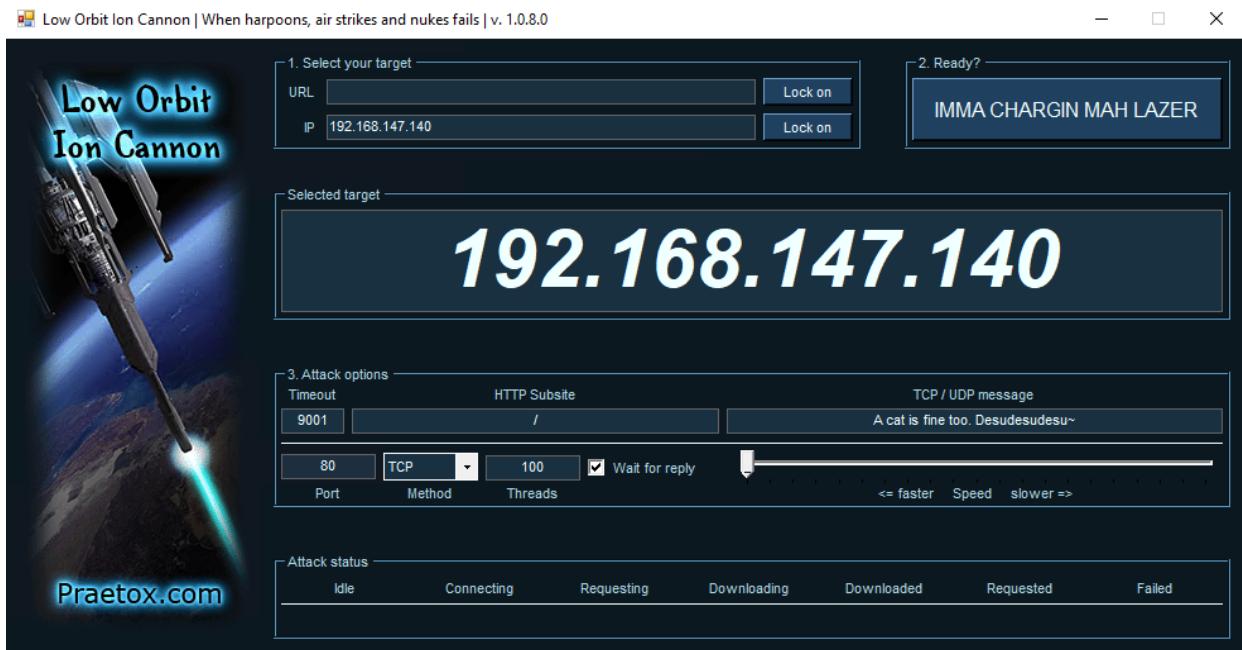
**FORENSIC ANALYSIS**

**FILE INTEGRITY MONITORING**



## 5.6 WEEK – 6

- DDoS attack practical & Logs collection



- Mimikatz attack practical & Logs collection

```
mimikatz 2.2.0 x86 (oe.eo)

#####
# mimikatz 2.2.0 <x86> #18362 Feb 29 2020 11:13:10
## ^ ##
## /> ## /*** Benjamin DELPY 'gentilkiwi' <benjamin@gentilkiwi.com>
## </ ## > http://blog.gentilkiwi.com/mimikatz
## v ## Vincent TOUX <vincent.letoux@gmail.com>
##### > http://pingcastle.com / http://mysmartlogon.com ***

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 190245 <00000000:0002e725>
Session : Interactive from 1
User Name : Jay
Domain : Jay-PC
Logon Server : JAY-PC
Logon Time : 4/24/2023 6:54:55 AM
SID : S-1-5-21-1235924641-3164596700-3031777028-1000
msv :
[00000003] Primary
* Username : Jay
* Domain : Jay-PC
* LM : 8f5eba968ca13d4209752a3293831d17
```

```
mimikatz 2.2.0 x86 (oe.eo)

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 190245 <00000000:0002e725>
Session : Interactive from 1
User Name : Jay
Domain : Jay-PC
Logon Server : JAY-PC
Logon Time : 4/24/2023 6:54:55 AM
SID : S-1-5-21-1235924641-3164596700-3031777028-1000
msv :
[00000003] Primary
* Username : Jay
* Domain : Jay-PC
* LM : 8f5eba968ca13d4209752a3293831d17
* NTLM : ec927c9081dcebb888073aac526a2528
* SHA1 : c59c4cad0bf9a40c277ba52323cb364027147d41
tspkg :
* Username : Jay
* Domain : Jay-PC
* Password : Jay@2809
wdigest :
* Username : Jay
* Domain : Jay-PC
* Password : Jay@2809
```

I	Time	Event
>	4/24/23 7:14:26.000 AM	04/23/2023 06:44:26 PM LogName=Security SourceName=Microsoft Windows security auditing EventCode=4672 EventType=0 Type=Information ComputerName=Jay-PC TaskCategory=Special Logon OpCode=Info RecordNumber=249 Keywords=Audit Success Message=Special privileges assigned to new logon.

Subject:

Security ID:	S-1-5-18
Account Name:	SYSTEM
Account Domain:	NT AUTHORITY
Logon ID:	0x1e7

## **CHAPTER: 6 WAZUH DEPLOYMENT**

## CHAPTER: 6 WAZUH DEPLOYMENT

### WAZUH Indexer –

#### Wazuh Indexer Install

##### Install Prerequisites

```
apt-get install debconf adduser procps apt-get install gnupg apt-transport-https
```

```
[root@parrot]~[~/Desktop/Wazuh]
[root@parrot]# apt-get install debconf adduser procps
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
debconf is already the newest version (1.5.82).
adduser is already the newest version (3.134).
procps is already the newest version (2:4.0.2-3).
0 upgraded, 0 newly installed, 0 to remove and 91 not upgraded.
[root@parrot]~[~/Desktop/Wazuh]
[root@parrot]# apt-get install gnupg apt-transport-https
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.2.40-1.1).
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 91 not upgraded.
Need to get 25.2 kB of archives.
After this operation, 35.8 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 https://deb.parrot.sh/parrot lory/main amd64 apt-transport-https all 2.6.1 [25.2 kB]
Fetched 25.2 kB in 0s (93.8 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 518140 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.6.1_all.deb ...
Unpacking apt-transport-https (2.6.1) ...
Setting up apt-transport-https (2.6.1) ...
Scanning application launchers
Removing duplicate launchers or broken launchers
Launchers are updated
```

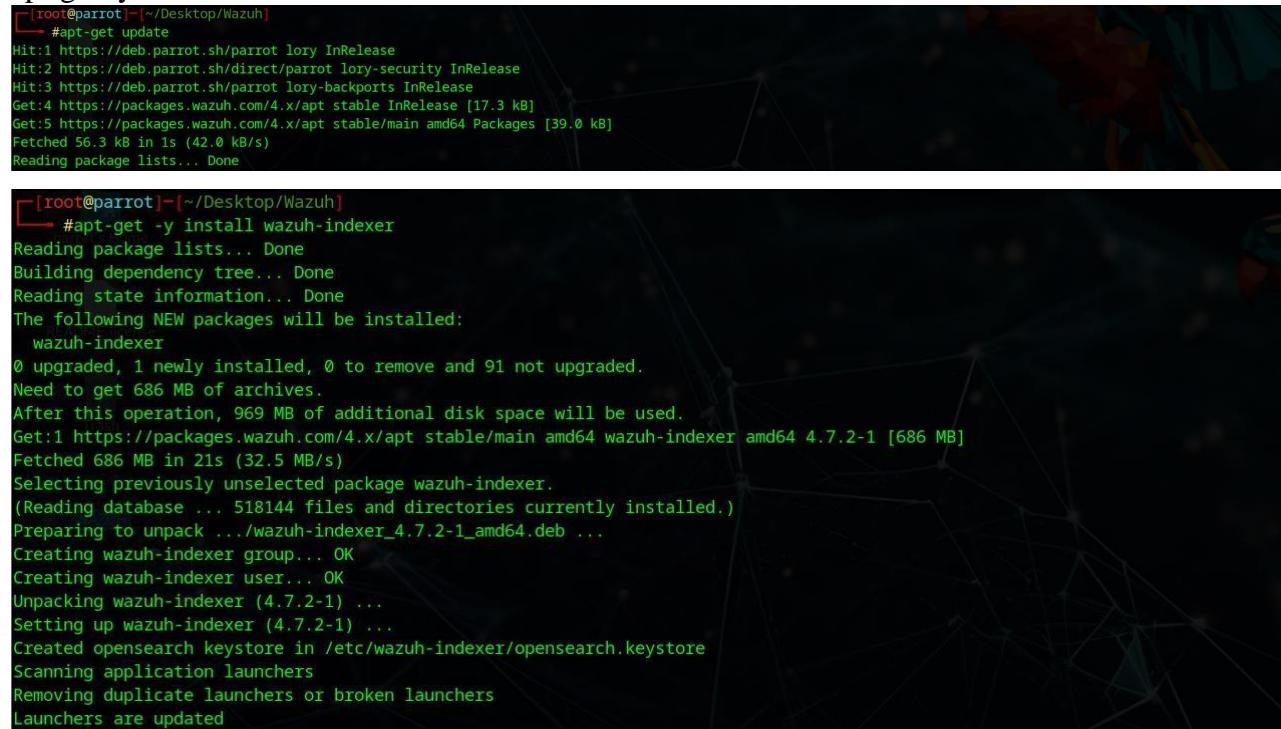
#### Install GPG Key and Add repo –

```
curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring \
--keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 \
/usr/share/keyrings/wazuh.gpg
echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ \
stable main" | tee -a \
/etc/apt/sources.list.d/wazuh.list
```

```
[root@parrot]~[~/Desktop/Wazuh]
[root@parrot]# curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && chmod 644 /usr/share/keyrings/wazuh.gpg
gpg: keyring '/usr/share/keyrings/wazuh.gpg' created
gpg: directory '/root/.gnupg' created
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: key 96B3E5F29111145: public key "Wazuh Signing Key" <support@wazuh.com> imported
gpg: Total number processed: 1
gpg: imported: 1
[root@parrot]~[~/Desktop/Wazuh]
[root@parrot]# echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | tee -a /etc/apt/sources.list.d/wazuh.list
deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main
[root@parrot]~[~/Desktop/Wazuh]
```

## Install

```
apt-get update  
apt-get -y install wazuh-indexer
```



```
[root@parrot]~[~/Desktop/Wazuh]  
# apt-get update  
Hit:1 https://deb.parrot.sh/parrot lory InRelease  
Hit:2 https://deb.parrot.sh/direct/parrot lory-security InRelease  
Hit:3 https://deb.parrot.sh/parrot lory-backports InRelease  
Get:4 https://packages.wazuh.com/4.x/apt stable InRelease [17.3 kB]  
Get:5 https://packages.wazuh.com/4.x/apt stable/main amd64 Packages [39.0 kB]  
Fetched 56.3 kB in 1s (42.0 kB/s)  
Reading package lists... Done  
  
[root@parrot]~[~/Desktop/Wazuh]  
#apt-get -y install wazuh-indexer  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  wazuh-indexer  
0 upgraded, 1 newly installed, 0 to remove and 91 not upgraded.  
Need to get 686 MB of archives.  
After this operation, 969 MB of additional disk space will be used.  
Get:1 https://packages.wazuh.com/4.x/apt stable/main amd64 wazuh-indexer amd64 4.7.2-1 [686 MB]  
Fetched 686 MB in 21s (32.5 MB/s)  
Selecting previously unselected package wazuh-indexer.  
(Reading database ... 518144 files and directories currently installed.)  
Preparing to unpack .../wazuh-indexer_4.7.2-1_amd64.deb ...  
Creating wazuh-indexer group... OK  
Creating wazuh-indexer user... OK  
Unpacking wazuh-indexer (4.7.2-1) ...  
Setting up wazuh-indexer (4.7.2-1) ...  
Created opensearch keystore in /etc/wazuh-indexer/opensearch.keystore  
Scanning application launchers  
Removing duplicate launchers or broken launchers  
Launchers are updated
```

## Certificate Deployment

Deploy certificates for encryption and security.

```
wget https://raw.githubusercontent.com/socfortress/Wazuh-Rules/main/wazuh-certstool.sh -q -O /tmp/wazuh-certs-tool.sh wget https://raw.githubusercontent.com/socfortress/Wazuh-Rules/main/config.yml -q -O /tmp/config.yml
```



```
[root@parrot]~[~/Desktop/Wazuh]  
# wget https://raw.githubusercontent.com/socfortress/Wazuh-Rules/main/wazuh-certs-tool.sh -q -O /tmp/wazuh-certs-tool.sh  
[root@parrot]~[~/Desktop/Wazuh]  
# wget https://raw.githubusercontent.com/socfortress/Wazuh-Rules/main/config.yml -q -O /tmp/config.yml
```

## 2. Update the /tmp/config.yml file to fit your hostname and IP.



```
root nano 7:2  
nodes:  
  # Wazuh indexer nodes  
  indexer:  
    - name: wazuh-indexer01.vraj.demo  
      ip: 10.10.9.130  
    # name: node-2  
    # ip: <indexer-node-ip>  
    # name: node-3  
    # ip: <indexer-node-ip>  
  
  # Wazuh server nodes  
  # If there is more than one Wazuh server  
  # node, each one must have a node_type  
  server:  
    - name: wazuh-indexer01.vraj.demo  
      ip: 10.10.9.130  
    # node_type: master  
    # name: wazuh-2  
    # ip: <wazuh-manager-ip>  
    # node_type: worker  
    # name: wazuh-3  
    # ip: <wazuh-manager-ip>  
    # node_type: worker  
  
  # Wazuh dashboard nodes  
  dashboard:  
    - name: wazuh-indexer01.vraj.demo  
      ip: 10.10.9.130
```

### 3. Run the /tmp/wazuh-certs-tool.sh -A script to generate the certificates.

```
[root@parrot:~/tmp]
# ./tmp/wazuh-certs-tool.sh -A
INFO: Admin certificates created.
19/02/2024 12:47:03 INFO: Wazuh indexer certificates created.
19/02/2024 12:47:03 INFO: Wazuh server certificates created.
19/02/2024 12:47:03 INFO: Wazuh dashboard certificates created.

[root@parrot:~/tmp]
# ls
config.yml
hyperdata_root
hyperdata.yaml
wazuh-XXXXXXunitmx
systemd-private-5d3507ae21b84125bb24c4bcc1441c07-bluetooth.service-iV9tQZ
systemd-private-5d3507ae21b84125bb24c4bcc1441c07-haveged.service-fqPla0
systemd-private-5d3507ae21b84125bb24c4bcc1441c07-modemmanager.service-RJ815j
systemd-private-5d3507ae21b84125bb24c4bcc1441c07-systemd-logind.service-kUgce
systemd-private-5d3507ae21b84125bb24c4bcc1441c07-upower.service-FRjTlq
wazuh-certs-tool.sh
wazuh-install-files
```

### 4. Obtain the value of the CN of the hostname.pem certificate.

openssl x509 -in wazuh-indexer01.vraj.demo -text -noout  
 NODE\_NAME=wazuh-indexer01.vraj.demo

```
[root@parrot:~/tmp/wazuh-certificates]
# openssl x509 -in wazuh-indexer01.vraj.demo -text -noout
certificate from wazuh-indexer01.vraj.demo
407760F1847F0000:error:16000069:STORE routines:ossl_store_get0_loader_int:unregistered scheme:../crypto/store/store_register.c:237:scheme=file
407760F1847F0000:error:00000002:system library:file_open:No such file or directory:../providers/implementations/storemgmt/file_store.c:267:calling stat(wazuh-indexer01.vraj.demo)
Unable to load certificate
[X] #root@parrot:~/tmp/wazuh-certificates]
#openssl x509 -in wazuh-indexer01.vraj.demo.pem -text -noout
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
 05:81:9c:91:6e:7e:23:4a:f4:b5:cb:cd:b0:1e:fc:17:0:a:ea:27:c2
Signature Algorithm: sha256WithRSAEncryption
Issuer: OU = Wazuh, O = Wazuh, L = California
Validity
  Not Before: Feb 19 07:17:03 2024 GMT
  Not After : Feb 16 07:17:03 2034 GMT
Subject: C = US, L = Texas, O = SOCFortress, OU = SOCFortress, CN = wazuh-indexer01.vraj.demo
Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
      Modulus:
        00:bd:ad:bc:2b:30:45:5e:f2:77:5f:7d:ab:1e:fc:
```

### 5. Copy certs into /etc/wazuh-indexer/certs

mkdir /etc/wazuh-indexer/certs cd /tmp/wazuh-certificates cp ./\${NODE\_NAME}.pem  
 ./\${NODE\_NAME}-key.pem ./admin.pem ./admin-key.pem ./rootca.pem /etc/wazuh-indexer/certs/  
 mv -n /etc/wazuh-indexer/certs/\${NODE\_NAME}.pem /etc/wazuhindexer/certs/indexer.pem mv -n /etc/wazuh-indexer/certs/\${NODE\_NAME}-key.pem /etc/wazuhindexer/certs/indexer-key.pem

```
[root@parrot:~/]
# cd tmp/wazuh-certificates
[root@parrot:~/tmp/wazuh-certificates]
# cp ./${NODE_NAME}.pem ./${NODE_NAME}-key.pem ./admin.pem ./admin-key.pem ./root-ca.pem /etc/wazuh-indexer/certs/
[root@parrot:~/tmp/wazuh-certificates]
# mv -n /etc/wazuh-indexer/certs/${NODE_NAME}.pem /etc/wazuh-indexer/certs/indexer.pem
[root@parrot:~/tmp/wazuh-certificates]
# mv -n /etc/wazuh-indexer/certs/${NODE_NAME}-key.pem /etc/wazuh-indexer/certs/indexer-key.pem
[root@parrot:~/tmp/wazuh-certificates]
# cd /etc/wazuh-indexer/certs
[root@parrot:~/etc/wazuh-indexer/certs]
```

### 6. Set ownership and permissions

chmod 500 /etc/wazuh-indexer/certs chmod 400 /etc/wazuh-indexer/certs/\* chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs

```
[root@parrot:~/tmp/wazuh-certificates]
#ls
total 24K
-rwx--r-- 1 root root 1.7K Feb 19 12:47 admin-key.pem
-rwx--r-- 1 root root 1.7K Feb 19 12:47 admin.pem
-rwx--r-- 1 root root 1.7K Feb 19 12:47 root-ca.key
-rwx--r-- 1 root root 1.2K Feb 19 12:47 root-ca.pem
-rwx--r-- 1 root root 1.7K Feb 19 12:47 wazuh-indexer01.vraj.demo.key.pem
-rwx--r-- 1 root root 1.4K Feb 19 12:47 wazuh-indexer01.vraj.demo.pem
[root@parrot:~/tmp/wazuh-certificates]
#chmod 500 /etc/wazuh-indexer/certs
[root@parrot:~/tmp/wazuh-certificates]
#chmod 400 /etc/wazuh-indexer/certs/*
[root@parrot:~/tmp/wazuh-certificates]
#chown -R wazuh-indexer:wazuh-indexer /etc/wazuh-indexer/certs
[root@parrot:~/tmp/wazuh-certificates]
#
```

## Wazuh Indexer Configuration

```
File Edit View Search Terminal Help
GNU nano 7.2                                     /opensearch.yml
network.host: "0.0.0.0"
node.name: "wazuh-indexer01.vraj.demo"
cluster.initial_master_nodes:
- "wazuh-indexer01.vraj.demo"
cluster.name: "vraj_demo"
discovery.seed_hosts:
- "wazuh-indexer01.vraj.demo"
node.max_local_storage_nodes: "3"
path.data: "/var/lib/wazuh-indexer"
path.logs: "/var/log/wazuh-indexer"

bootstrap.memory_lock: true

plugins.security.ssl.http.pemcert_filepath: /etc/wazuh-indexer/certs/indexer.pem
plugins.security.ssl.http.pemkey_filepath: /etc/wazuh-indexer/certs/indexer-key.pem
plugins.security.ssl.http.pemtrustedcas_filepath: /etc/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.transport.pemcert_filepath: /etc/wazuh-indexer/certs/indexer.pem
plugins.security.ssl.transport.pemkey_filepath: /etc/wazuh-indexer/certs/indexer-key.pem
plugins.security.ssl.transport.pemtrustedcas_filepath: /etc/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.http.enabled: true
plugins.security.ssl.transport.enforce_hostname_verification: false
plugins.security.ssl.transport.resolve_hostname: false

plugins.security.authcz.admin_dn:
  "CN=admin,OU=50CFortress,O=50CFortress,L=Texas,C=US"
plugins.security.check_snapshot_restore_write_privileges: true
plugins.security.enable_snapshot_restore_privilege: true
plugins.security.nodes_dn:

[[{"Help": "Help", "Read File": "Read File", "Replace": "Replace", "Paste": "Paste", "Go To Line": "Go To Line", "Redo": "Redo", "Copy": "Copy", "Where Was": "Where Was", "Next": "Next", "Forward": "Forward", "Prev Word": "Prev Word", "Home": "Home"}, {"Exit": "Exit", "Where Is": "Where Is", "Cut": "Cut", "Execute": "Execute", "Undo": "Undo", "Set Mark": "Set Mark", "To Bracket": "To Bracket", "Previous": "Previous", "Back": "Back", "Next Word": "Next Word", "Home": "Home"}], [{"text": "\u25a0 Read 39 lines"}]]
```

## Memory Locking

Uncomment or add this line to the /etc/wazuh-indexer/opensearch.yml file:  
bootstrap.memory\_lock: true

```
File Edit View Search Terminal Help
GNU nano 7.2                                     /opensearch.yml
network.host: "0.0.0.0"
node.name: "wazuh-indexer01.vraj.demo"
cluster.initial_master_nodes:
- "wazuh-indexer01.vraj.demo"
cluster.name: "vraj_demo"
discovery.seed_hosts:
- "wazuh-indexer01.vraj.demo"
node.max_local_storage_nodes: "3"
path.data: "/var/lib/wazuh-indexer"
path.logs: "/var/log/wazuh-indexer"

bootstrap.memory_lock: true

plugins.security.ssl.http.pemcert_filepath: /etc/wazuh-indexer/certs/indexer.pem
plugins.security.ssl.http.pemkey_filepath: /etc/wazuh-indexer/certs/indexer-key.pem
plugins.security.ssl.http.pemtrustedcas_filepath: /etc/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.transport.pemcert_filepath: /etc/wazuh-indexer/certs/indexer.pem
plugins.security.ssl.transport.pemkey_filepath: /etc/wazuh-indexer/certs/indexer-key.pem
plugins.security.ssl.transport.pemtrustedcas_filepath: /etc/wazuh-indexer/certs/root-ca.pem
plugins.security.ssl.http.enabled: true
plugins.security.ssl.transport.enforce_hostname_verification: false
plugins.security.ssl.transport.resolve_hostname: false

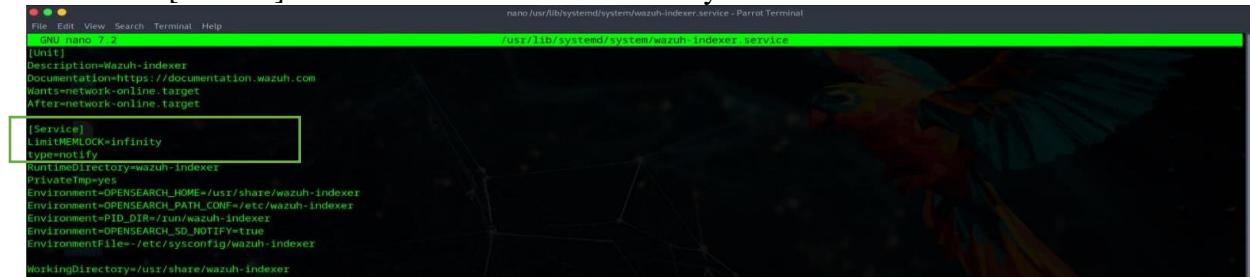
plugins.security.authcz.admin_dn:
  "CN=admin,OU=50CFortress,O=50CFortress,L=Texas,C=US"
plugins.security.check_snapshot_restore_write_privileges: true
plugins.security.enable_snapshot_restore_privilege: true
plugins.security.nodes_dn:

[[{"Help": "Help", "Read File": "Read File", "Replace": "Replace", "Paste": "Paste", "Go To Line": "Go To Line", "Redo": "Redo", "Copy": "Copy", "Where Was": "Where Was", "Next": "Next", "Forward": "Forward", "Prev Word": "Prev Word", "Home": "Home"}, {"Exit": "Exit", "Where Is": "Where Is", "Cut": "Cut", "Execute": "Execute", "Undo": "Undo", "Set Mark": "Set Mark", "To Bracket": "To Bracket", "Previous": "Previous", "Back": "Back", "Next Word": "Next Word", "Home": "Home"}], [{"text": "\u25a0 Read 39 lines"}]]
```

2. Edit the limit of system resources:

nano /usr/lib/systemd/system/wazuh-indexer.service

Place under [Service] block LimitMEMLOCK=infinity



```
File Edit View Search Terminal Help
GNU nano 7.2
[Unit]
Description=Wazuh-indexer
Documentation=https://documentation.wazuh.com
Wants=network-online.target
After=network-online.target

[Service]
LimitMEMLOCK=infinity
WorkingDirectory=/var/lib/wazuh/Indexer
PrivateTmp=yes
Environment=OPENSEARCH_HOME=/usr/share/wazuh-indexer
Environment=OPENSEARCH_PATH_CONF=/etc/wazuh-indexer
Environment=PID_DIR=/run/wazuh-indexer
Environment=OPENSEARCH_SD_NOTIFY=true
Environmentfile=/etc/sysconfig/wazuh-indexer
WorkingDirectory=/usr/share/wazuh-indexer
```

3. Set JVM Options to 50% of total memory available

nano /etc/wazuh-indexer/jvm.options



```
File Edit View Search Terminal Help
GNU nano 7.2
# JVM configuration

#####
## IMPORTANT: JVM heap size
## You should always set the min and max JVM heap
## size to the same value. For example, to set
## the heap to 4 GB, set:
##
## -Xms4g
## -Xmx4g
##
## See https://opensearch.org/docs/opensearch/install/important-settings/
## for more information
##
#####

# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space

-Xms2g
-Xmx2g

#####
## Expert settings
#####
##
```

change Xms1g to Xms2g.

## Start the Service

```
systemctl daemon-reload
```

```
systemctl enable wazuh-indexer
```

```
systemctl start wazuh-indexer
```

```
[root@parrot ~]# systemctl daemon-reload
[root@parrot ~]# systemctl enable wazuh-indexer
[root@parrot ~]# systemctl start wazuh-indexer
[root@parrot ~]# systemctl status wazuh-indexer.service
● wazuh-indexer.service - Wazuh-indexer
   Loaded: loaded (/lib/systemd/system/wazuh-indexer.service; enabled; preset: enabled)
     Active: active (running) since Mon 2024-02-19 15:07:43 IST; 27s ago
       Docs: https://documentation.wazuh.com
      Main PID: 1083 (java)
        Tasks: 58 (limit: 9374)
       Memory: 2.7G
          CPU: 51.172s
         CGroup: /system.slice/wazuh-indexer.service
                  └─1083 /usr/share/wazuh-indexer/jdk/bin/java -Xshare:auto -Dopensearch.networkAddress.cache.ttl=60 -Dopensearch.networkAddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m

Feb 19 15:07:50 parrot systemd-entrypoint[1083]: WARNING: A terminally deprecated method in java.lang.System has been called
Feb 19 15:07:50 parrot systemd-entrypoint[1083]: WARNING: System::setSecurityManager has been called by org.opensearch.bootstrap.OpenSearch (file:/usr/share/wazuh-indexer/lib/opensearch-2.8.0.jar)
Feb 19 15:07:50 parrot systemd-entrypoint[1083]: WARNING: Please consider reporting this to the maintainers of org.opensearch.bootstrap.OpenSearch
Feb 19 15:07:50 parrot systemd-entrypoint[1083]: WARNING: System::setSecurityManager will be removed in a future release
Feb 19 15:07:36 parrot systemd-entrypoint[1083]: WARNING: A terminally deprecated method in java.lang.System has been called
Feb 19 15:07:36 parrot systemd-entrypoint[1083]: WARNING: System::setSecurityManager has been called by org.opensearch.bootstrap.Security (file:/usr/share/wazuh-indexer/lib/opensearch-2.8.0.jar)
Feb 19 15:07:36 parrot systemd-entrypoint[1083]: WARNING: Please consider reporting this to the maintainers of org.opensearch.bootstrap.Security
Feb 19 15:07:36 parrot systemd-entrypoint[1083]: WARNING: System::setSecurityManager will be removed in a future release
Feb 19 15:07:59 parrot systemd[1]: /lib/systemd/system/wazuh-indexer.service:9: Unknown key name 'type' in section 'Service', ignoring.
Feb 19 15:08:03 parrot systemd[1]: /lib/systemd/system/wazuh-indexer.service:9: Unknown key name 'type' in section 'Service', ignoring.
```

## Cluster Initialization

```
/usr/share/wazuh-indexer/bin/indexer-security-init.sh
```

```
[x]-[root@parrot ~]#
[root@parrot ~]# ./usr/share/wazuh-indexer/bin/indexer-security-init.sh
=====
** This tool will be deprecated in the next major release of OpenSearch **
** https://github.com/opensearch-project/security/issues/1755           **
=====

Security Admin v7
Will connect to 127.0.0.1:9200 ... done
Connected as "CN=admin,OU=SOCFortress,O=SOCFortress,L=Texas,C=US"
OpenSearch Version: 2.8.0
Contacting opensearch cluster 'opensearch' and wait for YELLOW clusterstate ...
clustername: vraj_demo
Clusterstate: GREEN
Number of nodes: 1
Number of data nodes: 1
.opendistro_security index does not exists, attempt to create it ... done (0-all replicas)
Populate config from /etc/wazuh-indexer/opensearch-security/
Will update '/config' with /etc/wazuh-indexer/opensearch-security/config.yml
  SUCC: Configuration for 'config' created or updated
Will update '/roles' with /etc/wazuh-indexer/opensearch-security/roles.yml
  SUCC: Configuration for 'roles' created or updated
Will update '/rolesmapping' with /etc/wazuh-indexer/opensearch-security/roles_mapping.yml
  SUCC: Configuration for 'rolesmapping' created or updated
Will update '/internalusers' with /etc/wazuh-indexer/opensearch-security/internal_users.yml
  SUCC: Configuration for 'internalusers' created or updated
Will update '/actiongroups' with /etc/wazuh-indexer/opensearch-security/action_groups.yml
  SUCC: Configuration for 'actiongroups' created or updated
Will update '/tenants' with /etc/wazuh-indexer/opensearch-security/tenants.yml
  SUCC: Configuration for 'tenants' created or updated
Will update '/nodesdn' with /etc/wazuh-indexer/opensearch-security/nodes_dn.yml
```

## Install Wazuh-Dashboard

Only needs to be installed on Wazuh-Indexer Node 01

Install Prerequisites:

```
apt-get install debhelper tar curl libcap2-bin -y apt-get update
```

```
[root@parrot ~]# apt-get install debhelper tar curl libcap2-bin -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
tar is already the newest version (1.34+dfsg-1.2).
curl is already the newest version (8.5.0-2-bpo12+1).
libcap2-bin is already the newest version (1:2.66-4).
The following additional packages will be installed:
  autopoint dh-autoreconf dh-strip-nondeterminism dwz libarchive-cpio-perl libdebscript-perl libfile-stripnondeterminism-perl libmail-sendmail-perl libsub-override-perl
  libsys-hostname-long-perl po-debconf
Suggested packages:
  dh-make libmail-box-perl
The following NEW packages will be installed:
  autopoint debhelper dh-autoreconf dh-strip-nondeterminism dwz libarchive-cpio-perl libdebscript-perl libfile-stripnondeterminism-perl libmail-sendmail-perl libsub-override-perl
  libsys-hostname-long-perl po-debconf
0 upgraded, 12 newly installed, 0 to remove and 91 not upgraded.
Need to get 1,978 kB of archives.
After this operation, 5,297 kB of additional disk space will be used.
Get:1 https://deb.parrot.sh/parrot lory/main amd64 autopoint all 0.21-12 [495 kB]
Get:2 https://deb.parrot.sh/parrot lory/main amd64 libdebscript-perl all 13.11.4 [81.2 kB]
Get:3 https://deb.parrot.sh/parrot lory/main amd64 dh-autoreconf all 20 [17.1 kB]
[1/3] # apt-get update
Hit:1 https://deb.parrot.sh/parrot lory InRelease
Hit:2 https://deb.parrot.sh/direct/parrot lory-security InRelease
Hit:3 https://packages.wazuh.com/4.x/apt stable InRelease
Reading package lists...
W: Target Packages (main/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
W: Target Packages (main/binary-all/Packages) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
W: Target Translations (main/i18n/Translation-en_IN) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
W: Target Translations (main/i18n/Translation-en) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
W: Target Packages (contrib/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
W: Target Packages (contrib/binary-all/Packages) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
W: Target Translations (contrib/i18n/Translation-en_IN) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
W: Target Translations (contrib/i18n/Translation-en) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
W: Target Packages (non-free/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
W: Target Packages (non-free/binary-all/Packages) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
W: Target Translations (non-free/i18n/Translation-en_IN) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
W: Target Translations (non-free/i18n/Translation-en) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
W: Target Packages (main/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
W: Target Translations (main/i18n/Translation-en_IN) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
```

## 2. Install:

```
apt-get -y install wazuh-dashboard
```

```
[root@parrot ~]# apt-get -y install wazuh-dashboard
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  wazuh-dashboard
0 upgraded, 1 newly installed, 0 to remove and 91 not upgraded.
Need to get 179 MB of archives.
After this operation, 965 MB of additional disk space will be used.
Get:1 https://packages.wazuh.com/4.x/apt stable/main amd64 wazuh-dashboard amd64 4.7.2-1 [179 MB]
Fetched 179 MB in 6s (30.7 MB/s)
Selecting previously unselected package wazuh-dashboard.
(Reading database ... 519911 files and directories currently installed.)
Preparing to unpack .../wazuh-dashboard_4.7.2-1_amd64.deb ...
Creating wazuh-dashboard group... OK
Creating wazuh-dashboard user... OK
Unpacking wazuh-dashboard (4.7.2-1) ...
Setting up wazuh-dashboard (4.7.2-1) ...
Scanning application launchers
Removing duplicate launchers or broken launchers
Launchers are updated
```

## Configure Wazuh-Dashboard

Configure certificates so that Wazuh-Dashboard service can connect to the Wazuh-Indexer cluster.

```
mkdir /etc/wazuh-dashboard/certs
cp /etc/wazuh-indexer/certs/indexer.pem /etc/wazuh-dashboard/certs/
cp /etc/wazuh-indexer/certs/indexer-key.pem /etc/wazuh-dashboard/certs/
cp /etc/wazuh-indexer/certs/root-ca.pem /etc/wazuh-dashboard/certs/
chmod 500 /etc/wazuh-dashboard/certs/
chmod 400 /etc/wazuh-dashboard/certs/*
chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs
```

```
# cp /etc/wazuh-indexer/certs/indexer.pem /etc/wazuh-dashboard/certs/
# cp /etc/wazuh-indexer/certs/indexer-key.pem /etc/wazuh-dashboard/certs/
# cp /etc/wazuh-indexer/certs/root-ca.pem /etc/wazuh-dashboard/certs/
# chmod 500 /etc/wazuh-dashboard/certs/
# chmod 400 /etc/wazuh-dashboard/certs/*
# chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs
```

```
[root@parrot]# cd /etc/wazuh-indexer/certs
[root@parrot]# mkdir /etc/wazuh-dashboard/certs
[root@parrot]# cp /etc/wazuh-indexer/certs/indexer.pem /etc/wazuh-dashboard/certs/
[root@parrot]# cp /etc/wazuh-indexer/certs/indexer-key.pem /etc/wazuh-dashboard/certs/
[root@parrot]# cp /etc/wazuh-indexer/certs/root-ca.pem /etc/wazuh-dashboard/certs/
[root@parrot]# chmod 500 /etc/wazuh-dashboard/certs/
[root@parrot]# chmod 400 /etc/wazuh-dashboard/certs/*
[root@parrot]# chown -R wazuh-dashboard:wazuh-dashboard /etc/wazuh-dashboard/certs
```

```
[root@parrot]# nano /etc/wazuh-dashboard/opensearch_dashboards.yml
server.host: 0.0.0.0
server.port: 443
opensearch.hosts: ["https://wazuh-indexer01.vraj.demo:9200"]
opensearch.ssl.verificationMode: certificate
#opensearch.username:
#opensearch.password:
opensearch.requestHeadersWhitelist: ["securitytenant", "Authorization"]
opensearch_security.multitenancy.enabled: false
opensearch_security.readonly_mode.roles: ["kibana_read_only"]
server.ssl.enabled: true
server.ssl.key: "/etc/wazuh-dashboard/certs/indexer-key.pem"
server.ssl.certificate: "/etc/wazuh-dashboard/certs/indexer.pem"
opensearch.ssl.certificateAuthorities: ["/etc/wazuh-dashboard/certs/root-ca.pem"]
uiSettings.overrides.defaultRoute: /app/wazuh
```

## Start Wazuh-Dashboard

```
systemctl daemon-reload
systemctl enable wazuh-dashboard
systemctl start wazuh-dashboard
```

```
File Edit View Search Terminal Help
Every 2.0s: systemctl status wazuh-dashboard.service
Every 2.0s: systemctl status wazuh-dashboard.service
parrot: Mon Feb 19 15:35:44 2024
● wazuh-dashboard.service - wazuh-dashboard
   Loaded: loaded (/etc/systemd/system/wazuh-dashboard.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-02-19 15:26:19 IST; 9min ago
     Main PID: 3372 (node)
       Tasks: 11 (limit: 9374)
      Memory: 143.3M
        CPU: 8.640s
       CGroup: /system.slice/wazuh-dashboard.service
           └─3372 /usr/share/wazuh-dashboard/node/bin/node --no-warnings --max-http-header-size=65536 --unhandled-rejections=warn /usr/share/wazuh-dashboard/src/cli/dist -c /etc/wazuh-dashboard/opensearch_dashboards.yml

Feb 19 15:35:19 parrot opensearch-dashboards[3372]: {"type": "log", "@timestamp": "2024-02-19T10:05:19Z", "tags": ["error", "opensearch", "data"], "pid": 3372, "message": "[ConnectionError]: getaddrinfo ENOTFOUND wazuh-indexer01.vraj.demo"}
Feb 19 15:35:22 parrot opensearch-dashboards[3372]: {"type": "log", "@timestamp": "2024-02-19T10:05:22Z", "tags": ["error", "opensearch", "data"], "pid": 3372, "message": "[ConnectionError]: getaddrinfo ENOTFOUND wazuh-indexer01.vraj.demo"}
Feb 19 15:35:24 parrot opensearch-dashboards[3372]: {"type": "log", "@timestamp": "2024-02-19T10:05:24Z", "tags": ["error", "opensearch", "data"], "pid": 3372, "message": "[ConnectionError]: getaddrinfo ENOTFOUND wazuh-indexer01.vraj.demo"}
Feb 19 15:35:27 parrot opensearch-dashboards[3372]: {"type": "log", "@timestamp": "2024-02-19T10:05:27Z", "tags": ["error", "opensearch", "data"], "pid": 3372, "message": "[ConnectionError]: getaddrinfo ENOTFOUND wazuh-indexer01.vraj.demo"}
Feb 19 15:35:29 parrot opensearch-dashboards[3372]: {"type": "log", "@timestamp": "2024-02-19T10:05:29Z", "tags": ["error", "opensearch", "data"], "pid": 3372, "message": "[ConnectionError]: getaddrinfo ENOTFOUND wazuh-indexer01.vraj.demo"}
Feb 19 15:35:32 parrot opensearch-dashboards[3372]: {"type": "log", "@timestamp": "2024-02-19T10:05:32Z", "tags": ["error", "opensearch", "data"], "pid": 3372, "message": "[ConnectionError]: getaddrinfo ENOTFOUND wazuh-indexer01.vraj.demo"}
Feb 19 15:35:34 parrot opensearch-dashboards[3372]: {"type": "log", "@timestamp": "2024-02-19T10:05:34Z", "tags": ["error", "opensearch", "data"], "pid": 3372, "message": "[ConnectionError]: getaddrinfo ENOTFOUND wazuh-indexer01.vraj.demo"}
Feb 19 15:35:37 parrot opensearch-dashboards[3372]: {"type": "log", "@timestamp": "2024-02-19T10:05:37Z", "tags": ["error", "opensearch", "data"], "pid": 3372, "message": "[ConnectionError]: getaddrinfo ENOTFOUND wazuh-indexer01.vraj.demo"}
Feb 19 15:35:39 parrot opensearch-dashboards[3372]: {"type": "log", "@timestamp": "2024-02-19T10:05:39Z", "tags": ["error", "opensearch", "data"], "pid": 3372, "message": "[ConnectionError]: getaddrinfo ENOTFOUND wazuh-indexer01.vraj.demo"}
Feb 19 15:35:42 parrot opensearch-dashboards[3372]: {"type": "log", "@timestamp": "2024-02-19T10:05:42Z", "tags": ["error", "opensearch", "data"], "pid": 3372, "message": "[ConnectionError]: getaddrinfo ENOTFOUND wazuh-indexer01.vraj.demo"}
```

## Securing The Cluster

Update the default passwords of the admin and wazuh-dashboard users.

On the Wazuh Indexer Node 01, use the Wazuh passwords tool to change the passwords of the Wazuh indexer users.

Run Script

```
/usr/share/wazuh-indexer/plugins/opensearch-security/tools/wazuh-passwordstool.sh --change-all
```

```
[root@parrot :~]# /usr/share/wazuh-indexer/plugins/opensearch-security/tools/wazuh-passwordstool.sh --change-all
[...]
# /usr/share/wazuh-indexer/plugins/opensearch-security/tools/wazuh-passwords-tool.sh --change-all
19/02/2024 15:39:39 INFO: Wazuh API admin credentials not provided, Wazuh API password not changed.
19/02/2024 15:39:52 INFO: The password for user admin is f3nSmfp93fe3Uf7kLvn-RtfvSNM5*T
19/02/2024 15:39:52 INFO: The password for user kibanauser is 20W7uuYx*Vz+NLzUch1iWKFSDyI2l*W
19/02/2024 15:39:52 INFO: The password for user kibanaro is 2jFx4MK8+RnGKH6yH1729jKJCKzQmcN
19/02/2024 15:39:52 INFO: The password for user logstash is Ep8rxELNxJ41sP2?MQ1E0c2buvxX+
19/02/2024 15:39:52 INFO: The password for user readall is hPVlqJZcpMzyo*NTsjc5726Acpg15C0
19/02/2024 15:39:52 INFO: The password for user snapshotrestorer is +PbOZ2tIvyk8V02SR41oLE4_Zr4sgH
19/02/2024 15:39:52 WARNING: Wazuh indexer passwords changed. Remember to update the password in the Wazuh dashboard and Filebeat nodes if necessary, and restart the services.
[root@parrot :~]
```

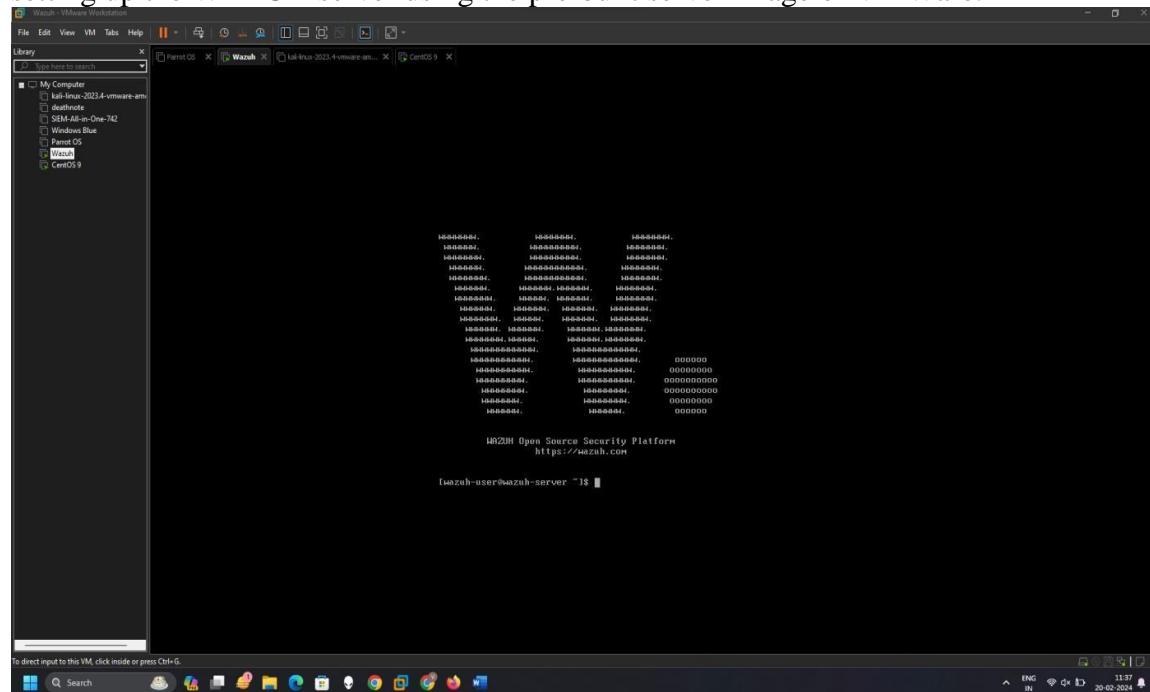
### 2. Change Wazuh Dashboard Password:

On your Wazuh dashboard node, run the following command to update the kibanaserver password in the Wazuh dashboard keystore. Replace <kibanaserver-password> with the random password generated in the first step. echo <kibanaserver-password> | /usr/share/wazuh-dashboard/bin/opensearchdashboards-keystore --allow-root add -f --stdin opensearch.password

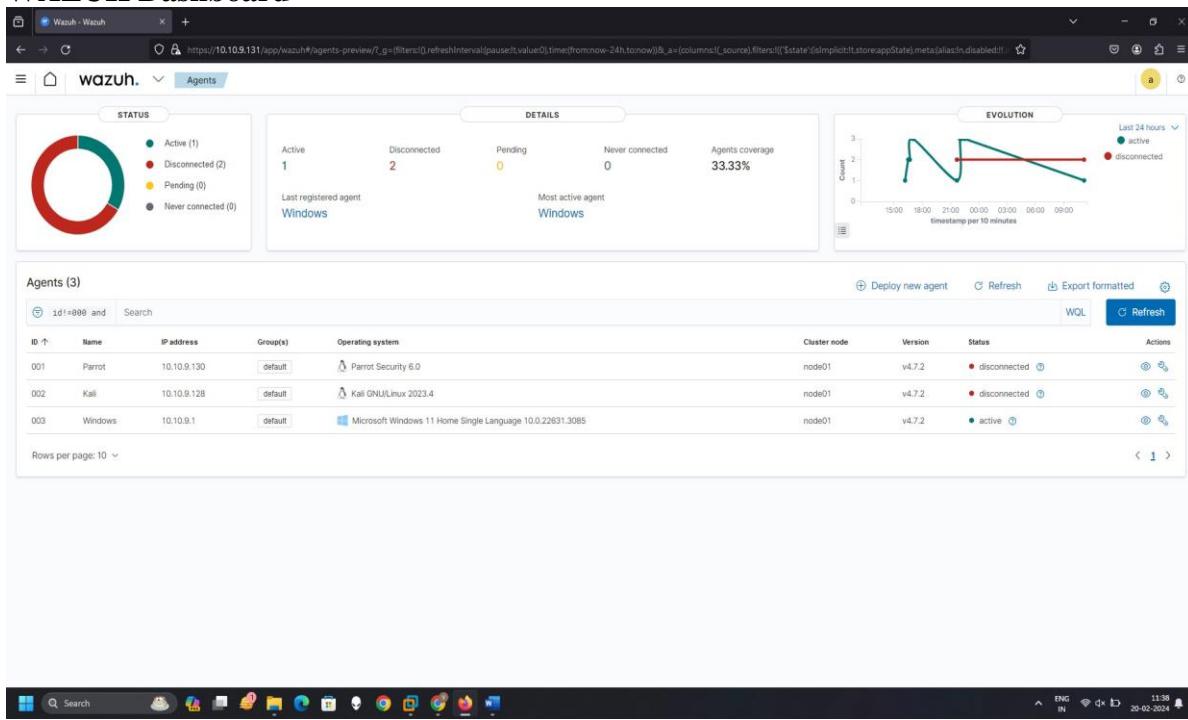
```
[root@parrot :~]# /etc/wazuh-indexer/certs
[...]
# echo 20W7uuYx*Vz+NLzUch1iWKFSDyI2l*W | /usr/share/wazuh-dashboard/bin/opensearchdashboards-keystore --allow-root add -f --stdin opensearch.password
v16.20.0
```

### 3. Restart the service systemctl restart wazuh-dashboard

Got some connection error for dashboard tried to troubleshoot it but not getting the solution so setting up the WAZUH server using the pre-built server image of VMWare.



## WAZUH Dashboard –



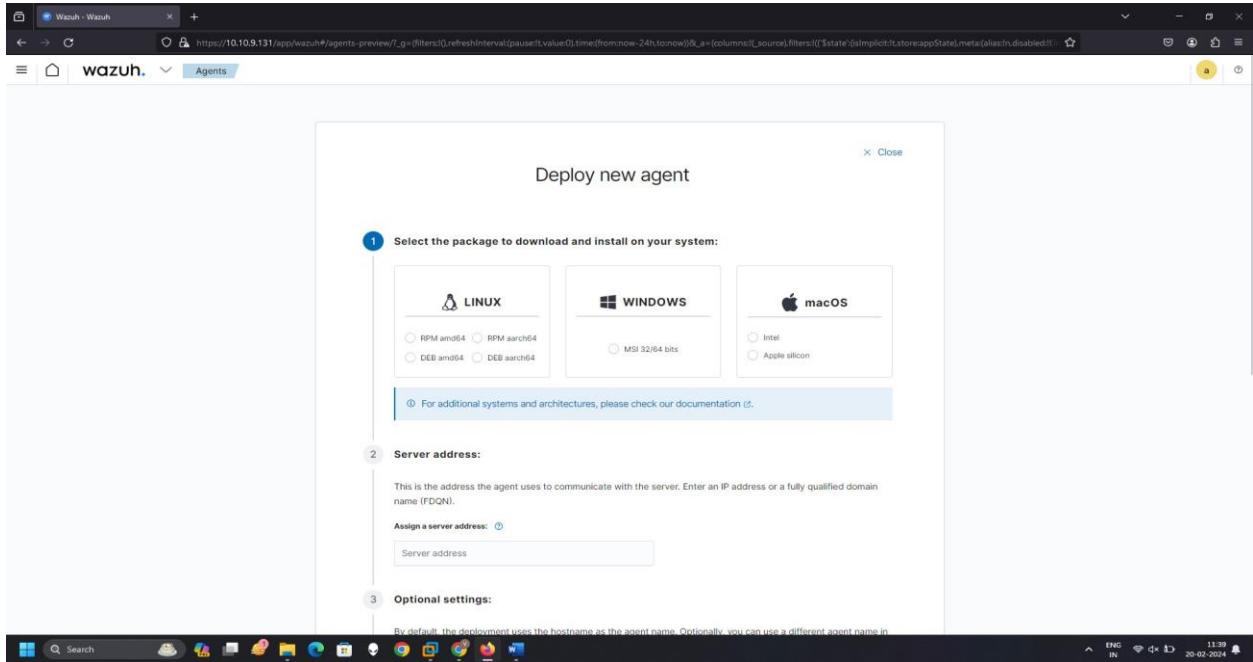
## Adding Windows Agent –



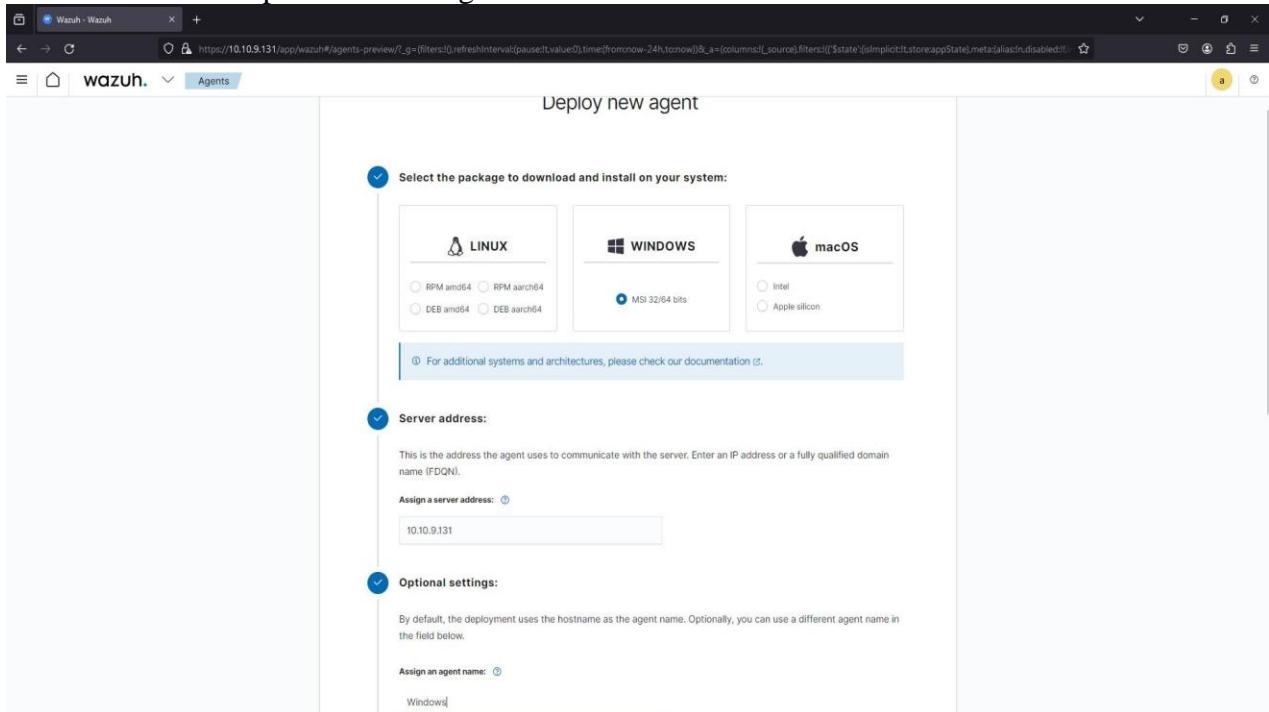
Goto -

- + Deploy new agent

## Select the source –



## Add Wazuh server ip address and agent name –



Run the following commands in windows powershell.

**4 Run the following commands to download and install the agent:**

```
Invoke-WebRequest -Uri https://packages.wazuh.com/4.x/windows/wazuh-agent-4.7.2-1.msi -OutFile ${env.tmp}\wazuh-agent; msieexec.exe /i ${env.tmp}\wazuh-agent /q WAZUH_MANAGER='10.10.9.131' WAZUH_AGENT_GROUP='default' WAZUH_AGENT_NAME='Windows' WAZUH_REGISTRATION_SERVER='10.10.9.131'
```

**① Requirements**

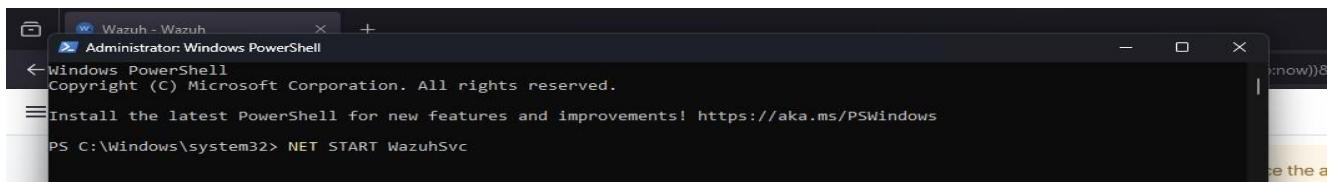
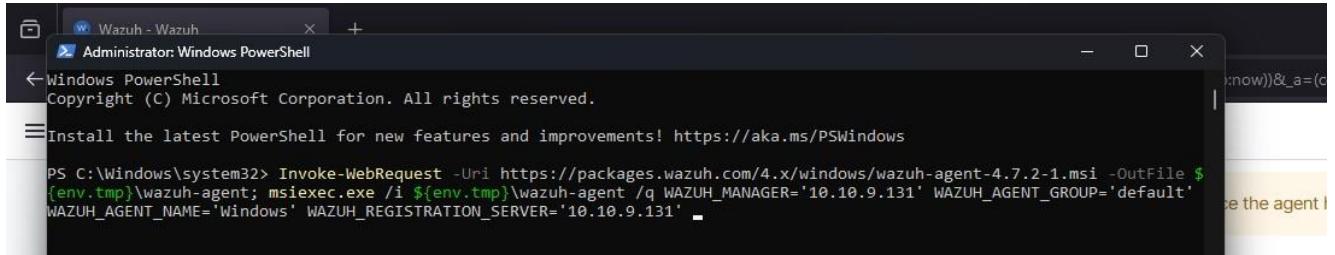
- You will need administrator privileges to perform this installation.
- PowerShell 3.0 or greater is required.

Keep in mind you need to run this command in a Windows PowerShell terminal.

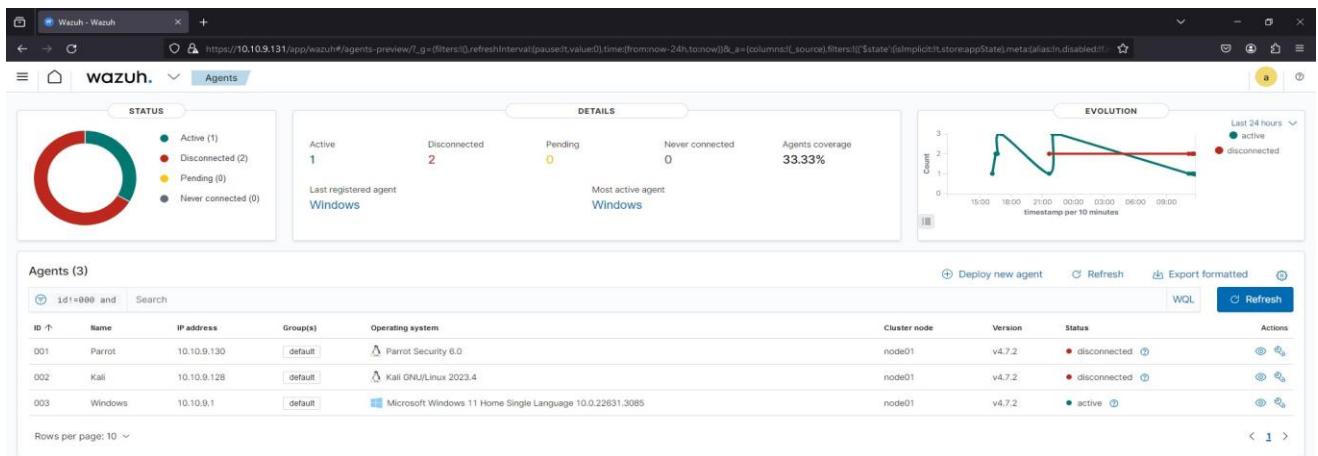
**5 Start the agent:**

```
NET START WazuhSvc
```

Run the command –



Agent added Successfully –



Windows agent added and status is active.

## Adding Ubuntu agent –

The screenshot shows the "Deploy new agent" form:

- Select the package to download and install on your system:**
  - LINUX:** RPM amd64, RPM aarch64, DEB amd64, DEB aarch64 (DEB amd64 is selected).
  - WINDOWS:** MSI 32/64 bits (MSI 32/64 bits is selected).
  - macOS:** Intel, Apple silicon (Apple silicon is selected).

For additional systems and architectures, please check our documentation.
- Server address:**  
This is the address the agent uses to communicate with the server. Enter an IP address or a fully qualified domain name (FQDN).  
Assign a server address:
- Optional settings:**  
By default, the deployment uses the hostname as the agent name. Optionally, you can use a different agent name in the field below.  
Assign an agent name:   
The agent name must be unique. It can't be changed once the agent has been enrolled.



### Run the following commands to download and install the agent:

```
wget https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.7.2-1_amd64.deb &&
sudo WAZUH_MANAGER='10.10.9.131' WAZUH_AGENT_GROUP='linux' WAZUH_AGENT_NAME='Ubuntu' dpkg -i ./wazuh-agent_4.7.2-1_amd64.deb
```

#### ① Requirements

- You will need administrator privileges to perform this installation.
- Shell Bash is required.

Keep in mind you need to run this command in a Shell Bash terminal.

5

### Start the agent:

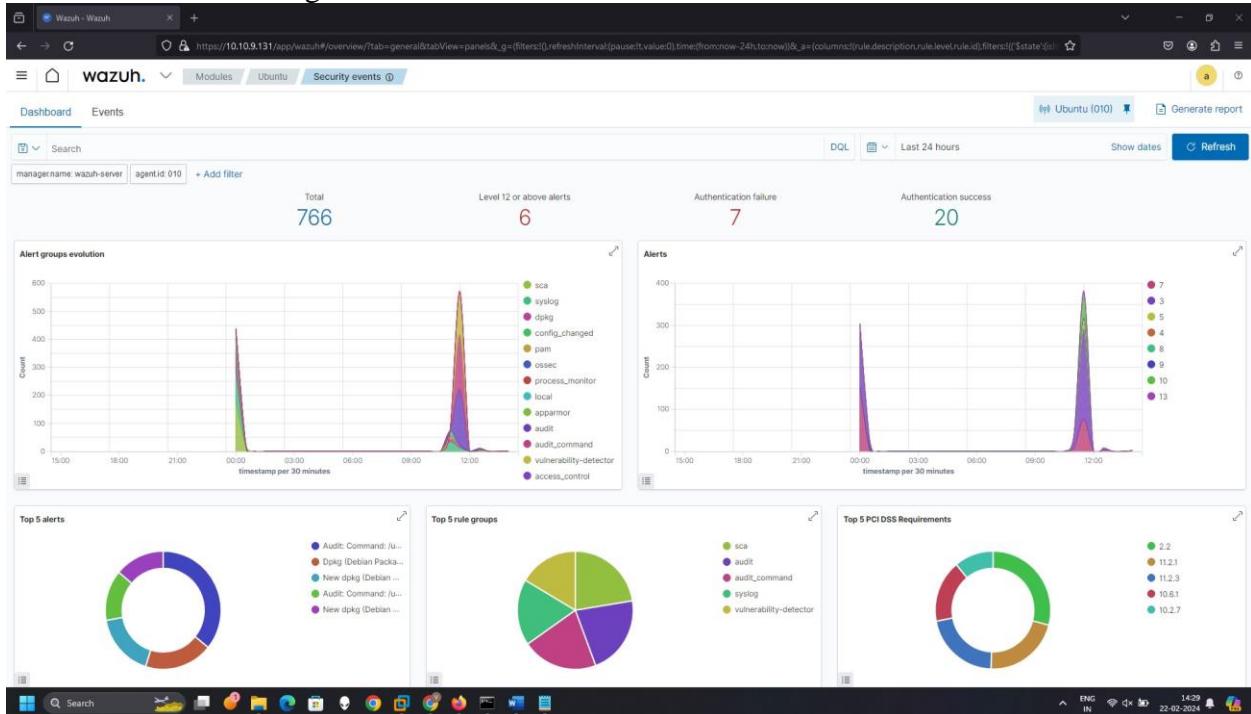
```
sudo systemctl daemon-reload
sudo systemctl enable wazuh-agent
sudo systemctl start wazuh-agent
```

Running the following commands in ubuntu.

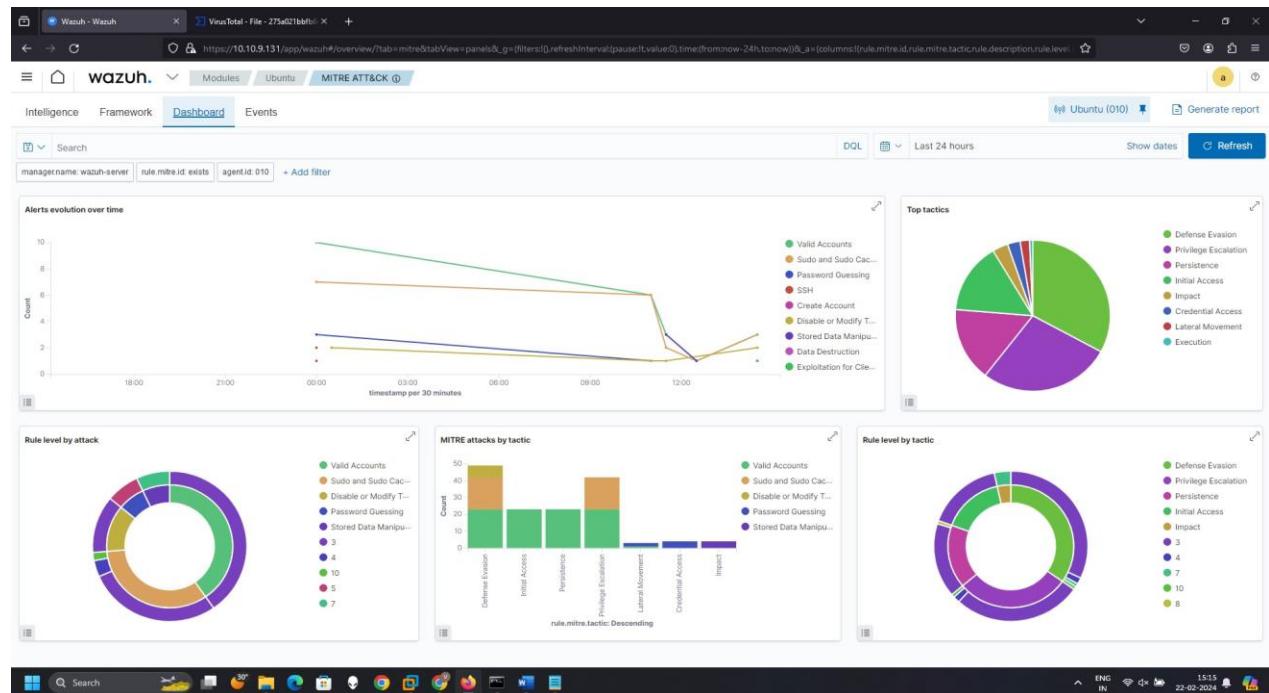
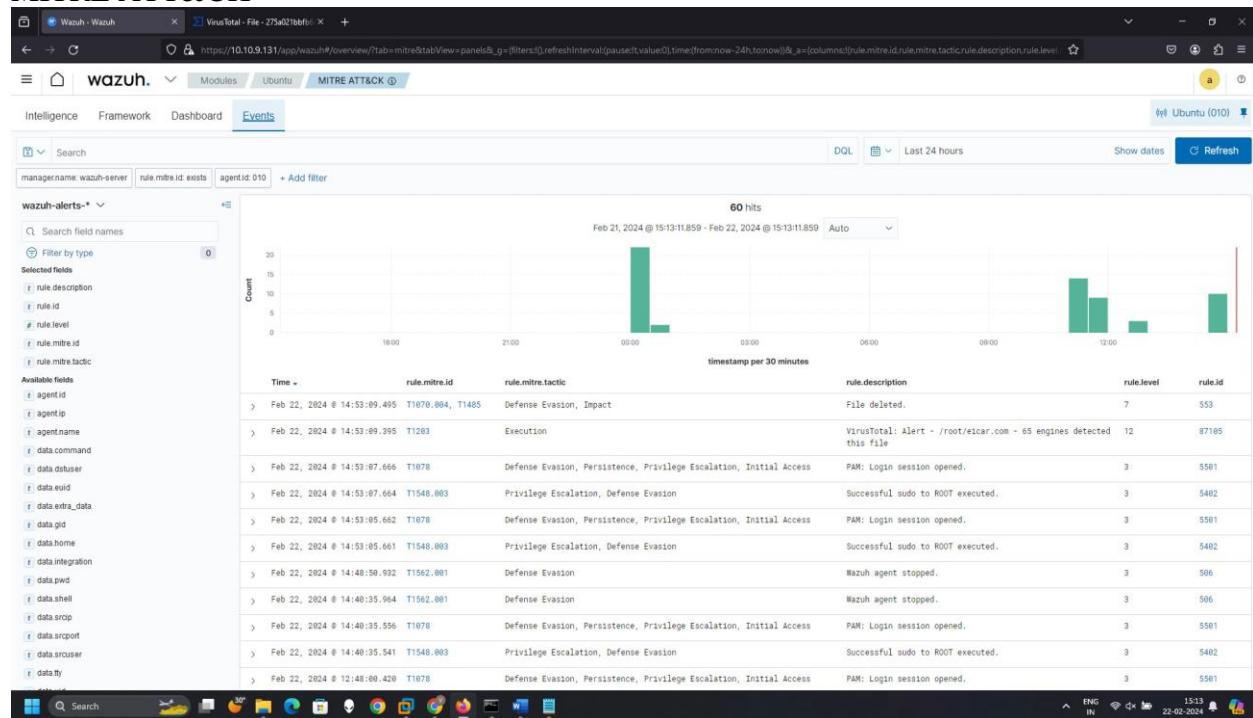
Agent Added –

009	Windows	10.10.9.1	windows	Microsoft Windows 11 Home Single Language 10.0.22831.3155	node01	v4.7.2	active		
010	Ubuntu	10.10.9.133	linux	Ubuntu 22.04.4 LTS	node01	v4.7.2	active		
Rows per page: 10									

## Looking at the basic logs – Dashboard of Ubuntu agent



## MITRE ATT&CK



## User login log

Feb 22, 2024 @ 11:27:01.934	PAM: Login session opened.
3	5501
<a href="#">View surrounding documents</a>	<a href="#">View single document</a>
<a href="#">Expanded document</a>	
<a href="#">Table</a>	<a href="#">JSON</a>
<pre>t _index           wazuh-alerts-4.x-2024.02.22 t agent.id         010 t agent.ip          10.10.9.133 t agent.name        Ubuntu t data.dstuser      root(uid=0) t data.srcuser       ubuntu t data.uid          0 t decoder.name      pam t decoder.parent    pam t full_log          Feb 22 11:27:00 ubuntu sudo: pam_unix(sudo:session): session opened for user root(uid=0) by ubuntu(uid=0) t id                1708581421.328256 t input.type        log t location          /var/log/auth.log t manager.name      wazuh-server t predecoder.hostname  ubuntu t predecoder.program_name sudo t predecoder.timestamp  Feb 22 11:27:00 t rule.description   PAM: Login session opened. # rule.firedtimes    1</pre>	

## Root user login log using sudo command

Feb 22, 2024 @ 12:48:00.419	Successful sudo to ROOT executed.
3	5402
<a href="#">View surrounding documents</a>	<a href="#">View single document</a>
<a href="#">Expanded document</a>	
<a href="#">Table</a>	<a href="#">JSON</a>
<pre>t _index           wazuh-alerts-4.x-2024.02.22 t agent.id         010 t agent.ip          10.10.9.133 t agent.name        Ubuntu t data.command      /bin/bash t data.dstuser      root t data.pwd          /root t data.srcuser       ubuntu t data.tty          pts/2 t decoder.ftacomment First time user executed the sudo command t decoder.name      sudo t decoder.parent    sudo t full_log          Feb 22 12:48:18 ubuntu sudo:  ubuntu : TTY=pts/2 ; PWD=/root ; USER=root ; COMMAND=/bin/bash t id                1708586280.1833984 t input.type        log t location          /var/log/auth.log t manager.name      wazuh-server t predecoder.hostname  ubuntu</pre>	

## New package installation

Feb 22, 2024 @ 11:40:48.582	New dpkg (Debian Package) installed.	7	2982
<a href="#">Expanded document</a>		<a href="#">View surrounding documents</a>	<a href="#">View single document</a>
<a href="#">Table</a> <a href="#">JSON</a>			
<pre>t _index           wazuh-alerts-4.x-2024.02.22 t agent.id        010 t agent.ip         10.10.9.133 t agent.name       Ubuntu t data.arch        amd64 t data.dpkg_status status installed t data.package     hping3 t data.version     3.a2.ds2-10 t decoder.name    dpkg-decoder t full_log         2024-02-22 11:40:48 status installed hping3:amd64 3.a2.ds2-10 t id               1708582248.614645 t input.type       log t location         /var/log/dpkg.log t manager.name     wazuh-server t rule.description New dpkg (Debian Package) installed. # rule.firetimes   2 t rule.gdpr        IV_35.7.d t rule.gpg13       4.10</pre>			

## Failed sudo user login –

Feb 22, 2024 @ 11:13:12.757	Three failed attempts to run sudo	10	5404
<a href="#">Expanded document</a>		<a href="#">View surrounding documents</a>	<a href="#">View single document</a>
<a href="#">Table</a> <a href="#">JSON</a>			
<pre>t _index           wazuh-alerts-4.x-2024.02.22 t agent.id        010 t agent.ip         10.10.9.133 t agent.name       Ubuntu t data.command     /bin/bash t data.dstuser     root t data.pwd         /root t data.srcuser     ubuntu t data.tty          pts/0 t decoder.ftcomment First time user executed the sudo command t decoder.name     sudo t decoder.parent   sudo t full_log         Feb 22 11:13:11 ubuntu sudo:  ubuntu : 3 incorrect password attempts ; TTY=pts/0 ; PWD=/root ; USER=root ; COMMAND=/bin/bash t id               1708580592.156941 t input.type       log t location         /var/log/auth.log t manager.name     wazuh-server t predecoder.hostname  ubuntu t predecoder.program_name sudo</pre>			

## Vulnerabilities

The screenshot shows the Wazuh Vulnerabilities dashboard. At the top, there are three main sections: a donut chart of severity (Critical: 5, High: 60, Medium: 66, Low: 22), a summary of scan details (Last full scan: Feb 22, 2024 @ 11:51:37.000, Last partial scan: Feb 22, 2024 @ 14:22:26.000), and another donut chart of package names. Below these is a table titled 'Vulnerabilities (153)' with columns for Name, Version, Architecture, Severity, CVE, CVSS2 Score, CVSS3 Score, and Detection Time. The table lists various packages like apparmor, apport, apport-gtk, bluez-cups, bluez-obexd, bsdutils, etc., with their respective details. The bottom of the screen shows a Windows taskbar with icons for File Integrity monitoring.

## File Integrity monitoring

The screenshot shows the Wazuh File Integrity monitoring dashboard. It features a search bar and filter options (managername: wazuh-server, rule groups: syscheck, agentid: 010). The main area displays a histogram titled '4 hits' showing the count of events per 30 minutes from Feb 21, 2024 to Feb 22, 2024. Below the histogram is a table of audit logs with columns: Time, syscheck.path, syscheck.event, rule.description, rule.level, and rule.id. The table shows several entries for modified files like /root/.bash\_history and /root/.lessht. The bottom of the screen shows a Windows taskbar with icons for Vulnerabilities.

## **CHAPTER: 7 INTEGRATING VIRUSTOTAL**

## CHAPTER: 7 INTEGRATING VIRUSTOTAL

Search for the <syscheck> block in the Wazuh agent configuration file /var/ossec/etc/ossec.conf. Make sure that <disabled> is set to no. This enables the Wazuh FIM to monitor for directory changes.

Add an entry within the <syscheck> block to configure a directory to be monitored in near real-time. In this case, you are monitoring the /root directory:

```
<directories realtime="yes">/root</directories>
```

```
GNU nano 6.2                               /var/ossec/etc/ossec.conf
<!-->
<enabled>yes</enabled>
<scan_on_start>yes</scan_on_start>
<interval>12h</interval>
<sk1p_nf>yes</sk1p_nf>
</scs>
<!-- File integrity monitoring -->
<syscheck>
<disabled>no</disabled>
<!-- Frequency that syscheck is executed default every 12 hours -->
<frequency>43200</frequency>
<scan_on_start>yes</scan_on_start>
<!-- Directories to check (perform all possible verifications) -->
<directories>/etc,/usr/bin,/usr/sbin</directories>
<directories>/bin,/sbin,/boot</directories>
<directories realtime="yes">/root</directories>
```

Install jq, a utility that processes JSON input from the active response script.  
sudo apt update sudo apt -y install jq

```
root@ubuntu:~# apt -y install jq
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libjq1 libbong5
The following NEW packages will be installed:
  jq libjq1 libbong5
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 357 kB of archives.
After this operation, 1,087 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libbong5 amd64 6.9.7.1-2build1 [172 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libjq1 amd64 1.6-2.1ubuntu3 [133 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 jq amd64 1.6-2.1ubuntu3 [52.5 kB]
Fetched 357 kB in 1s (299 kB/s)
Selecting previously unselected package libbong5:amd64.
(Reading database ... 203534 files and directories currently installed.)
Preparing to unpack .../libbong5_6.9.7.1-2build1_amd64.deb ...
Unpacking libbong5:amd64 (6.9.7.1-2build1) ...
Selecting previously unselected package libjq1:amd64.
Preparing to unpack .../libjq1_1.6-2.1ubuntu3_amd64.deb ...
Unpacking libjq1:amd64 (1.6-2.1ubuntu3) ...
Selecting previously unselected package jq.
Preparing to unpack .../jq_1.6-2.1ubuntu3_amd64.deb ...
Unpacking jq (1.6-2.1ubuntu3) ...
Setting up libbong5:amd64 (6.9.7.1-2build1) ...
Setting up libjq1:amd64 (1.6-2.1ubuntu3) ...
Setting up jq (1.6-2.1ubuntu3) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
root@ubuntu:~#
```

Create the /var/ossec/active-response/bin/remove-threat.sh active response script to remove malicious files from the endpoint:

```

#!/bin/bash
# Local variables
LOCAL=$(dirname $0)
cd $LOCAL
cd ..
PWD=$(pwd)

read INPUT_JSON
FILENAME=$(echo $INPUT_JSON | jq -r .parameters.alert.data.virustotal.source.file)
COMMAND=$(echo $INPUT_JSON | jq -r .command)
LOG_FILE=${PWD}../logs/active-responses.log

#----- Analyze command -----
if [ ${COMMAND} = "add" ]
then
    # Send control message to execd
    printf '{"version":1,"origin":{"name":"remove-threat","module":"active-response"},"command":"check_keys", "parameters":{"keys":[]}}\n'
    read RESPONSE
    COMMAND2=$(echo $RESPONSE | jq -r .command)
    if [ ${COMMAND2} != "continue" ]
    then
        echo "date "+%Y/%m/%d %H:%M:%S" $0: $INPUT_JSON Remove threat active response aborted" >> ${LOG_FILE}
        exit 0;
    fi
fi

# Removing file
rm -f $FILENAME
if [ $? -eq 0 ]; then
    echo "date "+%Y/%m/%d %H:%M:%S" $0: $INPUT_JSON Successfully removed threat" >> ${LOG_FILE}
else
    echo "date "+%Y/%m/%d %H:%M:%S" $0: $INPUT_JSON Error removing threat" >> ${LOG_FILE}
fi

exit 0;

```

Change the /var/ossec/active-response/bin/remove-threat.sh file ownership, and permissions:  
 sudo chmod 750 /var/ossec/active-response/bin/remove-threat.sh sudo chown root:wazuh /var/ossec/active-response/bin/remove-threat.sh

```

root@ubuntu:~# chmod 750 /var/ossec/active-response/bin/remove-threat.sh
root@ubuntu:~# chown root:wazuh /var/ossec/active-response/bin/remove-threat.sh
root@ubuntu:~#

```

Restart the Wazuh agent to apply the changes:

```

sudo systemctl restart wazuh-agent

```

## Wazuh server

Perform the following steps on the Wazuh server to alert for changes in the endpoint directory and enable the VirusTotal integration. These steps also enable and trigger the active response script whenever a suspicious file is detected.

Add the following rules to the /var/ossec/etc/rules/local\_rules.xml file on the Wazuh server.

These rules alert about changes in the /root directory that are detected by FIM scans:

```

<group name="syscheck,pci_dss_11.5,nist_800_53_SI.7,>
  <!-- Rules for Linux systems -->
  <rule id="100200" level="7">
    <if_sid>550</if_sid>
    <field name="file">/root</field>
    <description>File modified in /root directory.</description>
  </rule>
  <rule id="100201" level="7">
    <if_sid>554</if_sid>

```

```

<field name="file">/root</field>
<description>File added to /root directory.</description>
</rule>
</group>

<group name="syscheck,pci_dss_11.5,nist_800_53_SI.7,">
    <!-- Rules for Linux systems -->
    <rule id="100200" level="7">
        <if_sid>550</if_sid>
        <field name="file">/root</field>
        <description>File modified in /root directory.</description>
    </rule>
    <rule id="100201" level="7">
        <if_sid>554</if_sid>
        <field name="file">/root</field>
        <description>File added to /root directory.</description>
    </rule>
</group>

```

Add the following configuration to the Wazuh server /var/ossec/etc/ossec.conf file to enable the VirusTotal integration. Replace <YOUR\_VIRUS\_TOTAL\_API\_KEY> with your VirusTotal API key. This allows to trigger a VirusTotal query whenever any of the rules 100200 and 100201 are triggered:

```

<ossec_config>
    <integration>
        <name>virustotal</name>
        <api_key><YOUR_VIRUS_TOTAL_API_KEY></api_key> <!-- Replace with your
VirusTotal API key -->
        <rule_id>100200,100201</rule_id>
        <alert_format>json</alert_format>
    </integration>
</ossec_config>
<ossec_config>
    <integration>
        <name>virustotal</name>
        <api_key>196a8d525e15c1f01d4a397272af2076a8803ed0591dc7a72c9c9c6ee4211c00</api_key> <!-- Replace with your Virus$al:
        <rule_id>100200,100201</rule_id>
        <alert_format>json</alert_format>
    </integration>
</ossec_config>

```

Append the following blocks to the Wazuh server /var/ossec/etc/ossec.conf file. This enables active response and triggers the remove-threat.sh script when VirusTotal flags a file as malicious:

```

<ossec_config>
    <command>
        <name>remove-threat</name>
        <executable>remove-threat.sh</executable>    <timeout_allowed>no</timeout_allowed>
    </command>

```

```

<active-response>
<disabled>no</disabled>
<command>remove-threat</command>
<location>local</location>
<rules_id>87105</rules_id>
</active-response>
</ossec_config>
</ossec_config>
<command>
  <name>remove-threat</name>
  <executable>remove-threat.sh</executable>
  <timeout_allowed>no</timeout_allowed>
</command>

<active-response>
<disabled>no</disabled>
<command>remove-threat</command>
<location>local</location>
<rules_id>87105</rules_id>
</active-response>
</ossec_config>-

```

Add the following rules to the Wazuh server /var/ossec/etc/rules/local\_rules.xml file to alert about the active response results:

```

<group name="virustotal,">
<rule id="100092" level="12">
<if_sid>657</if_sid>
<match>Successfully removed threat</match>
<description>$(parameters.program) removed threat located at
$(parameters.alert.data.virustotal.source.file)</description>
</rule>

<rule id="100093" level="12">
<if_sid>657</if_sid>
<match>Error removing threat</match>
<description>Error removing threat located at
$(parameters.alert.data.virustotal.source.file)</description>
</rule>
</group>

```

```

<group name="virustotal,">
<rule id="100092" level="12">
<if_sid>657</if_sid>
<match>Successfully removed threat</match>
<description>$(parameters.program) removed threat located at $(parameters.alert.data.virustotal.source.file)</description>
</rule>

<rule id="100093" level="12">
<if_sid>657</if_sid>
<match>Error removing threat</match>
<description>Error removing threat located at $(parameters.alert.data.virustotal.source.file)</description>
</rule>
</group>-

```

Restart the Wazuh manager to apply the configuration changes:

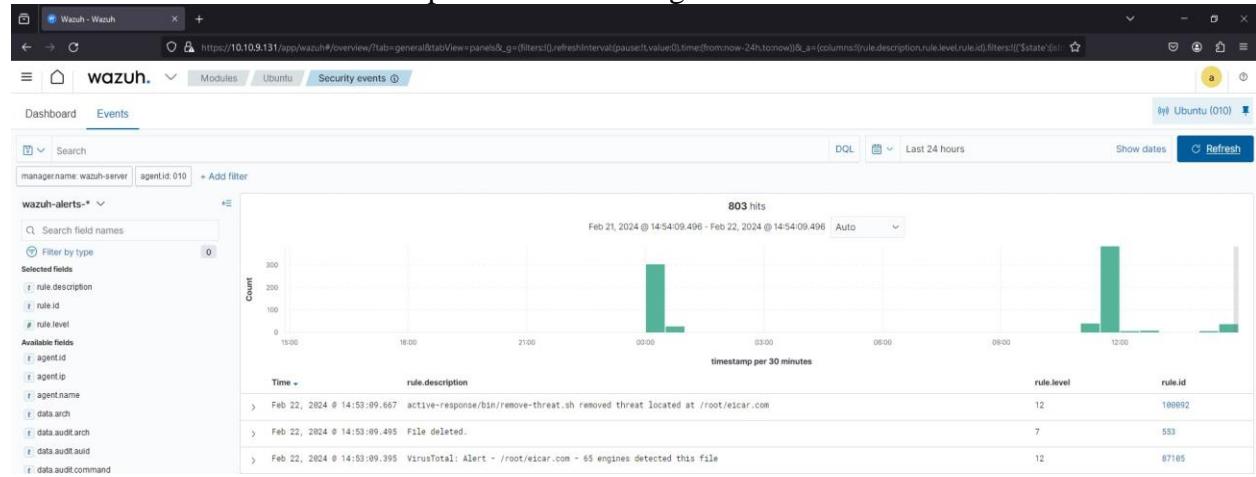
```
sudo systemctl restart wazuh-manager
```

```
[root@wazuh-server ~]# systemctl restart wazuh-manager
[root@wazuh-server ~]#
```

## Attack emulation

Download an EICAR test file to the /root directory on the Ubuntu endpoint:

```
sudo curl -Lo /root/eicar.com https://secure.eicar.org/eicar.com && sudo ls -lah /root/eicar.com
```



## **CHAPTER: 8 WAZUH POC**

## CHAPTER: 8 WAZUH POC

### → File Integrity monitoring –

Creating rule file –



```
< fin_win_test.xml >
Ruleset Test Save
1 <!-- Modify it at your will. -->
2 <group name="syscheck">
3   <rule id="100003" level="8">
4     <if_sid>553</if_sid>
5     <field name="ossec.name">explorer.exe$</field>
6     <field name="user">DESKTOP-7V9EV5I$</field>
7     <match>deleted</match>
8     <description>The user "${uname}" deleted a monitored file with File Explorer</description>
9     <mitre>
10    <id>T1070.0.04</id>
11    <id>T1485</id>
12  </mitre>
13 </rule>
14 </group>
```

Add this directory path to the windows agent ossec.conf file –

```
215 <syscheck>
216   <directories whodata="yes">C:\test</directories>
217 </syscheck>
218
219
```

Testing the rule –

Create a folder in name test in C:\ -

Add any file an delete it –

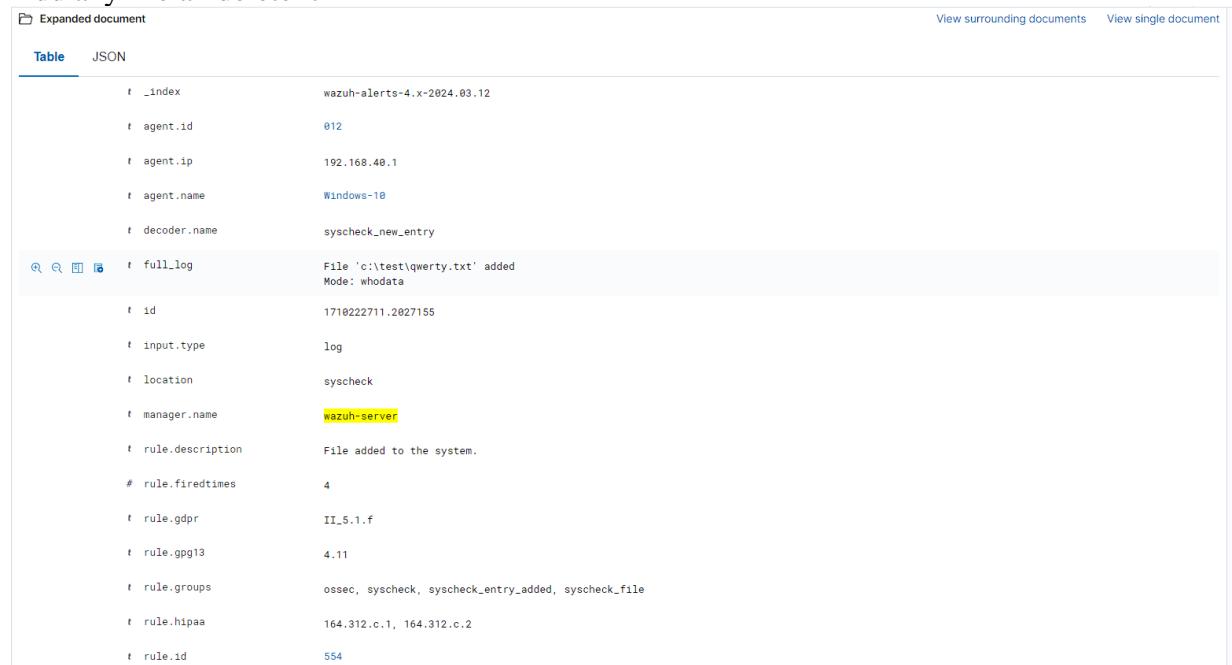


Table		JSON	View surrounding documents	View single document
t	_index	wazuh-alerts-4.x-2024.03.12		
t	agent.id	012		
t	agent.ip	192.168.48.1		
t	agent.name	Windows-10		
t	decoder.name	syscheck_new_entry		
t	full_log	File 'c:\test\qwerty.txt' added Mode: whodata		
t	id	1710222711.2027155		
t	input.type	log		
t	location	syscheck		
t	manager.name	wazuh-server		
t	rule.description	File added to the system.		
#	rule.firetimes	4		
t	rule.gdpr	II_5.1.f		
t	rule.gpg13	4.11		
t	rule.groups	ossec, syscheck, syscheck_entry_added, syscheck_file		
t	rule.hipaa	164.312.c.1, 164.312.c.2		
t	rule.id	554		

Table JSON	
t _index	wazuh-alerts-4.x-2024.03.12
t agent.id	012
t agent.ip	192.168.40.1
t agent.name	Windows-10
t decoder.name	syscheck_deleted
t full_log	File 'c:\test\new text document.txt' deleted Mode: whodata
t id	1710222711.2028565
t input.type	log
t location	syscheck
t manager.name	wazuh-server
t rule.description	File deleted.
# rule.firedtimes	3
t rule.gdpr	II_5.1.f
t rule.gpg13	4.11
t rule.groups	ossec, syscheck, syscheck_entry_deleted, syscheck_file
t rule.hipaa	164.312.c.1, 164.312.c.2
t rule.id	553
# rule.level	7

## ➔ Detecting Malware Persistence –

Add this the configuration file of wazuh agent – ossec.conf –

```
[220] <syscheck>
[221]   <directories realtime="yes">%PROGRAMDATA%\Microsoft\Windows\Start Menu\Programs\Startup</directories>
[222]
[223] </syscheck>
```

Testing the configuration –

Employ PowerShell to acquire the EICAR test file and save it to the

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup folder.

Run this in powershell –

▼ Mar 12, 2024 @ 12:10:56.172 c:\programdata\microsoft\windows\start menu\programs\startup\eicar.com	added	File added to the system	5	554
<a href="#">View surrounding documents</a> <a href="#">View single document</a>				
<a href="#">Expanded document</a>				
<a href="#">Table JSON</a>				
t _index				wazuh-alerts-4.x-2024.03.12
t agent.id				012
t agent.ip				192.168.40.1
t agent.name				Windows-10
t decoder.name				syscheck_new_entry
t full_log				File 'c:\programdata\microsoft\windows\start menu\programs\startup\eicar.com' added Mode: realtime
t id				1710225656.2252907
t input.type				log
t location				syscheck
t manager.name				wazuh-server
t rule.description				File added to the system.
# rule.firedtimes				2
t rule.gdpr				II_5.1.f
t rule.gpg13				4.11

## Now delete the file from endpoint –

Mar 12, 2024 @ 12:11:58.105 c:\programdata\microsoft\windows\start menu\programs\startup\elcar.com		deleted	File deleted.	7	553
Expanded document					<a href="#">View surrounding documents</a> <a href="#">View single document</a>
<b>Table</b> JSON					
<pre>t _index wazuh-alerts-4.x-2024.03.12 t agent.id 012 t agent.ip 192.168.40.1 t agent.name Windows-10 t decoder.name syscheck_deleted t full_log File 'c:\programdata\microsoft\windows\start menu\programs\startup\elcar.com' deleted Mode: realtime t id 1710225718.2255639 t input.type log t location syscheck t manager.name wazuh-server t rule.description File deleted. # rule.firedtimes 2 t rule.gdpr II_5.1.f t rule.gpg13 4.11 t rule.groups ossec, syscheck, syscheck_entry_deleted, syscheck_file</pre>					
t _index wazuh-alerts-4.x-2024.03.12 t agent.id 012 t agent.ip 192.168.40.1 t agent.name Windows-10 t decoder.name syscheck_deleted t full_log File 'c:\programdata\microsoft\windows\start menu\programs\startup\elcar.com' deleted Mode: realtime t id 1710225718.2255639 t input.type log t location syscheck t manager.name wazuh-server t rule.description File deleted. # rule.firedtimes 2 t rule.gdpr II_5.1.f t rule.gpg13 4.11 t rule.groups ossec, syscheck, syscheck_entry_deleted, syscheck_file	deleted	File deleted.	7	553	

→ Detecting malware persistence in windows registry –  
Edit the wazuh config file from wazuh agent (ossec.conf) –

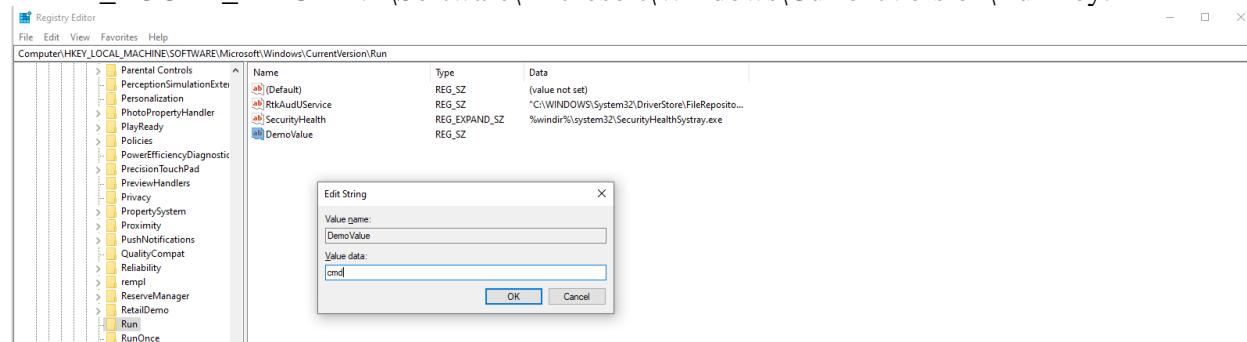
```
223
224   <syscheck>
225     <frequency>300</frequency>
226     <windows_registry arch="both">HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run</windows_registry>
227     <windows_registry arch="both">HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce</windows_registry>
228 </syscheck>
```

Restart the Wazuh Agent –

```
PS C:\WINDOWS\system32> Restart-Service -Name wazuh
PS C:\WINDOWS\system32>
```

Test the configuration –

Add the registry value name DemoValue and registry value data cmd to the HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run key.



Mar 12, 2024 @ 12:25:41.399 Registry Value Entry Added to the System

5 752

View surrounding documents View single document

Expanded document

Table JSON

t _index	wazuh-alerts-4.x-2024.03.12
t agent.id	012
t agent.ip	192.168.40.1
t agent.name	Windows-10
t decoder.name	syscheck_registry_value_added
t full_log	Registry Value '[x64] HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\DemoValue' added Mode: scheduled
t id	1710226541.2353524
t input.type	log
t location	syscheck
t manager.name	wazuh-server
t rule.description	Registry Value Entry Added to the System
# rule.firedtimes	1
t rule.gdpr	II_5.1.f
t rule.gpg13	4.13
t rule.groups	ossec, syscheck, syscheck_entry_added, syscheck_registry
t rule.hipaa	164.312.c.1, 164.312.c.2

Time	syscheck.path	syscheck.event	rule.description	rule.level	rule.id
> Mar 12, 2024 @ 12:25:41.399	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run	added	Registry Value Entry Added to the System	5	752
> Mar 12, 2024 @ 12:25:41.398	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run	modified	Registry Key Integrity Checksum Changed	5	594

## → Notepad –

Create a bat file – tasklist.bat in C Drive –



```

1 | Echo Off
2 | setlocal enableDelayedExpansion
3 | for /f "delims=" %%a in ('powershell -command "& tasklist"') do (
4 |   echo tasklist: %%a
5 |
6 | exit /B

```

## Edit ossec.conf –

```

231 | <ossec_config>
232 |   <wodle name="command">
233 |     <disabled/no></disabled>
234 |     <tag>tasklist</tag>
235 |     <command>PowerShell.exe C:\tasklist.bat</command>
236 |     <interval>2m</interval>
237 |     <run_on_start>yes</run_on_start>
238 |     <timeout>10</timeout>
239 |   </wodle>
240 |
241 | </ossec_config>
242
243

```

## Edit this file in Wazuh Server –

Nano /var/ossec/etc/decoders/local\_decoder.xml –

```

<decoder name="tasklist">
  <prematch>^tasklist: </prematch>
</decoder>

```

## Nano /var/ossec/etc/rules/local\_rules.xml –

```
<group name="process_monitor">
  <rule id="100010" level="6">
    <decoded_as>tasklist</decoded_as>
    <regex type="pcre2">(?i)notepad.exe</regex>
    <description>Notepad.exe is running.</description>
  </rule>
</group>
```

Mar 12, 2024 0 14:13:00.243 Notepad.exe is running.		6	100010
Expanded document		View surrounding documents	View single document
Table	JSON		
t _index	wazuh-alerts-4.x-2024.03.12		
t agent.id	012		
t agent.ip	192.168.40.1		
t agent.name	Windows-10		
t decoder.name	tasklist		
t full_log	tasklist: notepad.exe	6788 Console	1 15,124 K
t id	1710232980.2724268		
t input.type	log		
t location	command_tasklist		
t manager.name	wazuh-server		
t rule.description	Notepad.exe is running.		
# rule.firedtimes	1		
t rule.groups	process_monitor		
t rule.id	100010		
# rule.level	6		
(rule.mail	true		
timestamp	Mar 12, 2024 0 14:13:00.243		

## → CMD –

```
60
61 <group name="process_monitor,>
62 <rule id="100021" level="6">
63   <decoded_as>tasklist</decoded_as>
64   <regex type="pcre2">(?!cmd.exe)</regex>
65   <description>cmd.exe is running.</description>
66 </rule>
67 </group>
68
```

Edit the local\_rules.xml file and add the above code.

Alert -

> Mar 12, 2024 @ 15:13:23.219 cmd.exe is running.	6	100021
▼ Mar 12, 2024 @ 15:13:22.971 cmd.exe is running.	6	100021

View surrounding documents View single document

Table JSON

t _index	wazuh-alerts-4.x-2024.03.12
t agent.id	012
t agent.ip	192.168.40.1
t agent.name	Windows-10
t decoder.name	tasklist
t full_log	tasklist: cmd.exe
24924	Console
t id	1710236602.3302300
t input.type	log
t location	command_tasklist
t manager.name	wazuh-server
t rule.description	cmd.exe is running.
# rule.firedtimes	10
t rule.groups	process_monitor
t rule.id	100021
# rule.level	6

## → Logging the command of powershell in wazuh –

Press Windows + R keys on your keyboard to open the run dialog box.

Type gpedit.msc in the search box and click OK to open the local group policy editor.

Navigate to Computer Configuration > Administrative Templates > Windows Components > Windows PowerShell > Turn on PowerShell Script Block Logging.

Setting	State	Comment
Turn on Module Logging	Enabled	No
Turn on PowerShell Script Block Logging	Enabled	No
Turn on Script Execution	Not configured	No
Turn on PowerShell Transcription	Enabled	No
Set the default source path for Update-Help	Not configured	No

Enable the following settings.

Open ossec.conf file in windows agent and append this –

```
<localfile>
<location>Microsoft-Windows-PowerShell/Operational</location>
<log_format>eventchannel</log_format>
</localfile>
```

Restart the wazuh agent –

```
> Restart-Service -Name wazuh
```

Test the configuration –

On the Windows endpoint, run the following command via PowerShell with administrator privileges to add a registry entry NoofAlerts to the HKLM\Software\Microsoft\ADs registry key, and set the value to 2:

```
> New-ItemProperty -Path "HKLM:\Software\Microsoft\ADs" -Name "NoofAlerts" -Value 2

PS C:\WINDOWS\system32> New-ItemProperty -Path "HKLM:\Software\Microsoft\ADs" -Name "NoofAlerts" -Value 2

NoofAlerts    : 2
PSPath       : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\Software\Microsoft\ADs
PSParentPath  : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\Software\Microsoft
PSChildName   : ADs
PSDrive      : HKLM
PSProvider    : Microsoft.PowerShell.Core\Registry

PS C:\WINDOWS\system32> ■
```

▼ Mar 13, 2024 @ 12:01:58.854 Powershell executed "New-ItemProperty -Path". Possible addition of new item to registry

3 91843

View surrounding documents View single document

Expanded document

Table JSON

t _index	wazuh-alerts-4.x-2024.03.13
t agent.id	012
t agent.ip	192.168.40.1
t agent.name	Windows-10
t data.win.eventdata.messageNumber	1
t data.win.eventdata.messageTotal	1
t data.win.eventdata.scriptBlockId	f2805908-4680-4f86-8af5-0ce12fcc8392
t data.win.eventdata.scriptBlockText	New-ItemProperty -Path "HKLM:\Software\Microsoft\ADs" -Name "NoofAlerts" -Value 2
t data.win.system.channel	Microsoft-Windows-PowerShell/Operational
t data.win.system.computer	DESKTOP-7V9EVSI
t data.win.system.eventID	4104
t data.win.system.eventRecordID	3162
t data.win.system.keywords	0x0
t data.win.system.level	5
t data.win.system.message	"Creating Scriptblock text (1 of 1): New-ItemProperty -Path "HKLM:\Software\Microsoft\ADs" -Name "NoofAlerts" -Value 2

## ➔ Windows Defender logs –

Add the following configuration in the ossec.conf file in the windows agent –

```
68  <!-- Windows Defender -->
69  <localfile>
70    <location>Microsoft-Windows-Defender/Operational</location>
71    <log_format>eventchannel</log_format>
72  </localfile>
73
74
```

## Testing the configuration –

Turn off the windows defender.

The screenshot shows a log entry from the Wazuh interface. The log details a real-time protection scan being disabled by Microsoft Defender. The event was recorded on March 13, 2024, at 12:30:37.879. The log message is: "Windows Defender: Antivirus real-time protection is disabled". The event has an ID of 62152. Below the log, there is a table view of the event data in JSON format, which includes fields like \_index, agent.id, agent.ip, agent.name, data.win.eventdata.product\_name, data.win.eventdata.product\_version, data.win.system.channel, data.win.system.computer, data.win.system.eventID, data.win.system.eventRecordID, data.win.system.keywords, data.win.system.level, data.win.system.message, and data.win.system.opcode.

Field	Value
_index	wazuh-alerts-4.x-2024.03.13
agent.id	012
agent.ip	192.168.40.1
agent.name	Windows-10
data.win.eventdata.product_name	Microsoft Defender Antivirus
data.win.eventdata.product_version	4.18.24010.12
data.win.system.channel	Microsoft-Windows-Defender/Operational
data.win.system.computer	DESKTOP-7V9EVSI
data.win.system.eventID	5001
data.win.system.eventRecordID	310
data.win.system.keywords	0x8000000000000000
data.win.system.level	4
data.win.system.message	"Microsoft Defender Antivirus Real-time Protection scanning for malware and other potentially unwanted software was disabled."
data.win.system.opcode	0

## ➔ Monitoring Downloads directory using virustotal for malicious file download and delete–

Add the directory you want to monitor using virustotal in ossec.conf file of agent –

```
<syscheck>
  <directories whodata="yes">C:\test</directories>
<!-- Virustotal integration windows -->
  <directories realtime="yes">C:\Users\vrajp\Downloads</directories>
</syscheck>
```

## Create a python file (remove-threat.py) –

```
#!/usr/bin/python3
```

```
# Copyright (C) 2015-2022, Wazuh Inc.
```

```
# All rights reserved.
```

```
import os
import sys
import json
import datetime

if os.name == 'nt':
    LOG_FILE = "C:\\Program Files (x86)\\ossec-agent\\active-response\\active-responses.log"
else:
```

```

LOG_FILE = "/var/ossec/logs/active-responses.log"

ADD_COMMAND = 0
DELETE_COMMAND = 1
CONTINUE_COMMAND = 2
ABORT_COMMAND = 3

OS_SUCCESS = 0
OS_INVALID = -1

class message:
    def __init__(self):
        self.alert = ""
        self.command = 0

    def write_debug_file(ar_name, msg):
        with open(LOG_FILE, mode="a") as log_file:
            log_file.write(str(datetime.datetime.now().strftime('%Y/%m/%d %H:%M:%S')) + " " +
ar_name + ": " + msg + "\n")

    def setup_and_check_message(argv):
        # get alert from stdin
        input_str = ""
        for line in sys.stdin:
            input_str = line
            break

        try:
            data = json.loads(input_str)
        except ValueError:
            write_debug_file(argv[0], 'Decoding JSON has failed, invalid input format')
            message.command = OS_INVALID
            return message

        message.alert = data
        command = data.get("command")

        if command == "add":
            message.command = ADD_COMMAND
        elif command == "delete":
            message.command = DELETE_COMMAND
        else:
            message.command = OS_INVALID

```

```

        write_debug_file(argv[0], 'Not valid command: ' + command)

    return message

def send_keys_and_check_message(argv, keys):

    # build and send message with keys
    keys_msg = json.dumps({"version": 1,"origin":{"name": argv[0],"module":"active-response"},"command":"check_keys","parameters":{"keys":keys}})

    write_debug_file(argv[0], keys_msg)

    print(keys_msg)
    sys.stdout.flush()

    # read the response of previous message
    input_str = ""
    while True:
        line = sys.stdin.readline()
        if line:
            input_str = line
            break

    # write_debug_file(argv[0], input_str)

    try:
        data = json.loads(input_str)
    except ValueError:
        write_debug_file(argv[0], 'Decoding JSON has failed, invalid input format')
        return message

    action = data.get("command")

    if "continue" == action:
        ret = CONTINUE_COMMAND
    elif "abort" == action:
        ret = ABORT_COMMAND
    else:
        ret = OS_INVALID
        write_debug_file(argv[0], "Invalid value of 'command'")

    return ret

def main(argv):

```

```

write_debug_file(argv[0], "Started")

# validate json and get command
msg = setup_and_check_message(argv)

if msg.command < 0:
    sys.exit(OS_INVALID)

if msg.command == ADD_COMMAND:
    alert = msg.alert["parameters"]["alert"]
    keys = [alert["rule"]["id"]]
    action = send_keys_and_check_message(argv, keys)

    # if necessary, abort execution
    if action != CONTINUE_COMMAND:

        if action == ABORT_COMMAND:
            write_debug_file(argv[0], "Aborted")
            sys.exit(OS_SUCCESS)
        else:
            write_debug_file(argv[0], "Invalid command")
            sys.exit(OS_INVALID)

try:
    file_path = msg.alert["parameters"]["alert"]["data"]["virustotal"]["source"]["file"]
    if os.path.exists(file_path):
        os.remove(file_path)
    write_debug_file(argv[0], json.dumps(msg.alert) + " Successfully removed threat")
except OSError as error:
    write_debug_file(argv[0], json.dumps(msg.alert) + "Error removing threat")

else:
    write_debug_file(argv[0], "Invalid command")

write_debug_file(argv[0], "Ended")

sys.exit(OS_SUCCESS)

if __name__ == "__main__":
    main(sys.argv)

```

Convert the active response Python script remove-threat.py to a Windows executable application. Run the following PowerShell command as an administrator to create the executable:

pyinstaller -F \path\_to\_remove-threat.py

Take note of the path where pyinstaller created remove-threat.exe.

Move the executable file remove-threat.exe to the C:\Program Files (x86)\ossec-agent\active-response\bin directory.

Restart the Wazuh agent to apply the changes.

Restart-Service -Name wazuh

Add the virustotal api key in ossec.conf file of wazuh manager –

```
413
414 <ossec_config>
415   <integration>
416     | <name>virustotal</name>
417     | <api_key>5db1635483f100aa03869c48791c964982556c2896e2a6012a05b85f6a047c8</api_key> <!-- Replace with your VirusTotal API key -->
418     | <group>syscheck</group>
419     | <alert_format>json</alert_format>
420   </integration>
421 </ossec_config>
^--
```

Append the following blocks to the Wazuh server /var/ossec/etc/ossec.conf file. This enables active response and trigger the remove-threat.exe executable when the VirusTotal query returns positive matches for threats

```
444
445 <ossec_config>
446   <command>
447     | <name>remove-threat</name>
448     | <executable>remove-threat.exe</executable>
449     | <timeout_allowed>no</timeout_allowed>
450   </command>
451
452   <active-response>
453     | <disabled>no</disabled>
454     | <command>remove-threat</command>
455     | <location>local</location>
456     | <rules_id>87105</rules_id>
457   </active-response>
458 </ossec_config>
^--
```

Add the following rules to the Wazuh server /var/ossec/etc/rules/local\_rules.xml file to alert about the active response results

```
55 <!-- Virustotal Windows -->
56 <group name="virustotal">
57   <rule id="100092" level="12">
58     <if_sid>657</if_sid>
59     <match>Successfully removed threat</match>
60     <description>$(parameters.program) removed threat located at $(parameters.alert.data.virustotal.source.file)</description>
61   </rule>
62
63 <rule id="100093" level="12">
64   <if_sid>657</if_sid>
65   <match>Error removing threat</match>
66   <description>Error removing threat located at $(parameters.alert.data.virustotal.source.file)</description>
67 </rule>
68 </group>
69
```

Restart the Wazuh manager to apply the configuration changes:  
 sudo systemctl restart wazuh-manager

## Downloading the malicious file in Downloads Folder –

Mar 14, 2024 @ 11:12:43.383 VirusTotal: Alert - c:\users\vrajp\downloads\eicar.txt - 66 engines detected this file		12	87105
Expanded document		View surrounding documents	View single document
Table	JSON		
t _index	wazuh-alerts-4.x-2024.03.14		
t agent.id	012		
t agent.ip	192.168.40.1		
t agent.name	Windows-10		
t data.integration	virustotal		
t data.virustotal.found	1		
t data.virustotal.malicious	1		
t data.virustotal.permalink	> <a href="https://www.virustotal.com/gui/file/275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f/detection/f-275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f-1710394861">https://www.virustotal.com/gui/file/275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f/detection/f-275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f-1710394861</a>		
t data.virustotal.positives	66		
t data.virustotal.scan_date	2024-03-14 05:41:01		
t data.virustotal.sha1	3395856ce81f2b7382dee72602f798b642f14140		
t data.virustotal.source.alert_id	1710394935.3930351		
t data.virustotal.source.file	c:\users\vrajp\downloads\eicar.txt		
t data.virustotal.source.md5	44d88612fea8a8f36de82e1278abb02f		
t data.virustotal.source.sha1	3395856ce81f2b7382dee72602f798b642f14140		
t data.virustotal.total	67		
t decoder.name	json		
t id	1710394963.3931564		

## File deleted –

Mar 14, 2024 @ 11:12:15.330 File deleted.		7	553
Expanded document		View surrounding documents	View single document
Table	JSON		
t _index	wazuh-alerts-4.x-2024.03.14		
t agent.id	012		
t agent.ip	192.168.40.1		
t agent.name	Windows-10		
t decoder.name	syscheck_deleted		
t full_log	File 'c:\users\vrajp\downloads\eicar.txt' deleted Mode: realtime		
t id	1710394935.3930351		
t input.type	log		
t location	syscheck		
t manager.name	wazuh-server		
t rule.description	File deleted.		
# rule.firedtimes	1		
t rule.gdpr	II_5.1.f		
t rule.gpg13	4.11		
t rule.groups	ossec, syscheck, syscheck_entry_deleted, syscheck_file		
t rule.hipaa	164.312.c.1, 164.312.c.2		
t rule.id	553		
# rule.level	7		

## **CHAPTER: 9 WAZUH AND SOAR INTEGRATION**

## CHAPTER: 9 WAZUH AND SOAR INTEGRATION

### WAZUH & SHUFFLE SOAR INTEGRATION

Perform the steps below on the Shuffle dashboard to create a new workflow and extract the webhook URI.

1. Create a new workflow on Shuffle titled “Wazuh integration test.”

New workflow

Name: WAZUH INTEGRATION TEST

Description

Usecases Tags

Getting Started

Setup progress: 50%

Follow these steps to get you up and running!

Watch 2-min introduction video

1. Find relevant apps

2. Discover Usecases

3. Invite teammates

4. Security & Stability

2. Click on the Triggers tab in the bottom left and drag the Webhook to the workspace.

Workflow starters

- Webhook Custom HTTP input
- Schedule Specify time
- Office365 O365 email trigger
- Gmail Gmail email trigger

Mid-Workflow

- Shuffle Workflow Control a workflow
- User Input Wait for user input

Workflows > Wazuh integration test

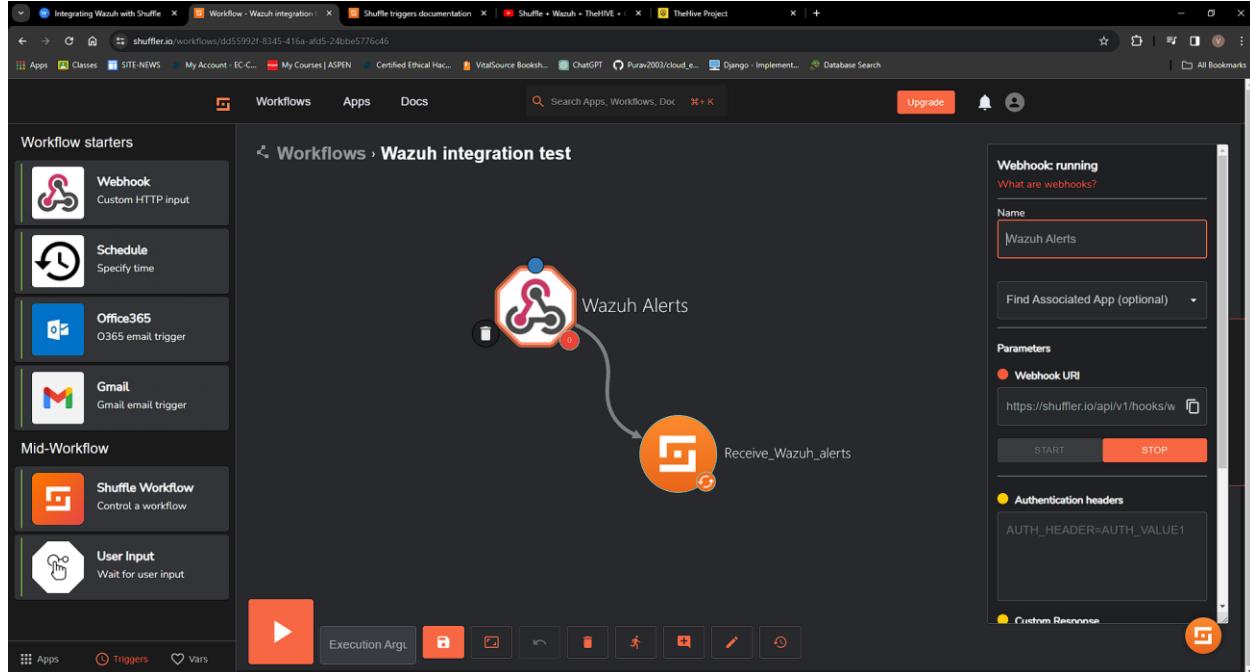
Wazuh Alerts

Receive\_Wazuh\_alerts

Explore runs

- Click on the webhook and rename it to Wazuh alerts. Copy and save the webhook URI because we use it when configuring the Wazuh server. Start the webhook. The webhook URI looks like the following:

[https://<YOUR\\_SHUFFLE\\_URL>/api/v1/hooks/webhook\\_18cdc939-5e7b-428e-b82f-e1481ffc8fe6](https://<YOUR_SHUFFLE_URL>/api/v1/hooks/webhook_18cdc939-5e7b-428e-b82f-e1481ffc8fe6)



## WAZUH SERVER

Perform the steps below on the Wazuh server to configure Wazuh to send alerts to Shuffle for analysis.

- Add the following configuration in between the <ossec\_config> block of the Wazuh server /var/ossec/etc/ossec.conf file:

```
<integration>
<name>shuffle</name>
<hook_url><YOUR_SHUFFLE_URL>/api/v1/hooks/<HOOK_ID></hook_url>
<level>3</level>
<alert_format>json</alert_format>
</integration>
```

Where:

<name>: This is the name of the integration.

<hook\_url>: This is the webhook URI copied from the Shuffle webhook. Your Shuffle URL depends on your deployment, for example, http://<SHUFFLE\_IP>:3001 for a Shuffle on-premise deployment and https://shuffler.io for Shuffle Cloud.

<level>: This is used to forward a specific alert level.

<alert\_format>: This forwards alerts to Shuffle in JSON format.

For more information on available options, check out this guide.

```
<!-- Shuffle SOAR -->
<integration>
  <name>shuffle</name>
  <hook_url>https://shuffler.io/api/v1/hooks/webhook_0ed0f798-02d4-40e5-893a-842170df7059</hook_url>
  <level>3</level>
  <alert_format>json</alert_format>
</integration>
```

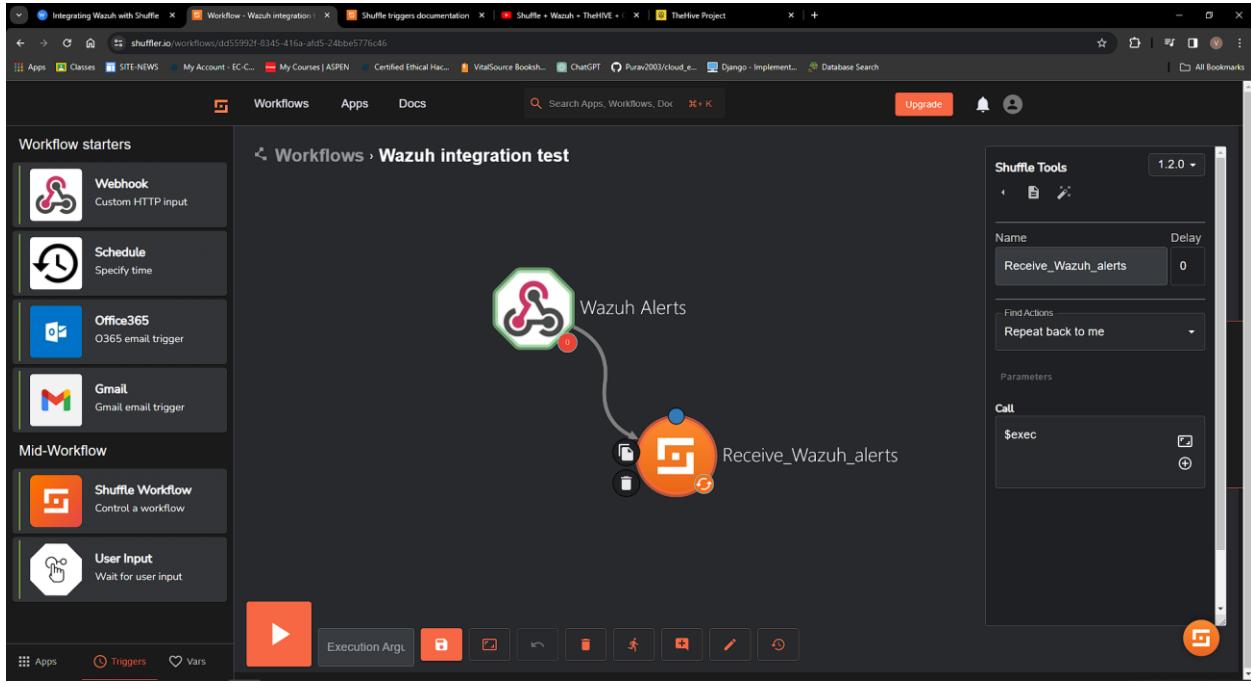
## 2. Restart the Wazuh manager service to apply changes:

sudo systemctl restart wazuh-manager

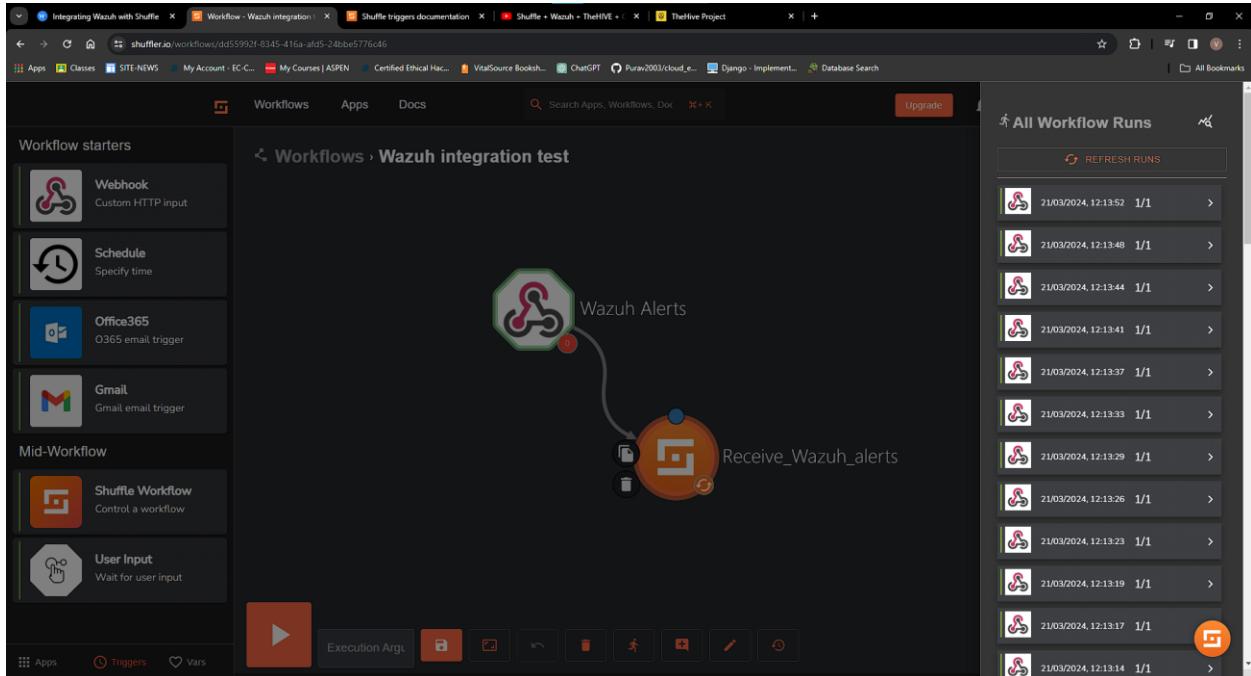
```
[root@wazuh-server ~]# sudo systemctl restart wazuh-manager
[root@wazuh-server ~]# sudo systemctl status wazuh-manager
● wazuh-manager.service - Wazuh manager
   Loaded: loaded (/usr/lib/systemd/system/wazuh-manager.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2024-03-21 06:06:00 UTC; 14s ago
     Process: 22738 ExecStop=/usr/bin/env /var/ossec/bin/wazuh-control stop (code=exited, status=0/SUCCESS)
    Process: 22896 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-control start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/wazuh-manager.service
           ├─22956 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
           ├─22957 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
           ├─22960 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
           ├─22963 /var/ossec/framework/python/bin/python3 /var/ossec/api/scripts/wazuh-apid.py
           ├─22988 /var/ossec/bin/wazuh-integratord
           ├─23008 /var/ossec/bin/wazuh-authd
           ├─23025 /var/ossec/bin/wazuh-db
           ├─23050 /var/ossec/bin/wazuh-execd
           ├─23062 /var/ossec/bin/wazuh-maild
           ├─23068 /var/ossec/bin/wazuh-analysisd
           ├─23089 /var/ossec/bin/wazuh-syscheckd
           ├─23106 /var/ossec/bin/wazuh-remoted
           ├─23140 /var/ossec/bin/wazuh-logcollector
```

## Shuffle SOAR

1. Click on the Shuffle Tools app named “Change me” and rename it to Receive\_Wazuh\_alerts. Set the call option to “\$exec”, and save the workflow. This Shuffle app now repeats the events that are received by the Wazuh alerts webhook. This allows us to test that Shuffle can receive Wazuh alerts.



2. Click on the show executions button.



3. Select any execution and expand it for details. You should see a Wazuh alert in the output.

The screenshot shows the Shuffle.io platform interface. On the left, there's a sidebar with 'Workflow starters' and 'Mid-Workflow' sections. In the main area, a workflow named 'Receive Wazuh alerts' is displayed, showing its status as 'SUCCESS'. The output of the workflow is a JSON object containing details about Wazuh alerts. On the right, a list titled 'All Workflow Runs' shows multiple executions of the workflow, each with a timestamp and a success rate of '1/1'.

**Workflow starters**

- Webhook
- Schedule
- Office365
- Gmail

**Mid-Workflow**

- Shuffle Workflow
- User Input

**Workflow Details**

**Receive Wazuh alerts**

Status SUCCESS

Results for Receive\_Wazuh\_alerts: { 8 items

```
severity": 2
"pretext": "Wazuh Alert"
"title": "PowerShell Information Eventlog"
"tek": [
    {
        "id": 1,
        "data": [
            {
                "system": [
                    {
                        "providerName": "Microsoft-Windows-PowerShell",
                        "providerGuid": "{a0c1053b-5c40-4b15-8766-3cf1c58f085a}",
                        "eventId": "535004",
                        "version": "1",
                        "level": "4",
                        "task": "111",
                        "opcode": "10",
                        "keywords": "0x0",
                        "systemTime": "2024-03-21T06:42:58.4302386Z",
                        "eventRecordID": "10800",
                        "processID": "17836",
                        "threadID": "21116",
                        "channel": "Microsoft-Windows_Powershell/Operational",
                        "computer": "DESKTOP-7VREV1I",
                        "severityValue": "INFORMATION",
                        "message": "Windows PowerShell has started an IPC listening thread on process: 17036 in"
                    }
                ]
            }
        ]
    }
]
```

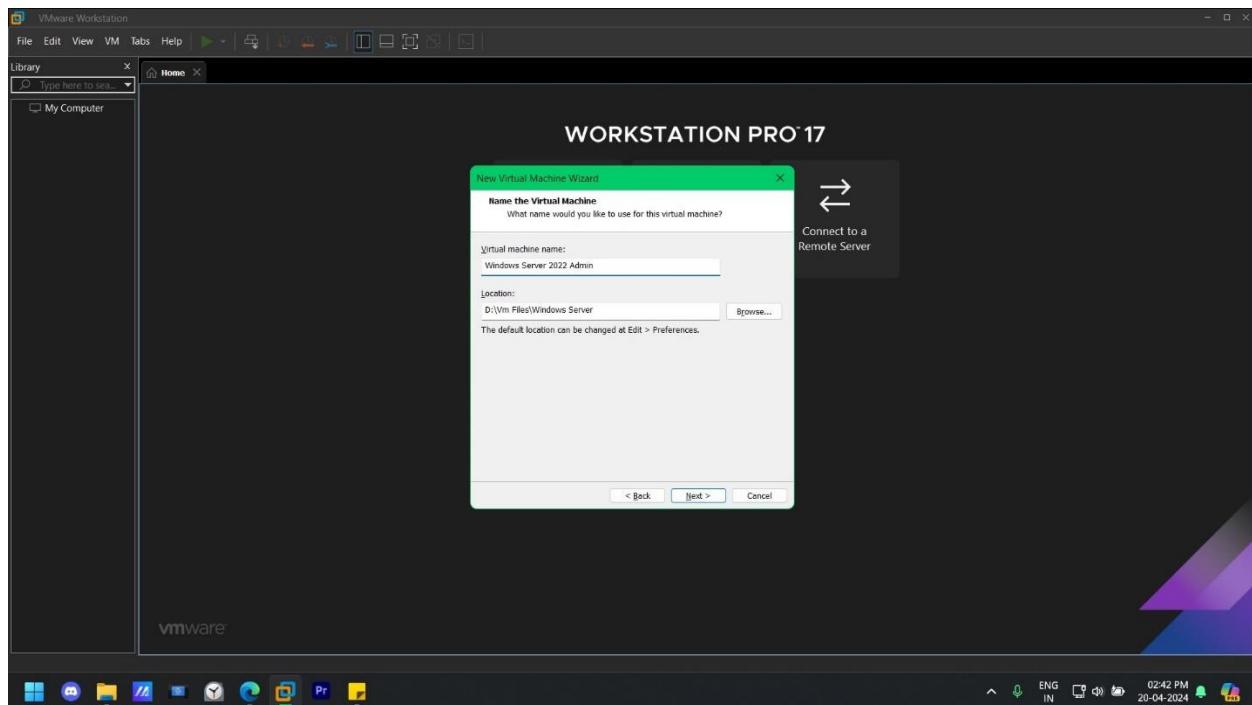
**All Workflow Runs**

Date	Runs
21/03/2024, 12:14:18	1/1
21/03/2024, 12:14:14	1/1
21/03/2024, 12:14:11	1/1
21/03/2024, 12:14:07	1/1
21/03/2024, 12:14:03	1/1
21/03/2024, 12:13:59	1/1
21/03/2024, 12:13:52	1/1
21/03/2024, 12:13:48	1/1
21/03/2024, 12:13:44	1/1
21/03/2024, 12:13:41	1/1
21/03/2024, 12:13:37	1/1
21/03/2024, 12:13:33	1/1

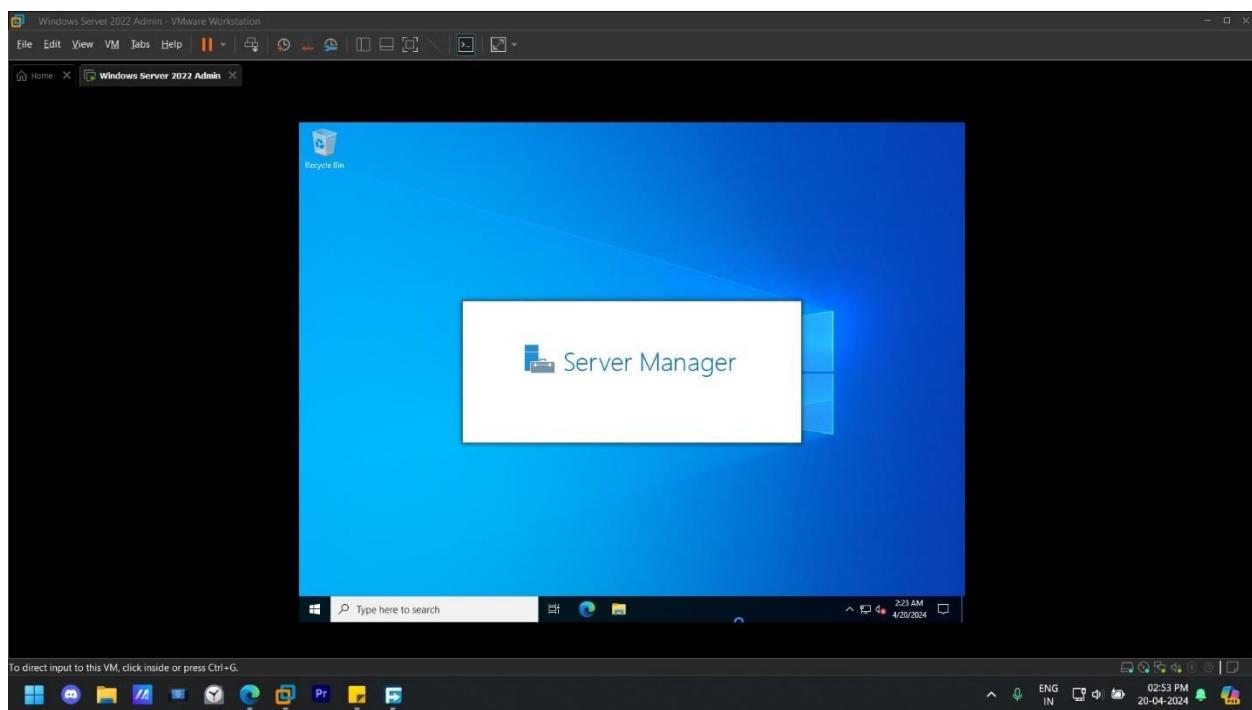
## **CHAPTER: 10 WINDOWS AD CREATION**

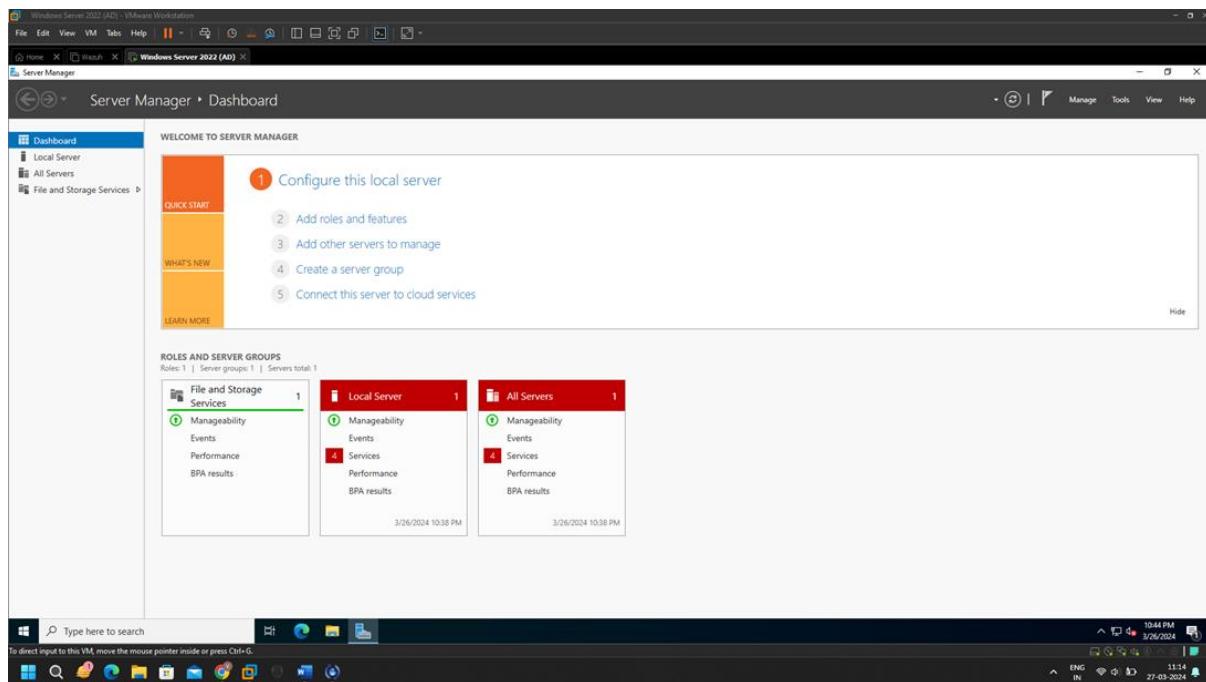
## CHAPTER: 10 WINDOWS AD CREATION

Installing windows server 2022 in VM –

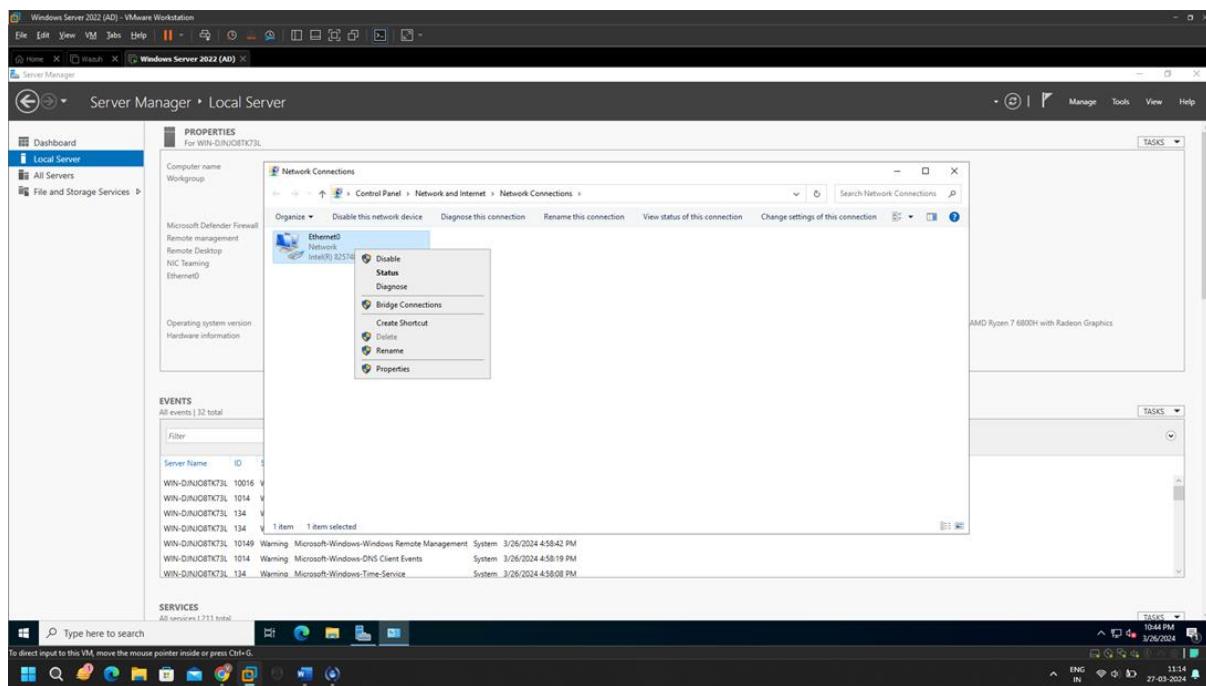


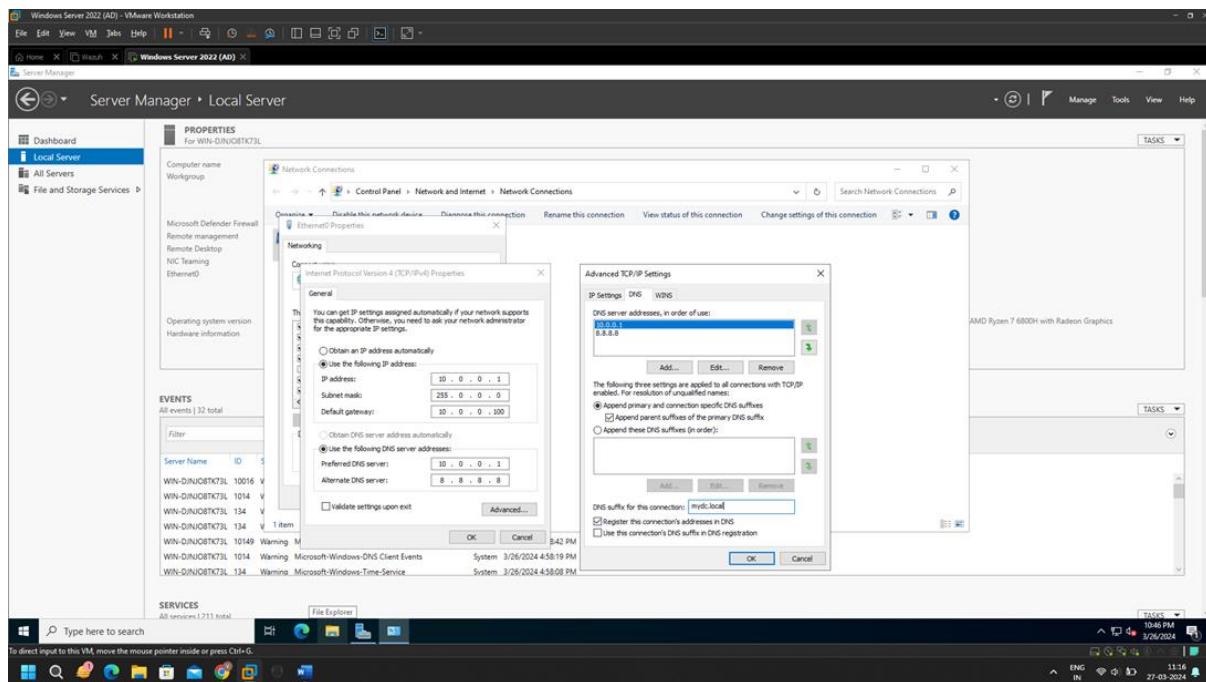
Start Server Manager after installation –



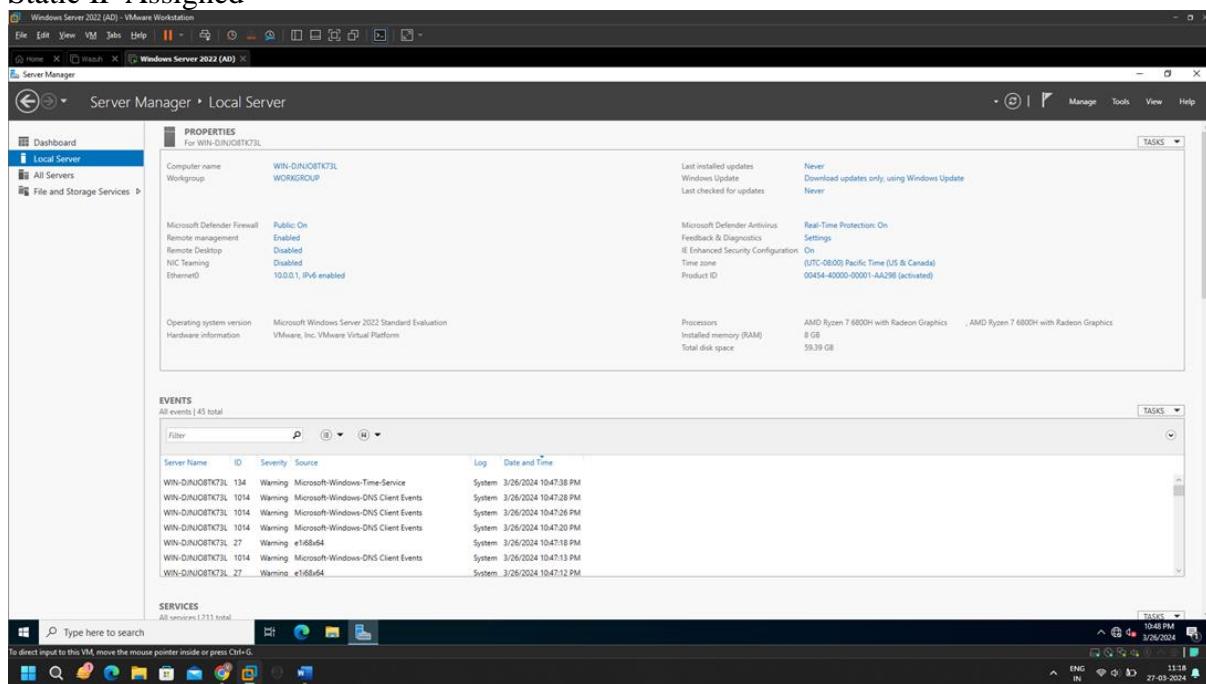


## Goto Local Server and Configure IP –





## Static IP Assigned –



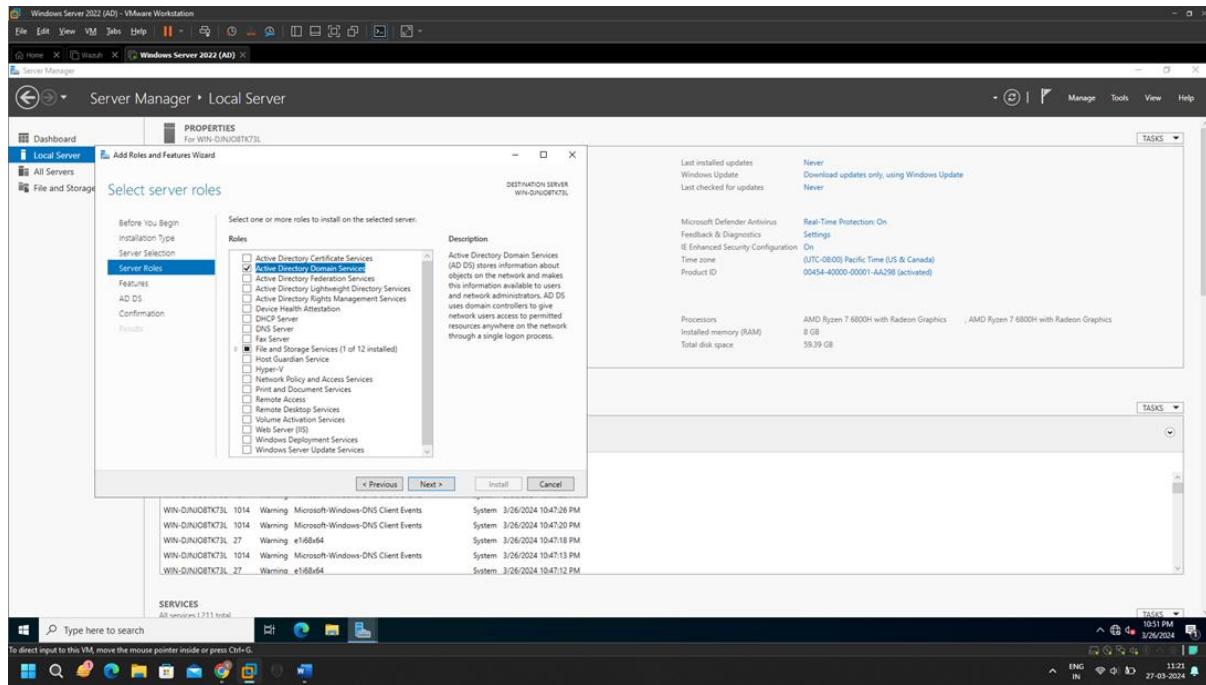
## Adding Server Role –

The screenshot shows the Windows Server 2022 Local Server Properties window. The 'PROPERTIES' tab is selected, showing basic server information like Computer name (WIN-DINOBTK73L), Workgroup (WORKGROUP), and Microsoft Defender Firewall status. The 'EVENTS' tab shows a log of 145 events, mostly warnings related to DNS Client and Windows Update. Below the properties window is a screenshot of the Windows desktop environment, showing the taskbar with various icons and system status.

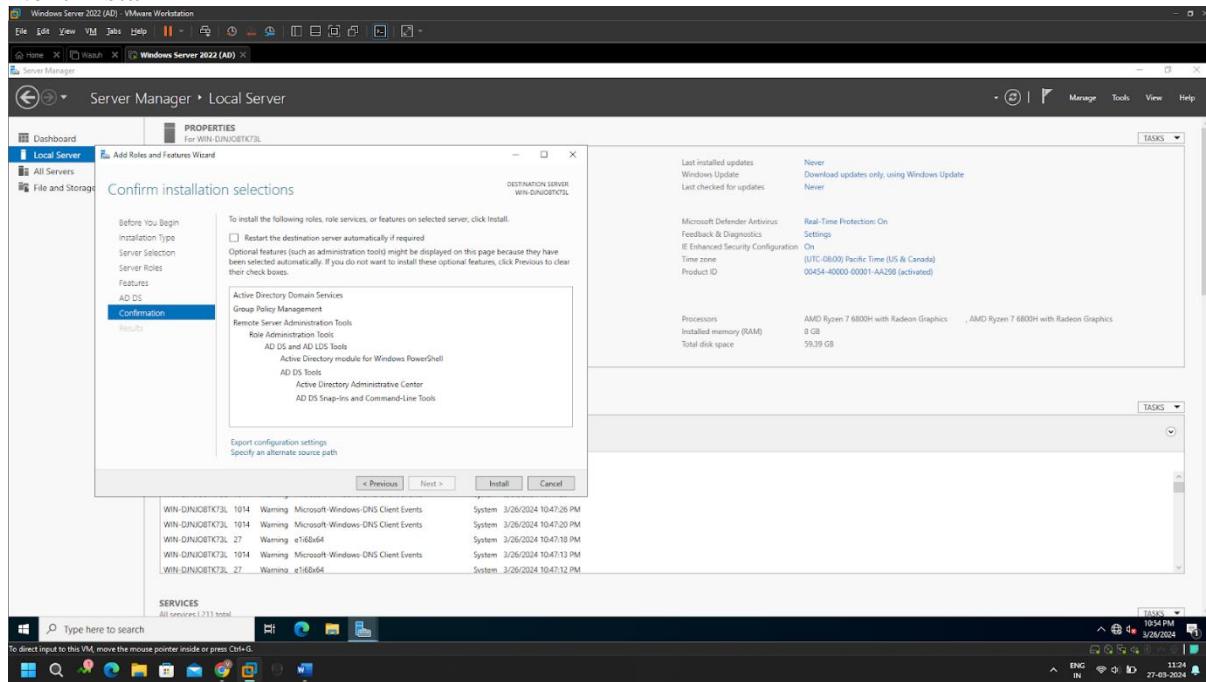
Click Next,

The screenshot shows the 'Add Roles and Features Wizard' window, specifically the 'Before you begin' step. It provides instructions for installing roles, services, or features. The 'PROPERTIES' tab of the Local Server is also visible, showing the same basic server information as the previous screenshot. The desktop environment at the bottom remains consistent.

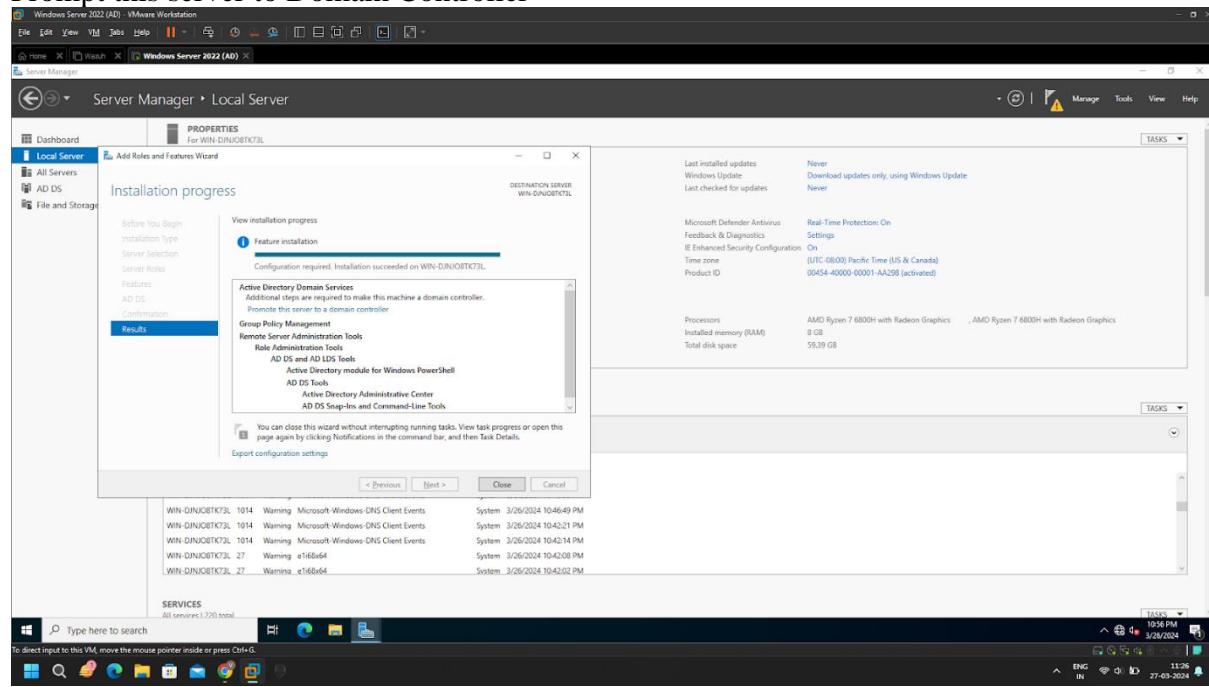
Select – Active Directory Domain Services and Click Next,



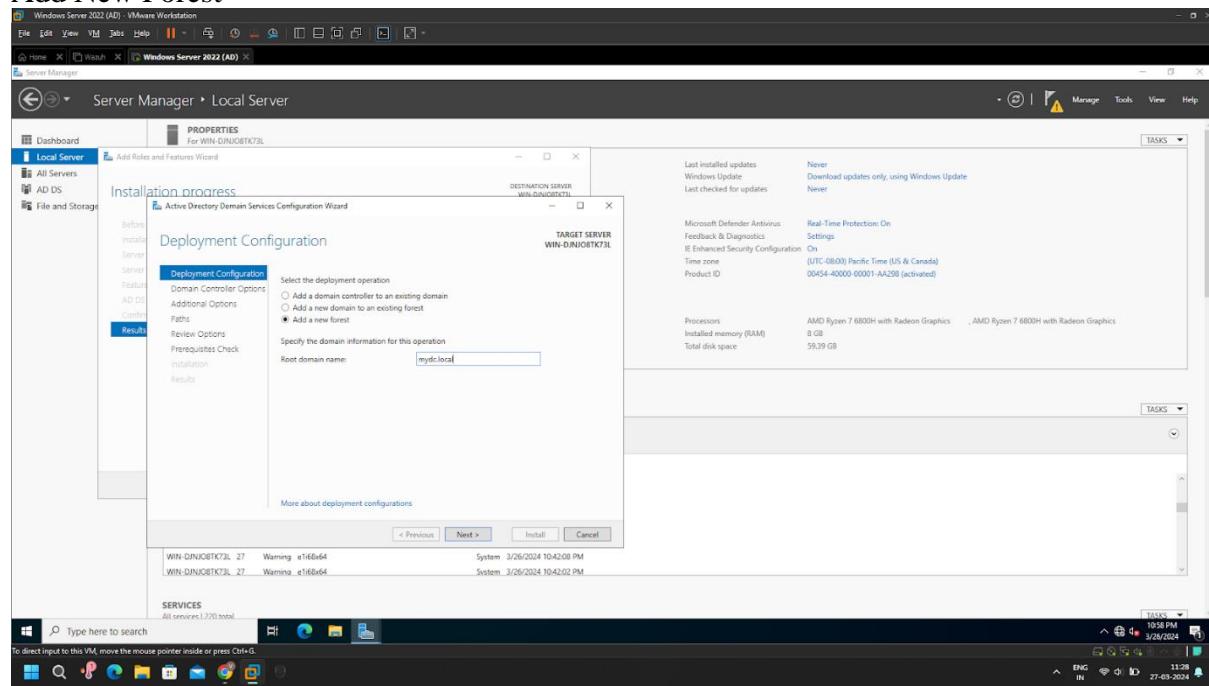
Next Install –

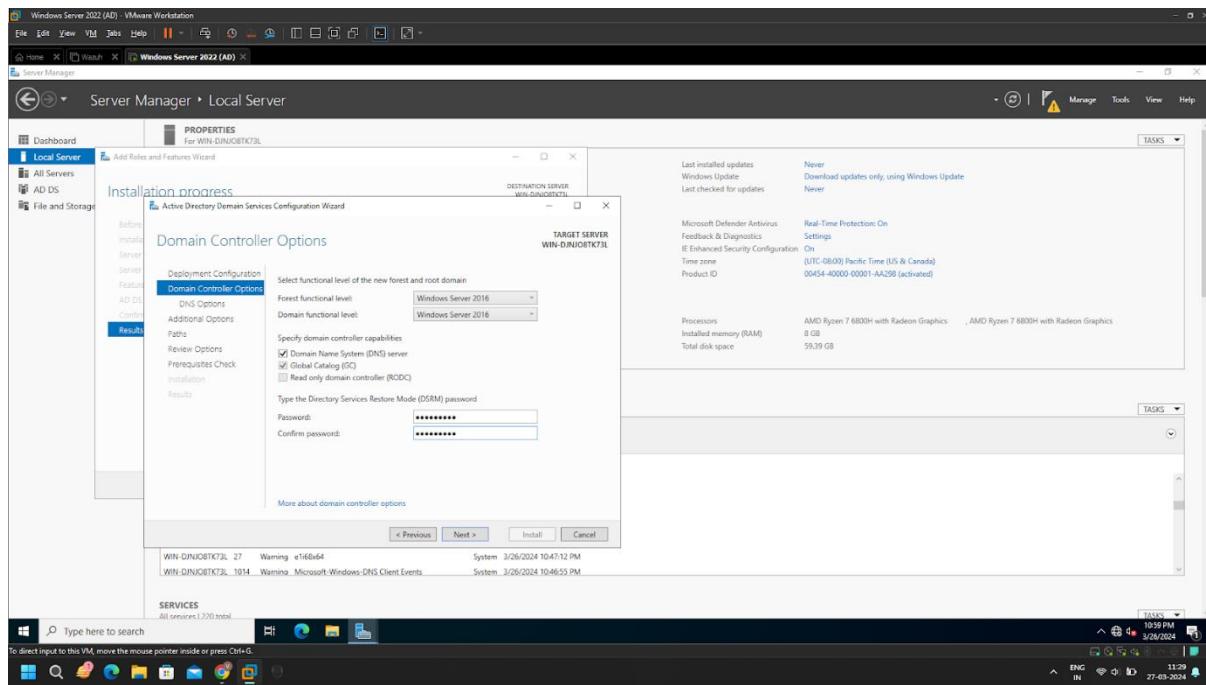


## Prompt this server to Domain Controller

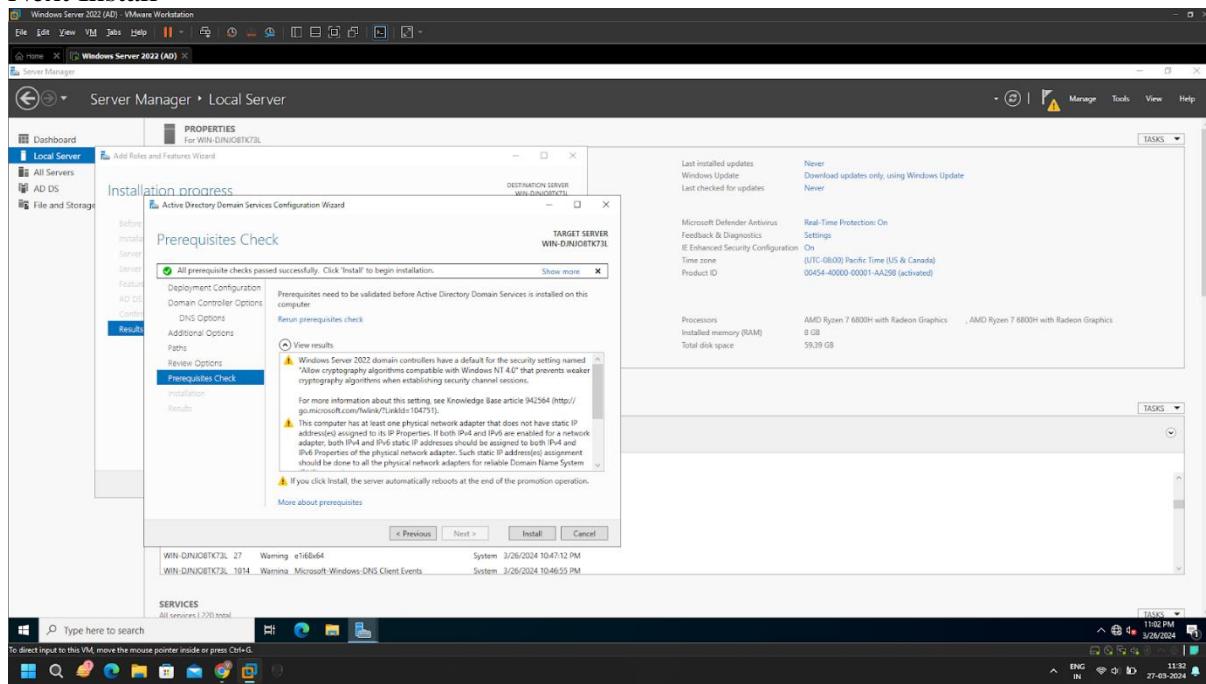


## Add New Forest –

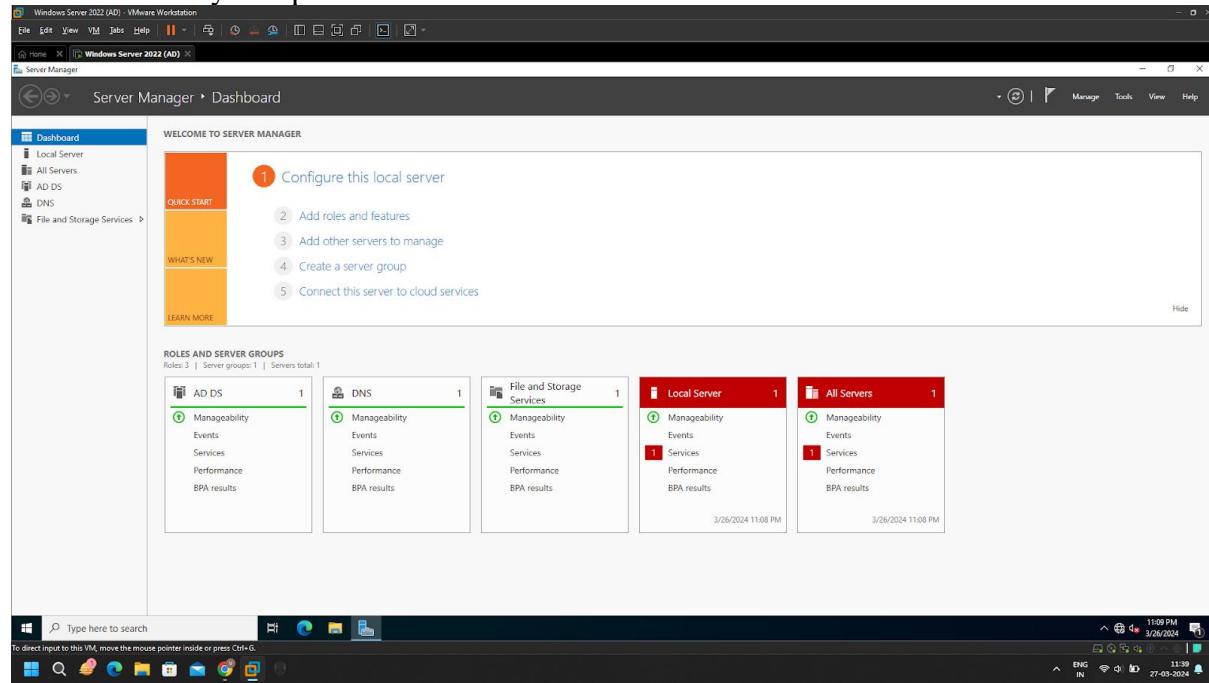




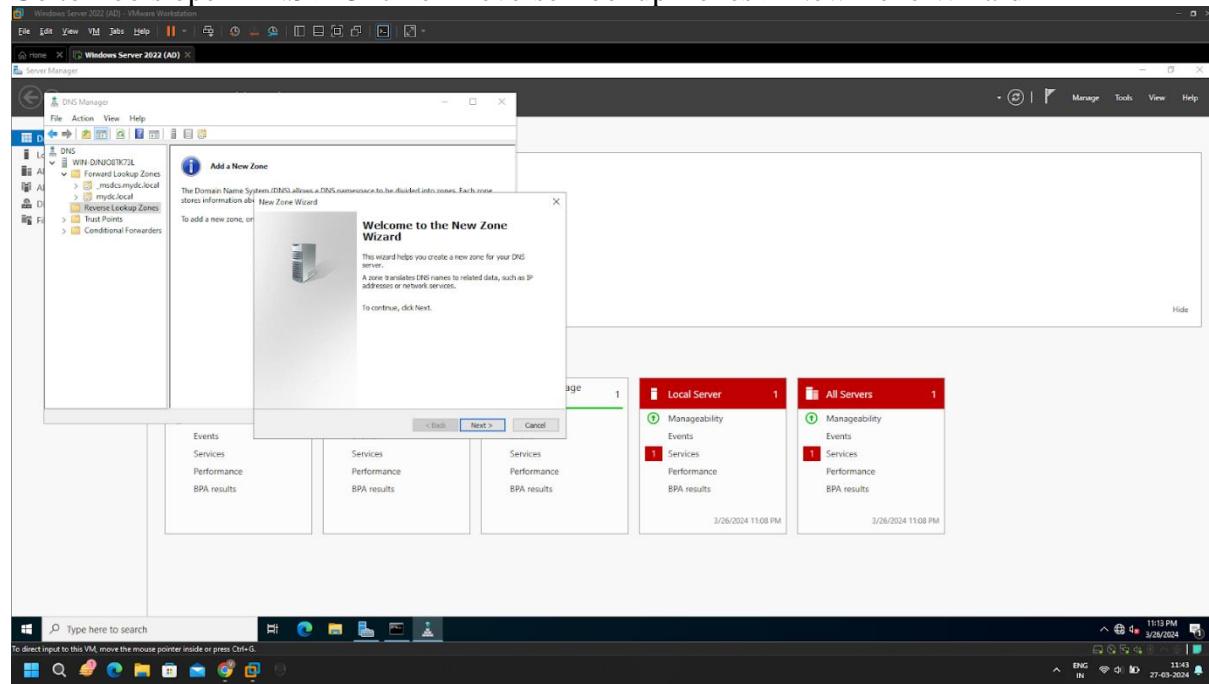
## Next Install –



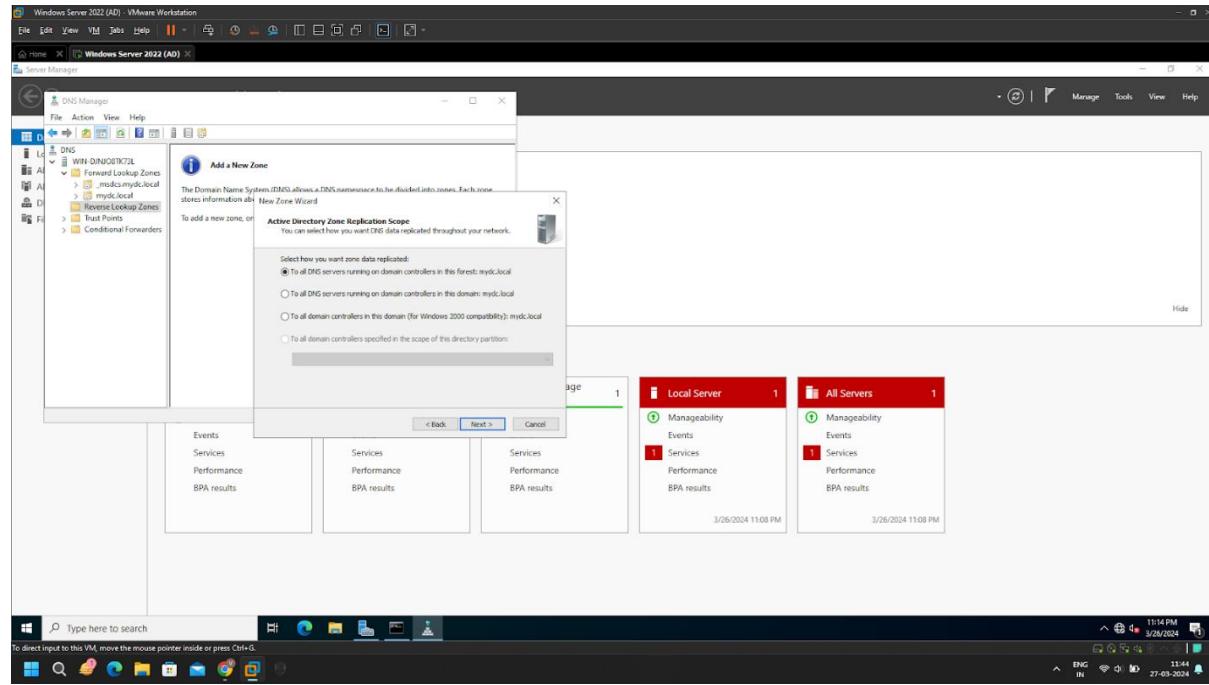
## Active Directory Setup Done



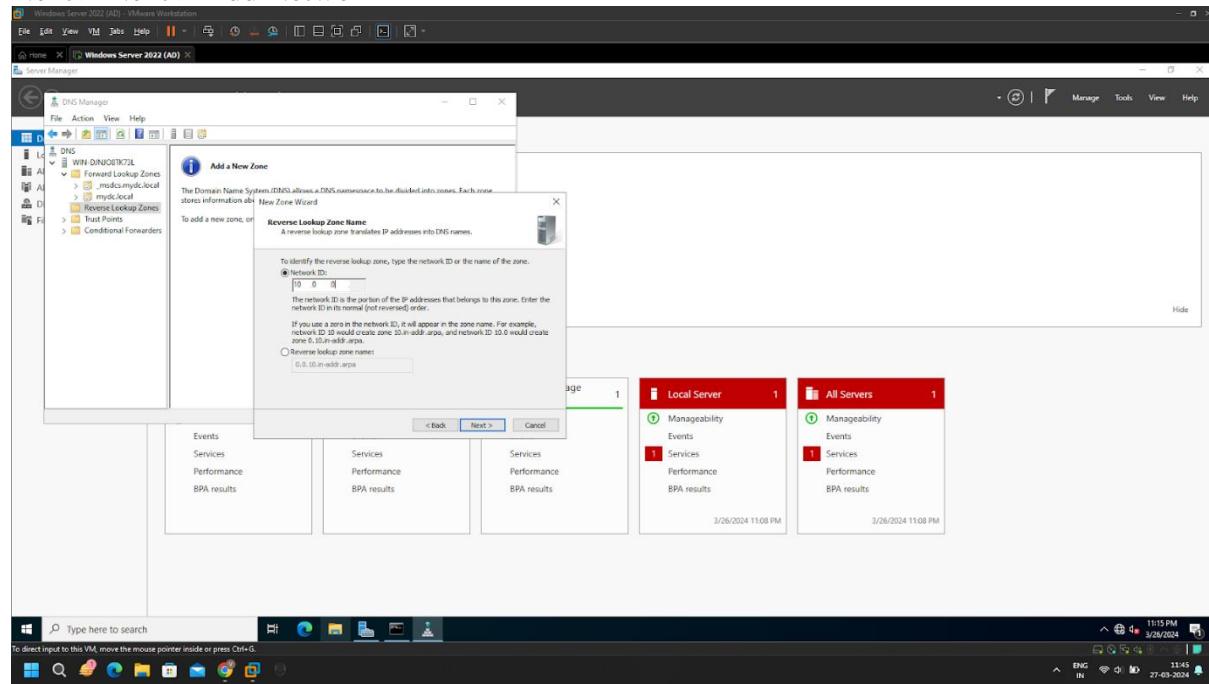
Go to Tools open DNS > Click on Reverse Lookup Zones > New Zone Wizard



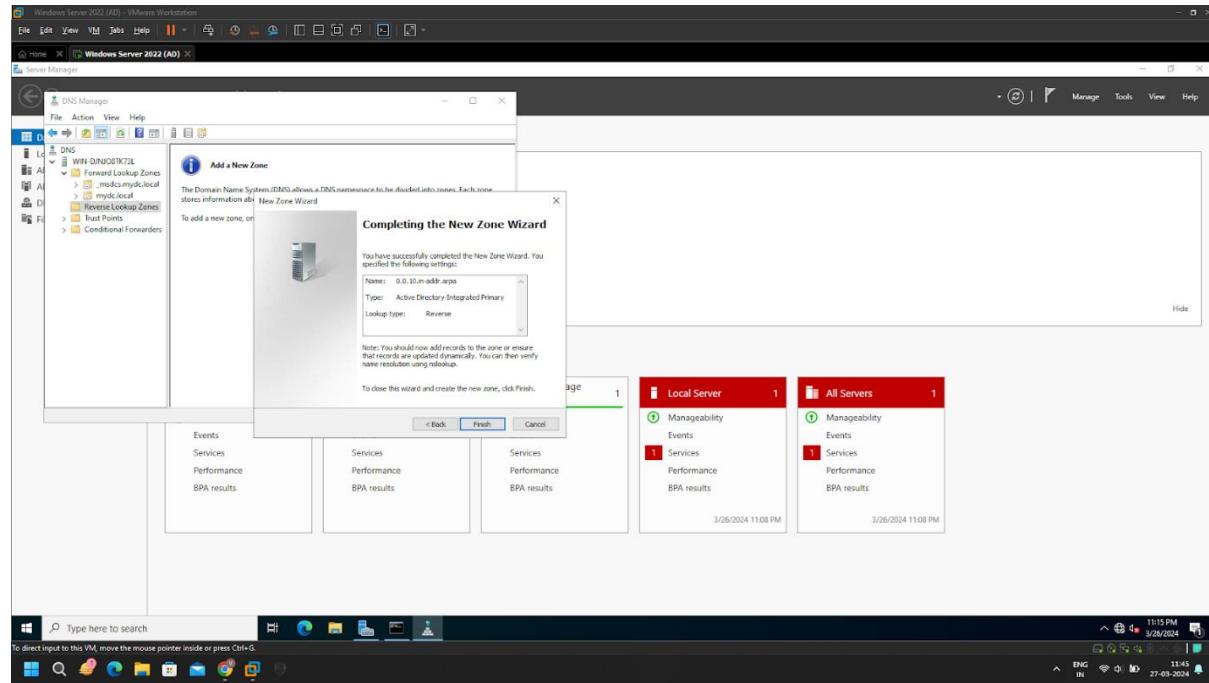
## Click Next –



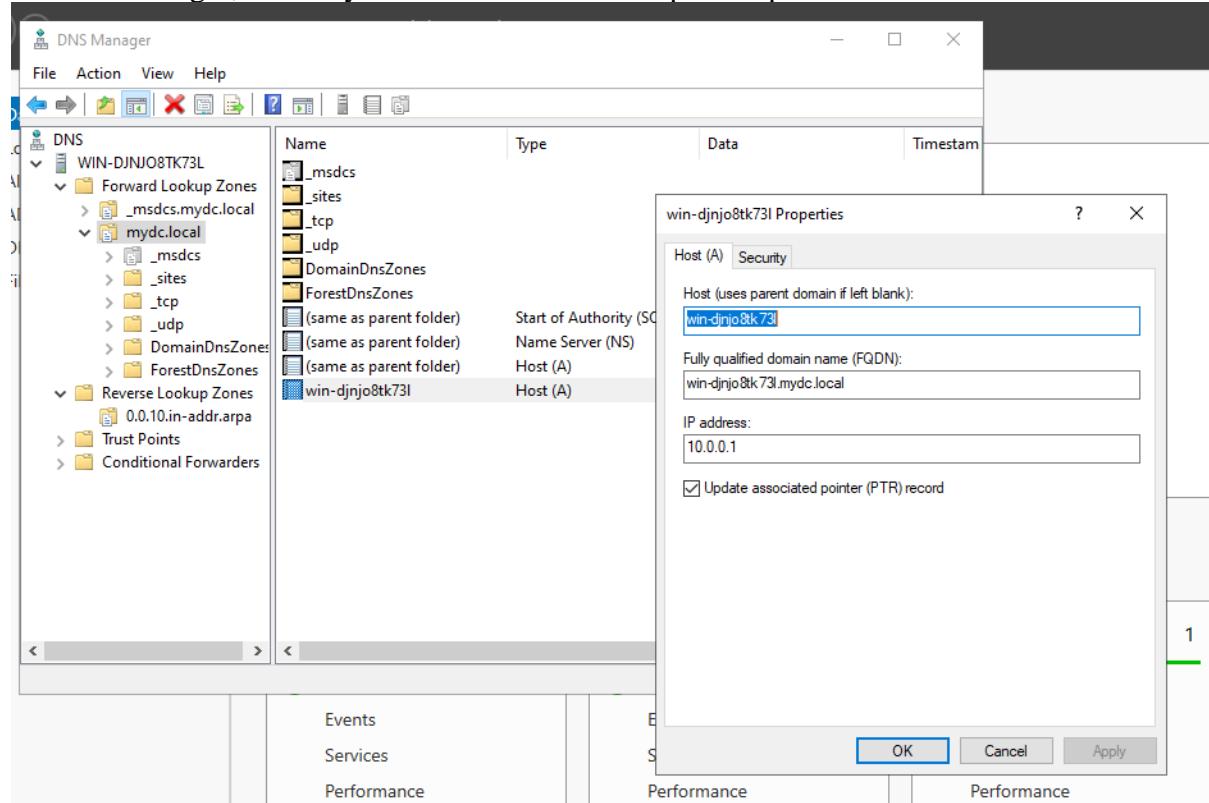
## Next > Next > Add Network ID –



## Next > Finish



In DNS Manager, Goto mydc.local > “username” open Properties > Tick on PTR



Click OK.

## DNS configured –

The screenshot shows a Windows Server 2022 environment within a VMware Workstation window. The taskbar at the bottom indicates the system is connected to a network (TEN IN) and has a battery level of 11:53 on 27-03-2024.

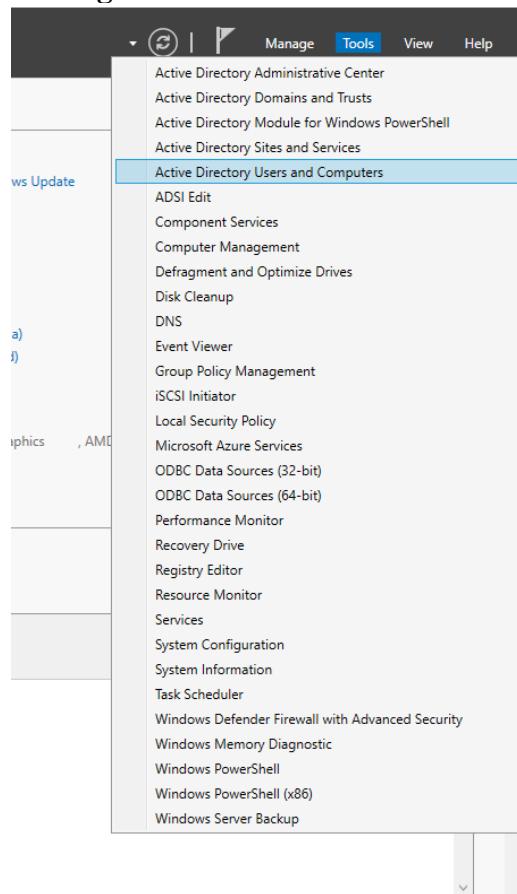
The main window displays the DNS Manager interface. On the left, the 'Ethernet adapter Ethernet' properties are shown, including the connection-specific DNS suffix (mydc.local), link-local IPv6 address, IPv4 address (10.0.0.1), subnet mask (255.0.0.0), and default gateway (10.0.0.100). The right pane shows the command prompt output of 'nslookup' commands. One command shows the local host entry for 'WIN-D3D08TK73L.mydc.local' with an address of '10.0.0.1'. Another command shows a query for 'www.google.com' with an address of '10.0.0.1'. Below the command prompt are four small windows showing server management status: 'Local Server' (1 item), 'All Servers' (1 item), 'Manageability' (Events, Services, Performance, BPA results), and another 'All Servers' window (1 item).

```
C:\Users\Administrator>nslookup
DNS request timed out.
    timeout was 2 seconds.
Default Server: Unknown
Address: ::1

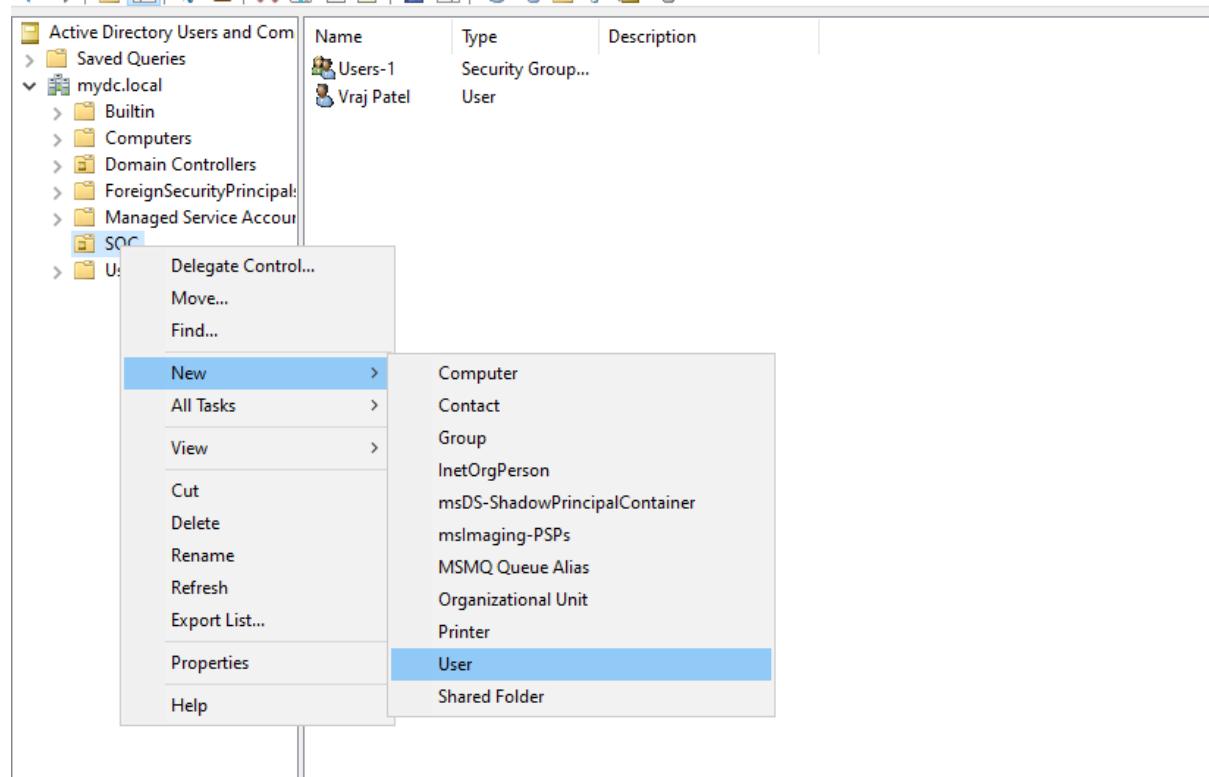
> exit
C:\Users\Administrator>nslookup
DNS request timed out.
    timeout was 2 seconds.
Default Server: Unknown
Address: ::1

< Name: WIN-D3D08TK73L.mydc.local
Address: 10.0.0.1
```

## Adding Users –



Click on mydc.local > Create New OU(SOC) > Right click on that and create New User,



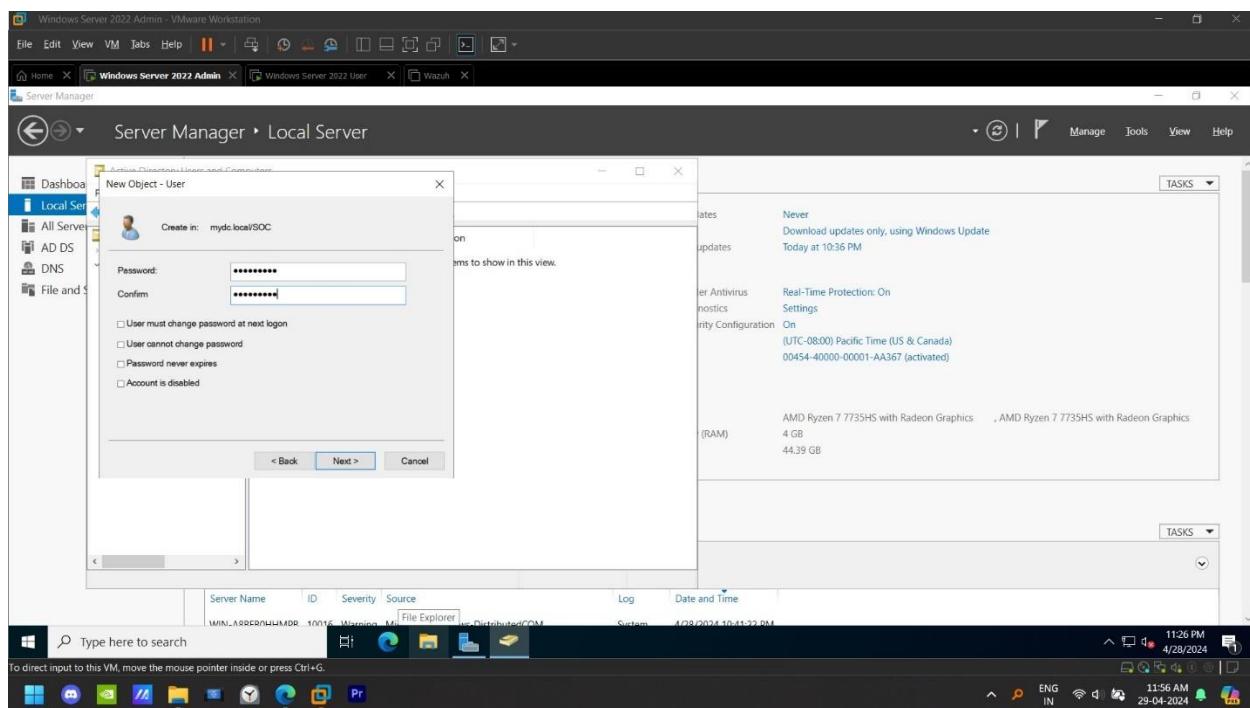
New Object - User

Create in: mydc.local/SOC

First name:	Akshar	Initials:	<input type="text"/>
Last name:	Patel		
Full name:	Akshar Patel		
User logon name:	vatana	@mydc.local	<input type="button" value="▼"/>
User logon name (pre-Windows 2000):	MYDC\	vatana	

< Back  Cancel

## Add Password –



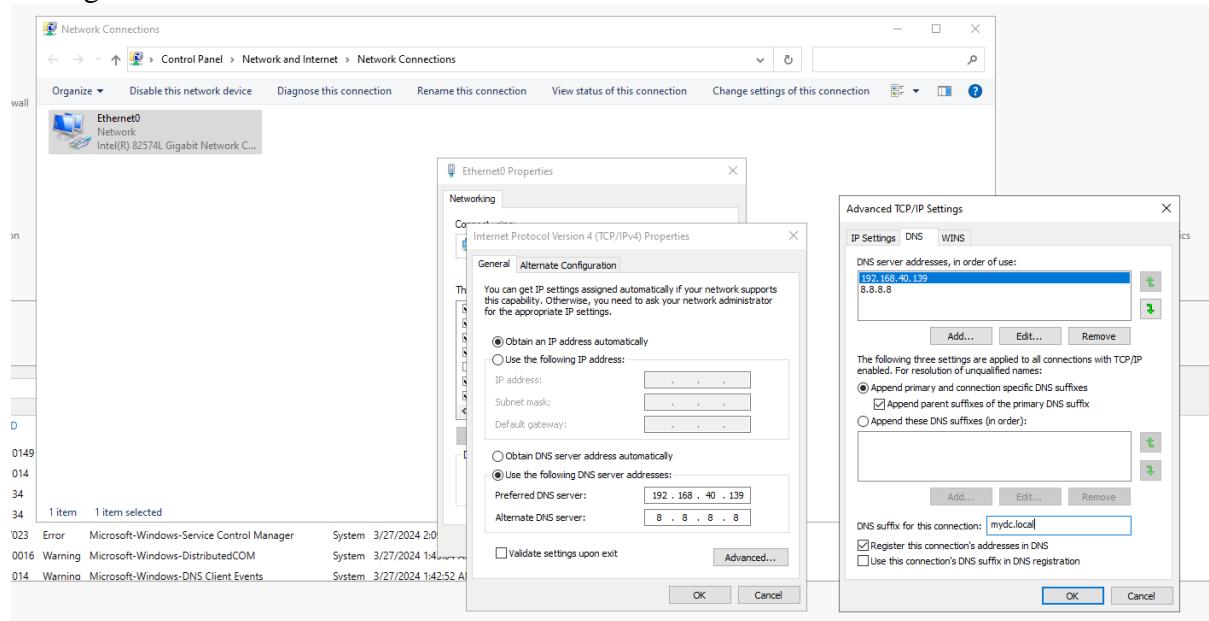
## User created –

The screenshot shows the Active Directory Users and Computers console. The left pane displays the navigation tree: 'Saved Queries', 'mydc.local' (with subfolders 'Builtin', 'Computers', 'Domain Controllers', 'ForeignSecurityPrincipal', 'Managed Service Account', 'SOC', and 'Users'). The right pane shows the 'Description' view with three entries: 'Akshar Patel' (User), 'Users-1' (Security Group...), and 'Vraj Patel' (User).

Create new VM to login with the created credentials

Configuring another pc to add it in the workgroup mydc.local

Configure Network

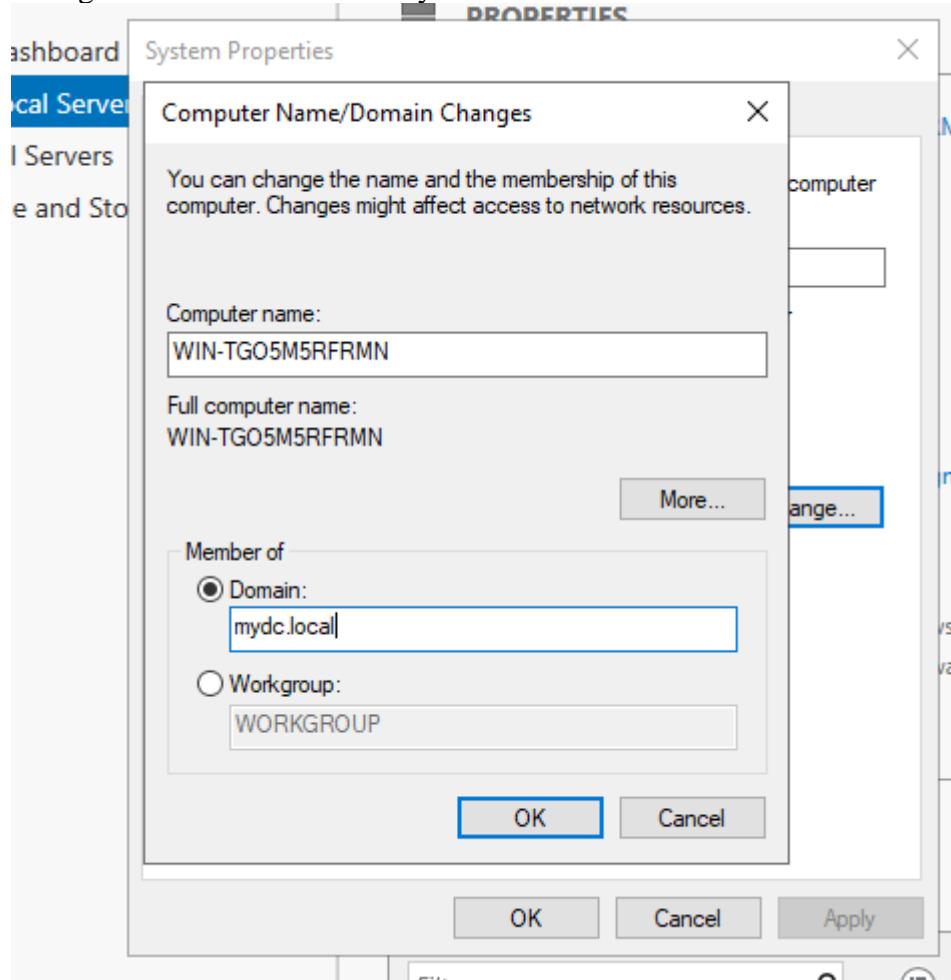


Goto Local Server > Workgroup

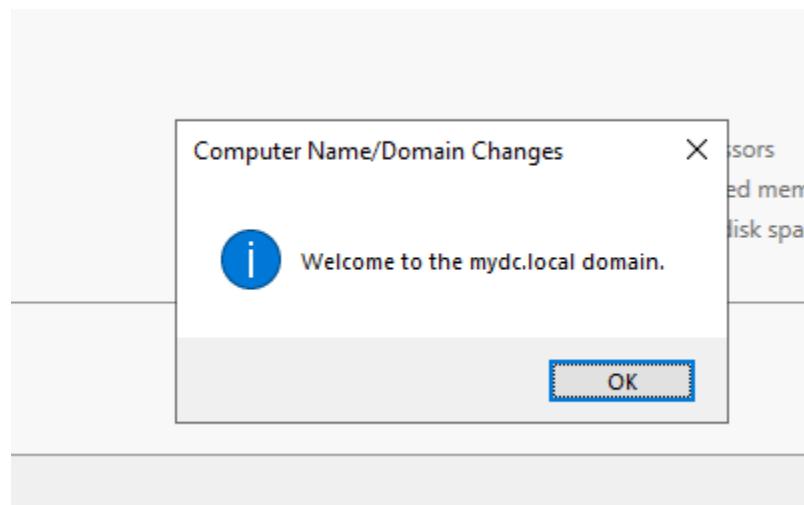
The screenshot shows the 'Local Server' properties page. The left sidebar lists 'Dashboard', 'Local Server' (selected), 'All Servers', and 'File and Storage Services'. The main pane displays the properties for 'WIN-TGO5M5RFRMN' with the following details:

Computer name	WIN-TGO5M5RFRMN
Workgroup	WORKGROUP

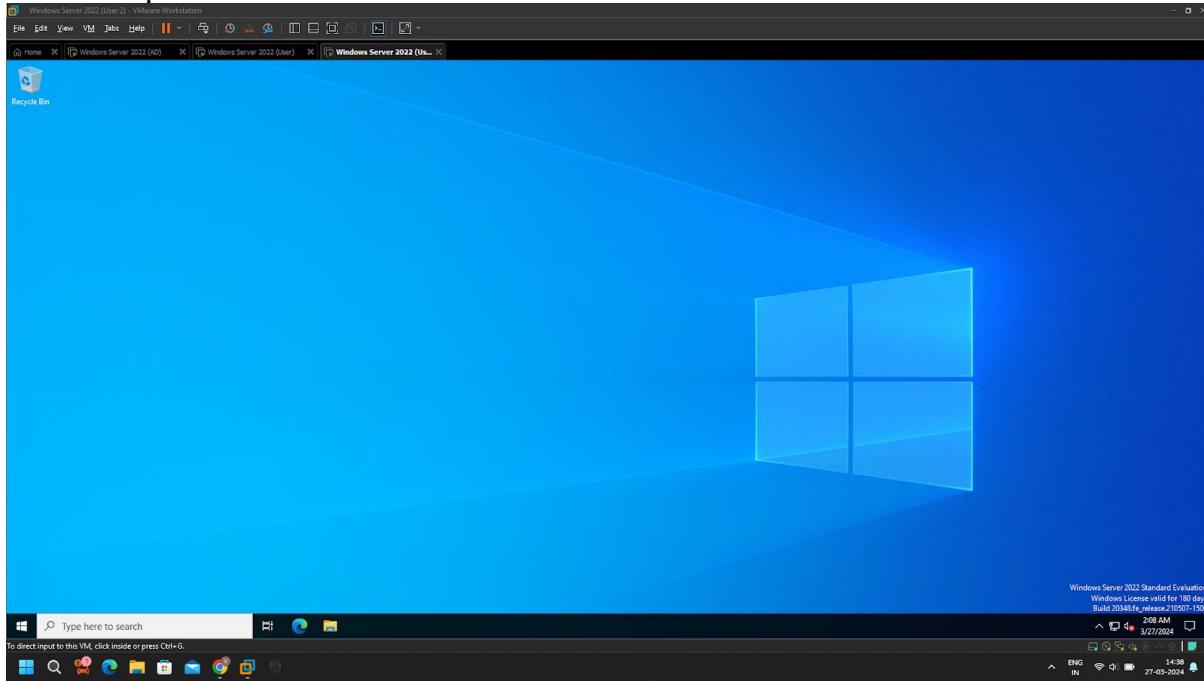
Change it to Domain and Enter your domain



Click OK.



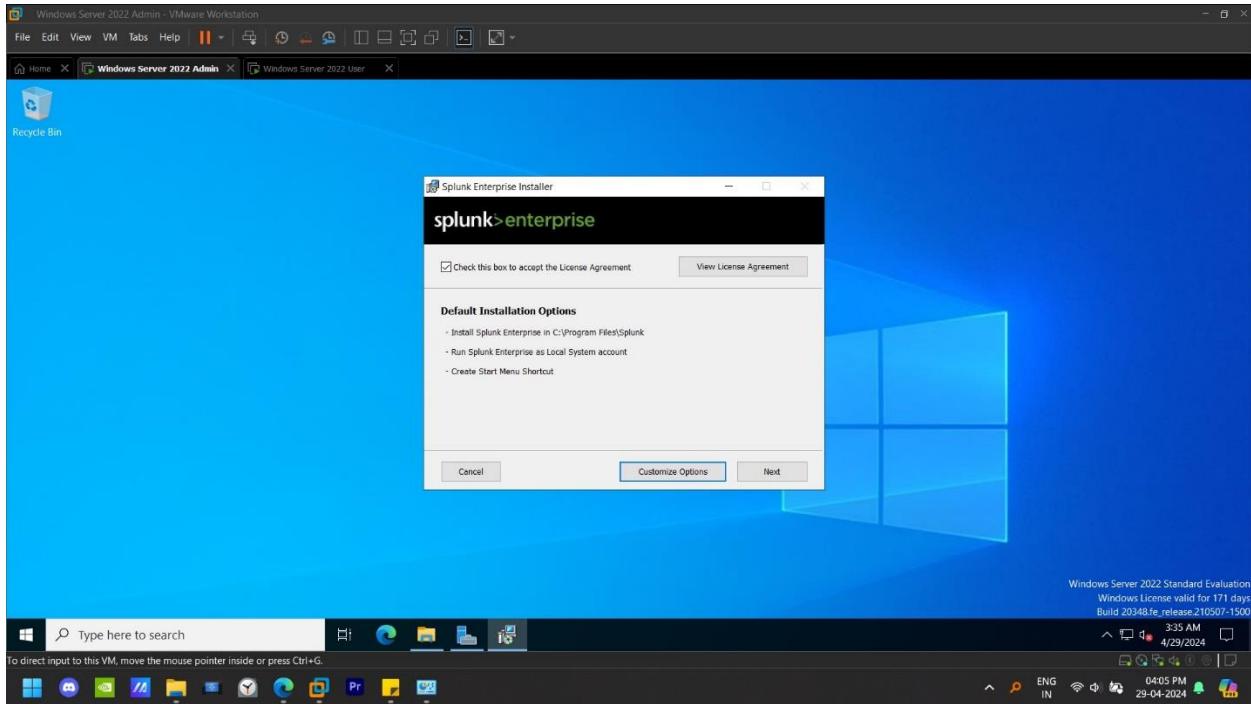
## Account Opened – User Akshar Patel



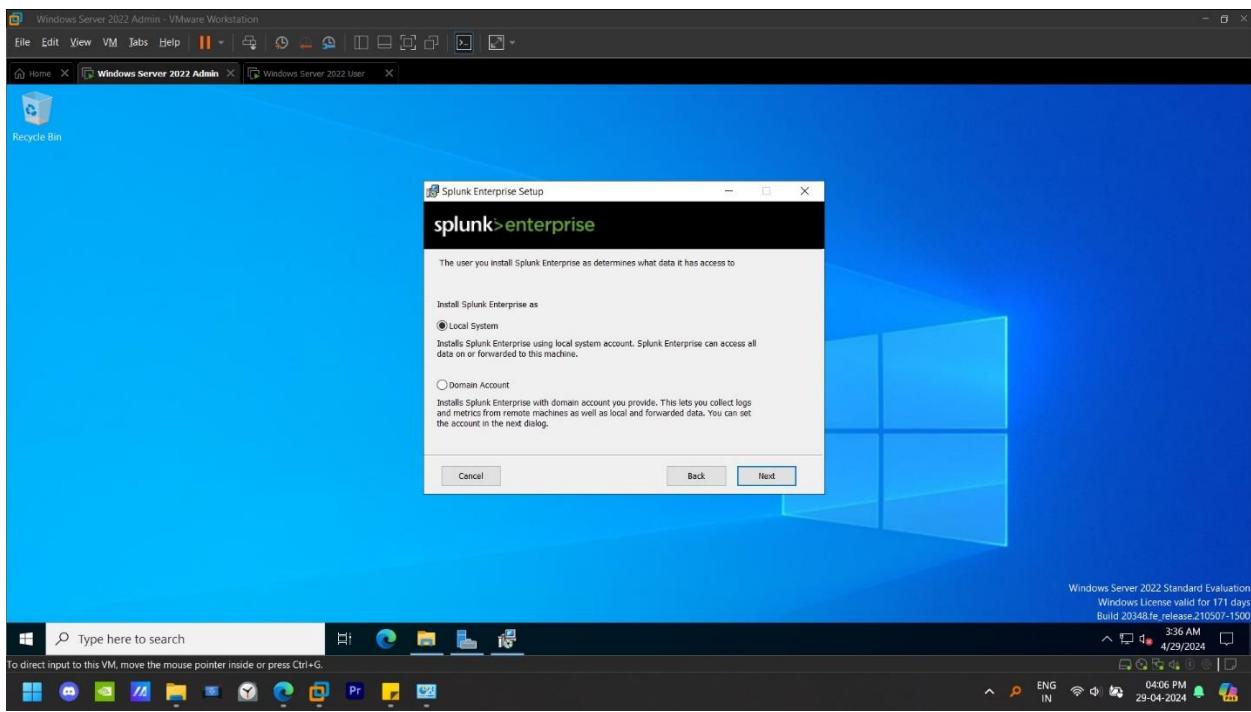
## **CHAPTER: 11 INSTALLING SPLUNK**

## CHAPTER: 11 INSTALLING SPLUNK

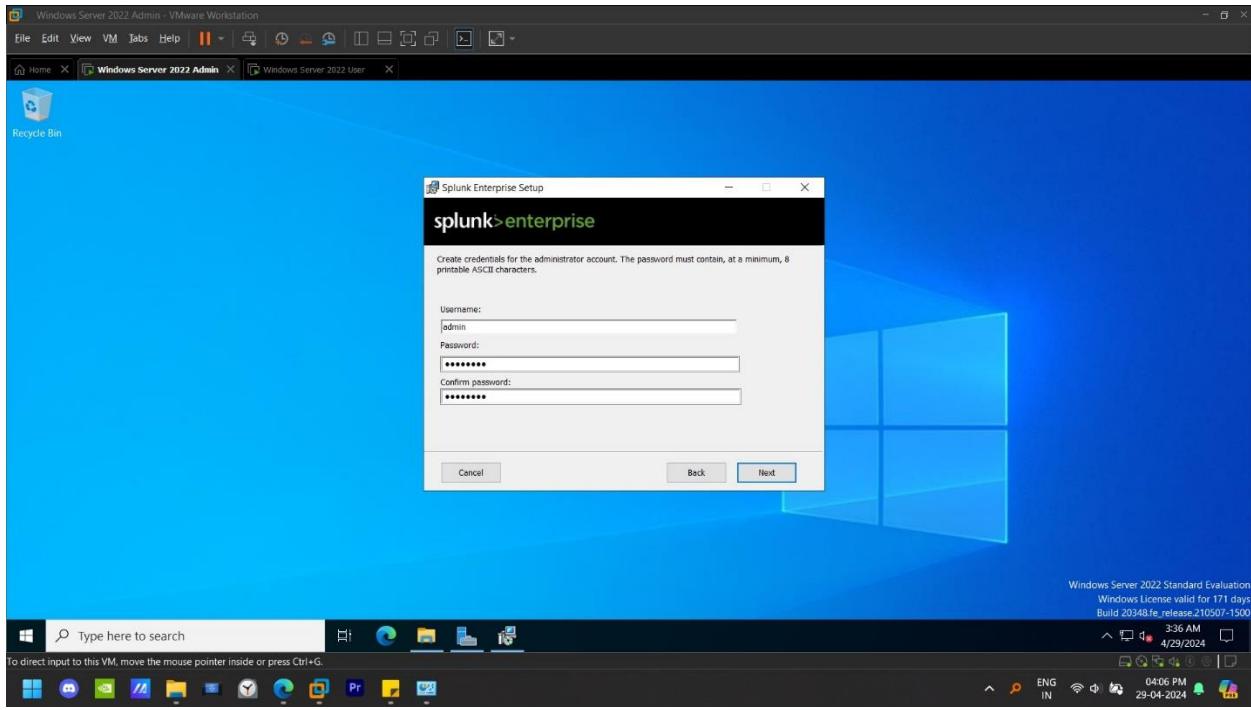
Installing Splunk Enterprise in Windows Server 2022 Admin Account



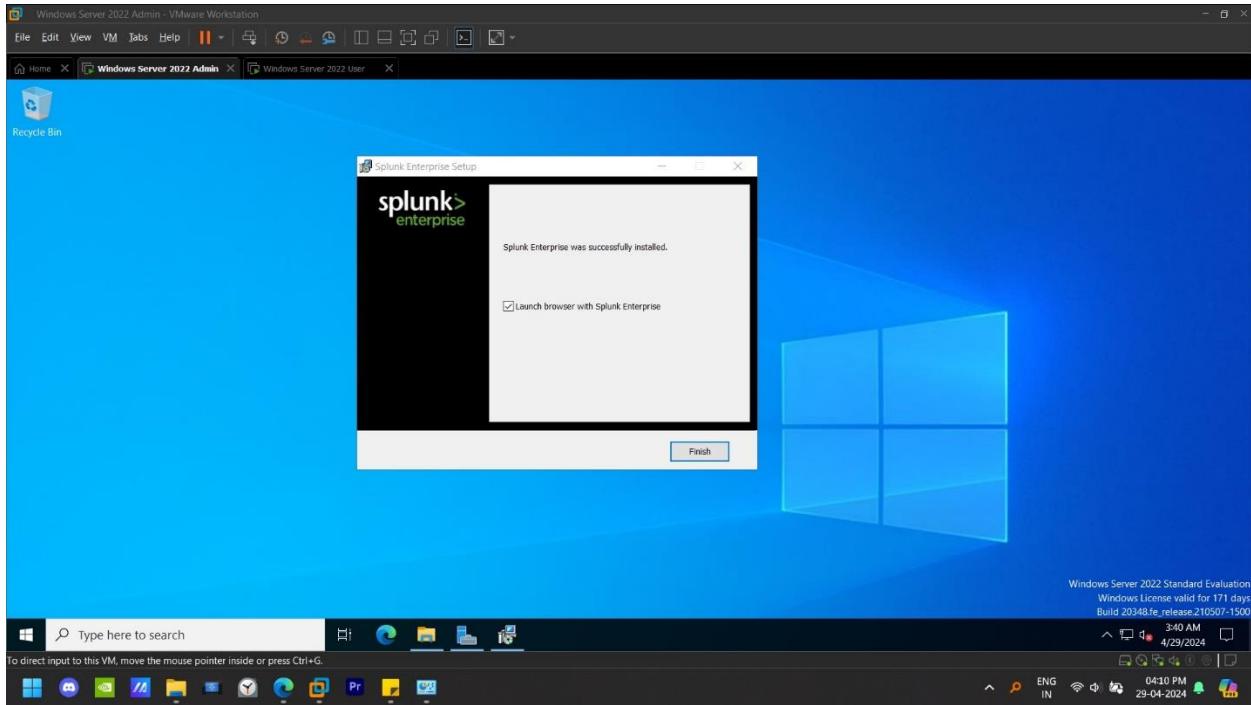
Select Local Account –



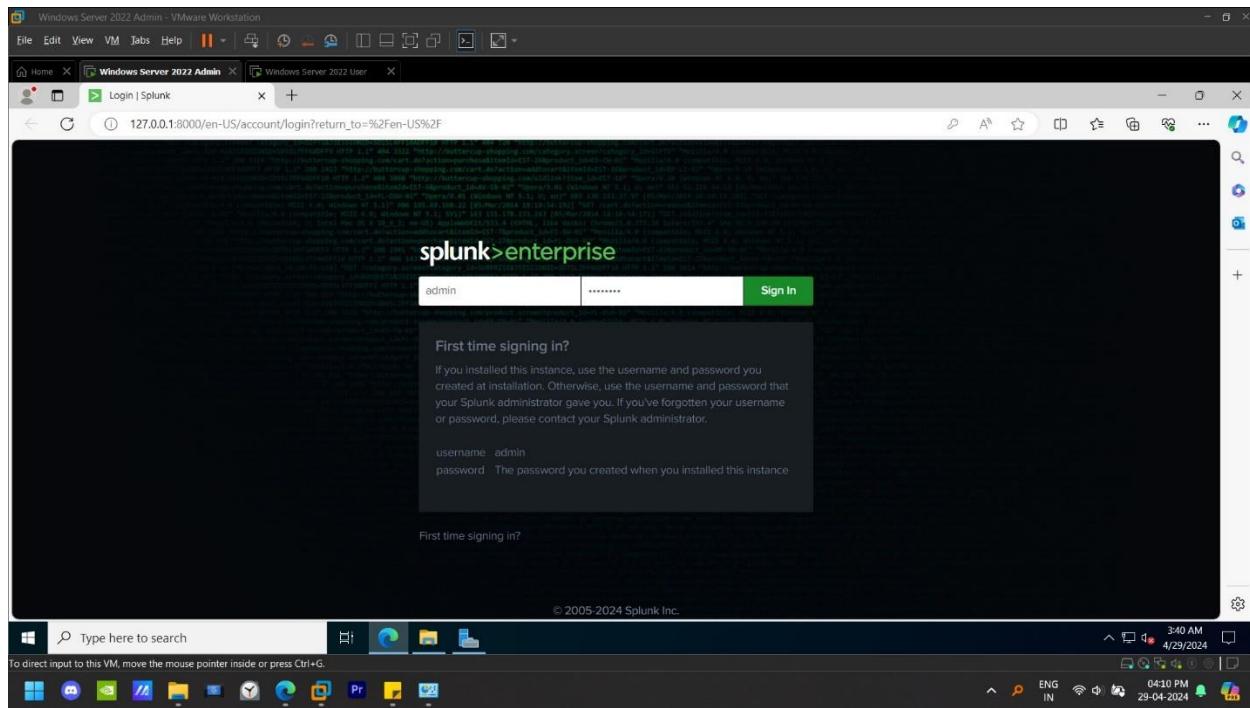
Enter Login credentials –



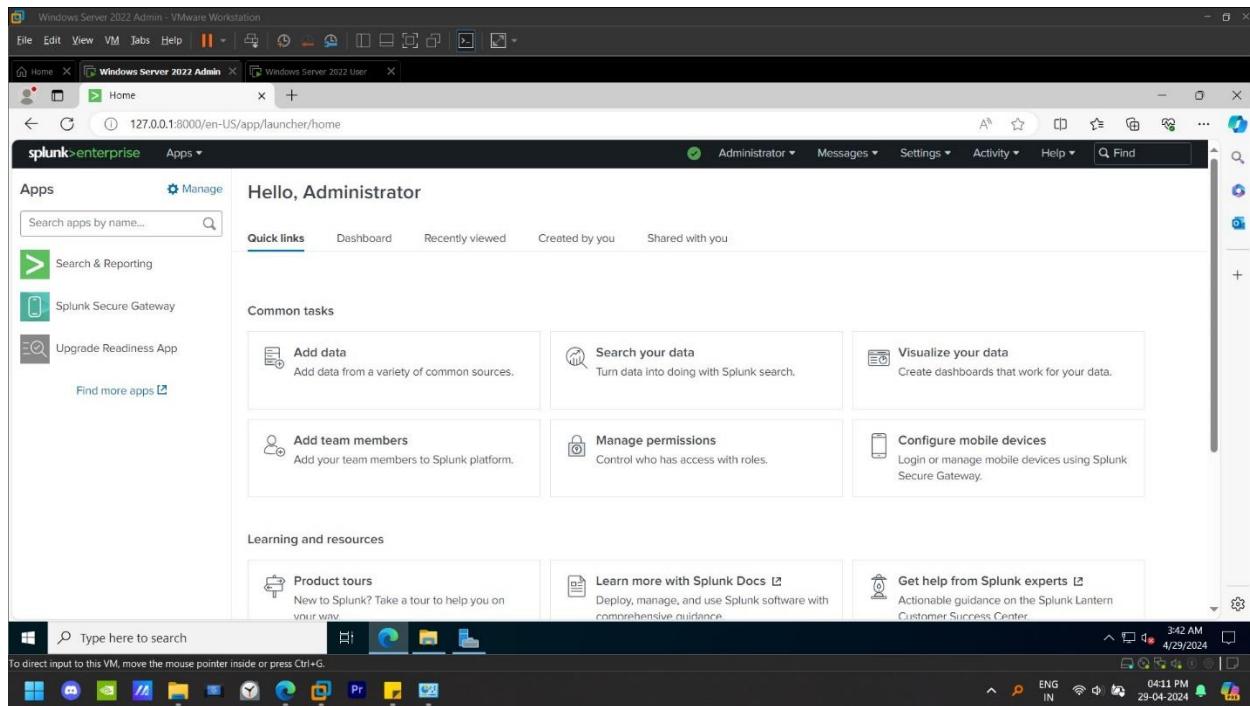
Installation Done –



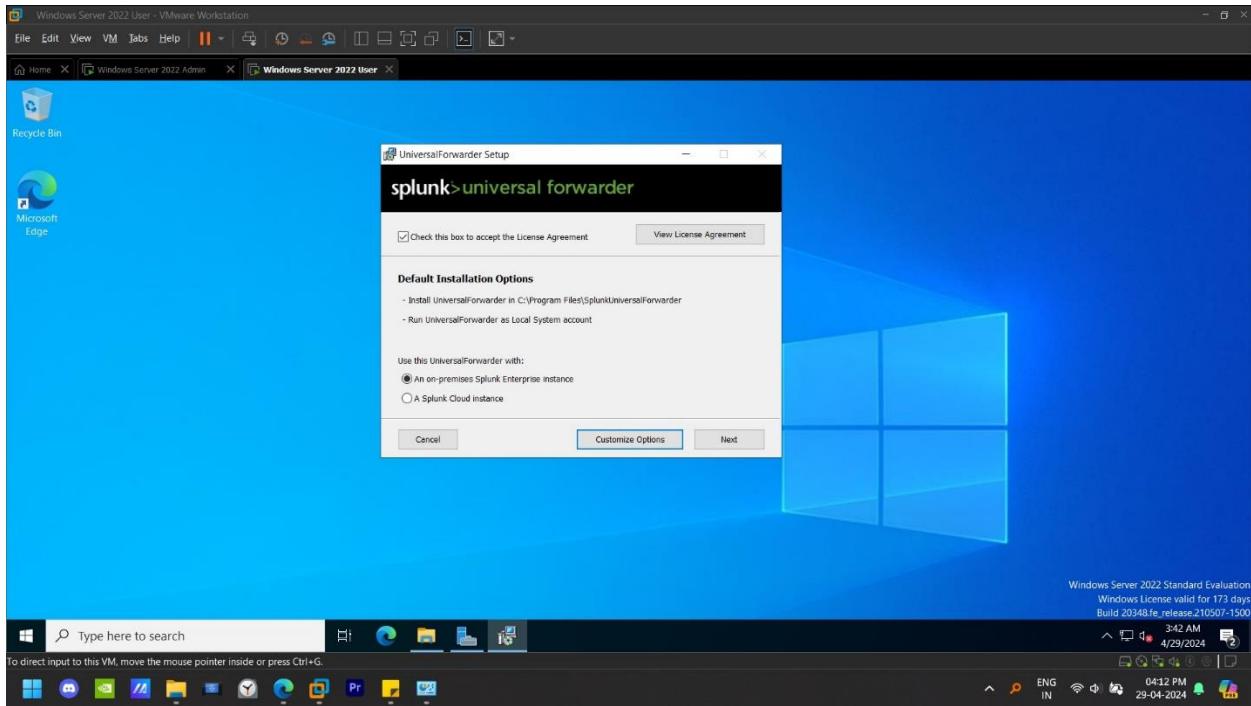
## Open Splunk Enterprise from Start Menu –



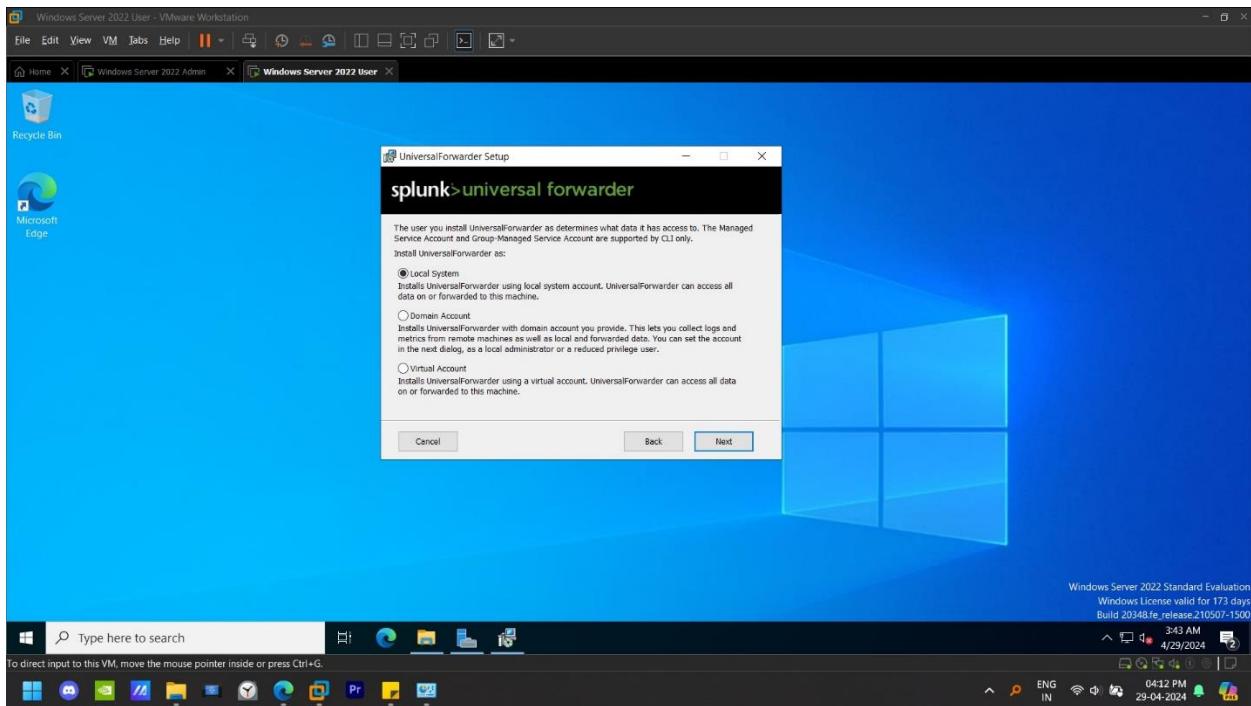
## Enter Login Credentials –



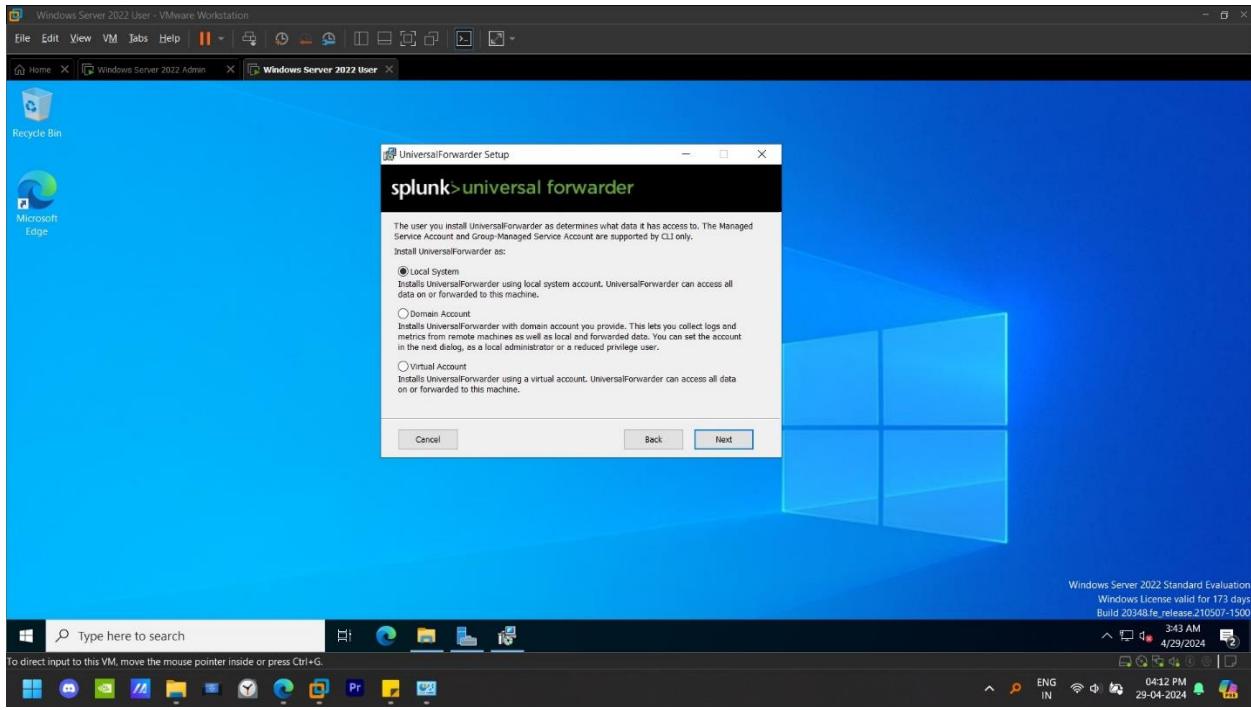
## Installing Splunk Forwarder in Windows Server 2022 User Account –



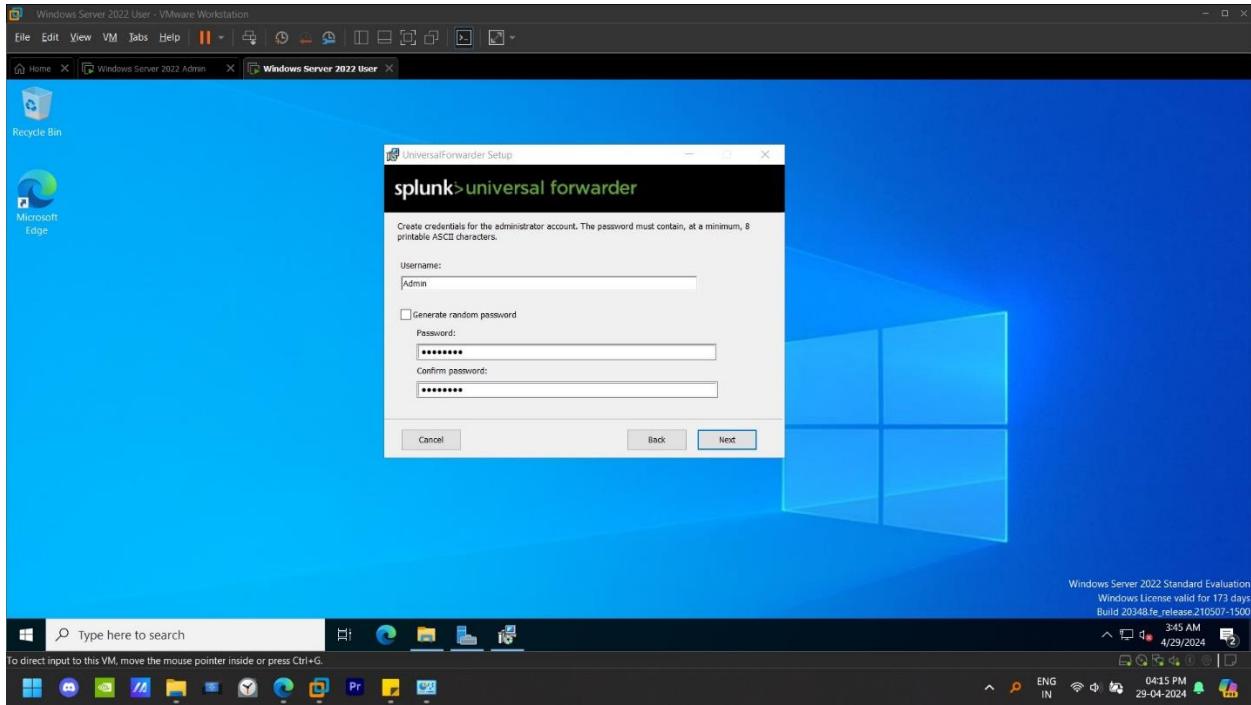
## Select Local System –



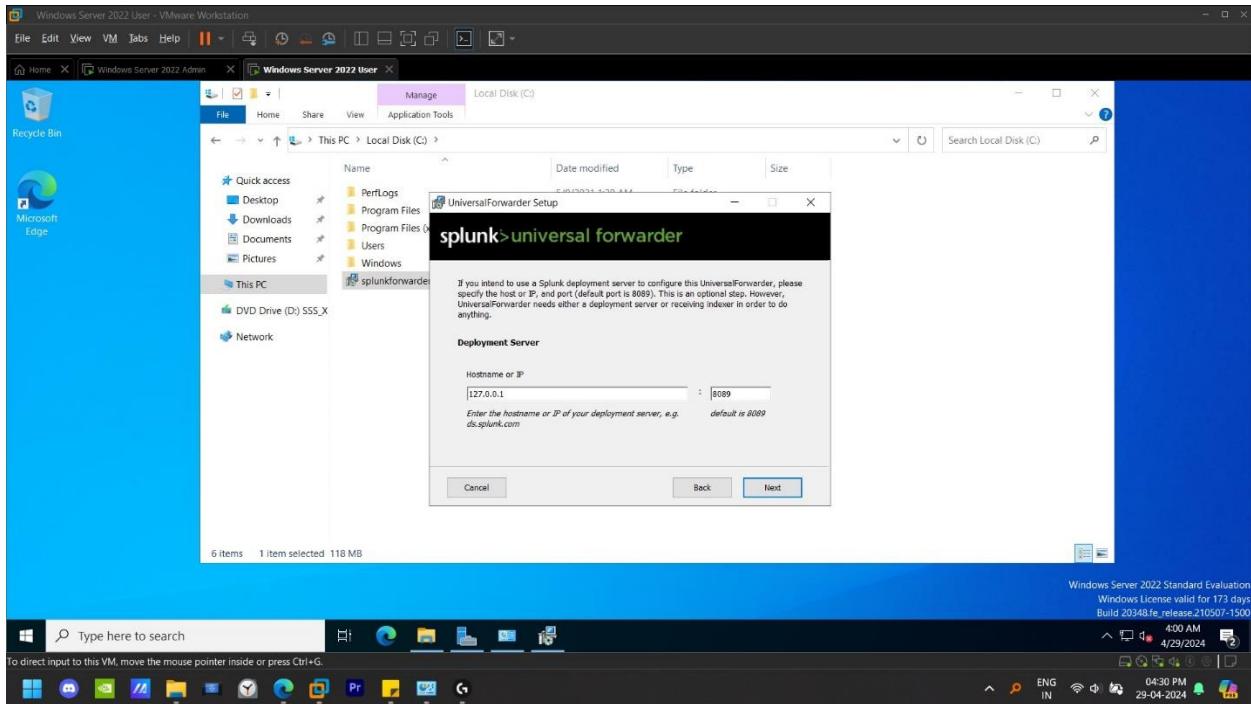
## Select All Logs –



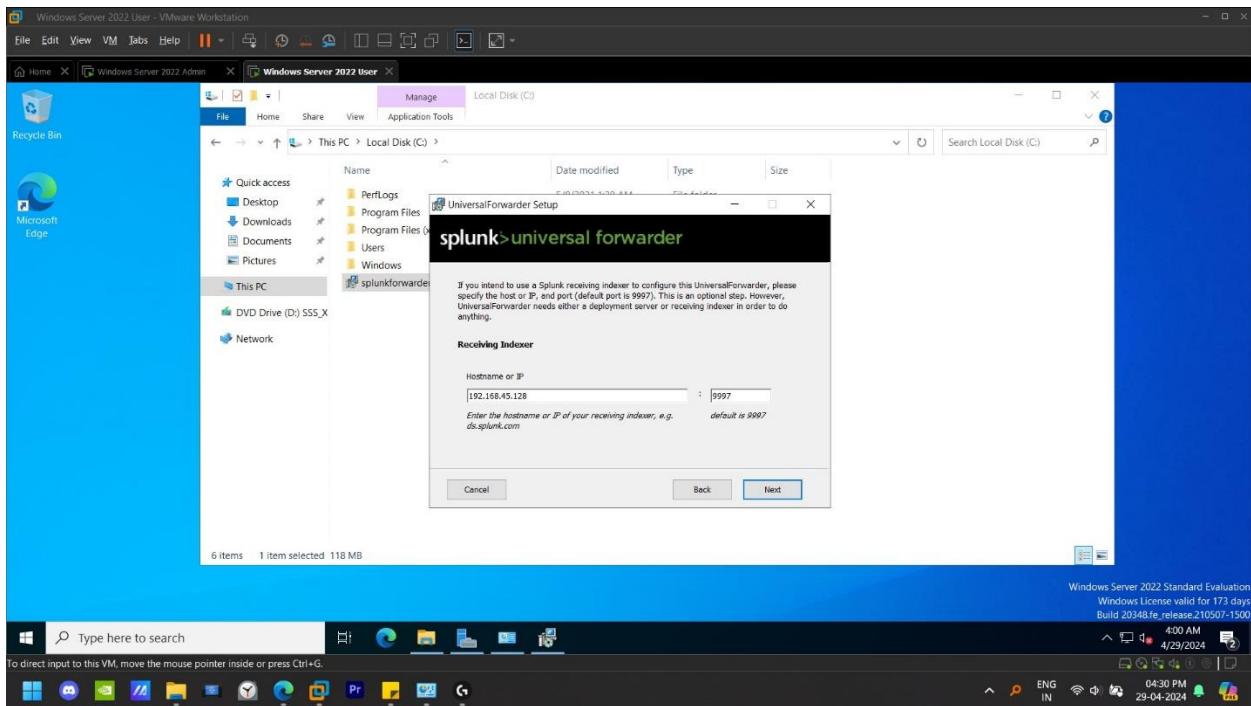
## Add admin Credentials –



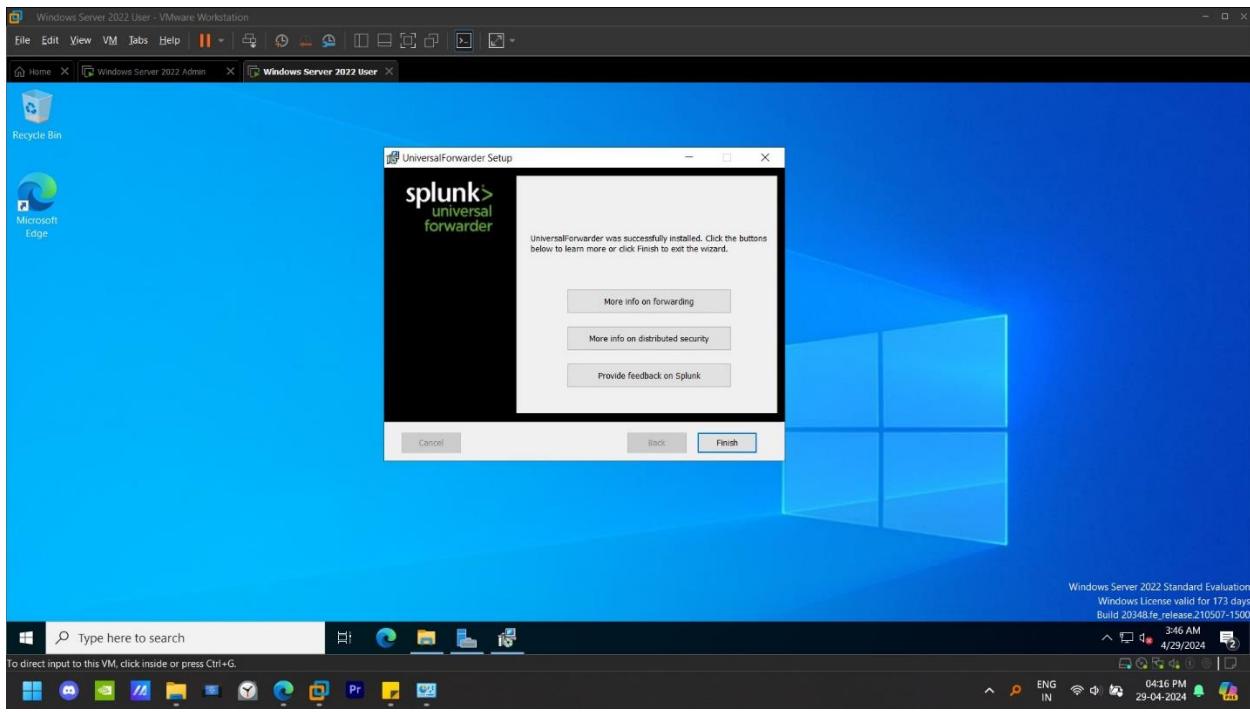
Enter Localhost IP and Port Number(8089) in Deployment Server –



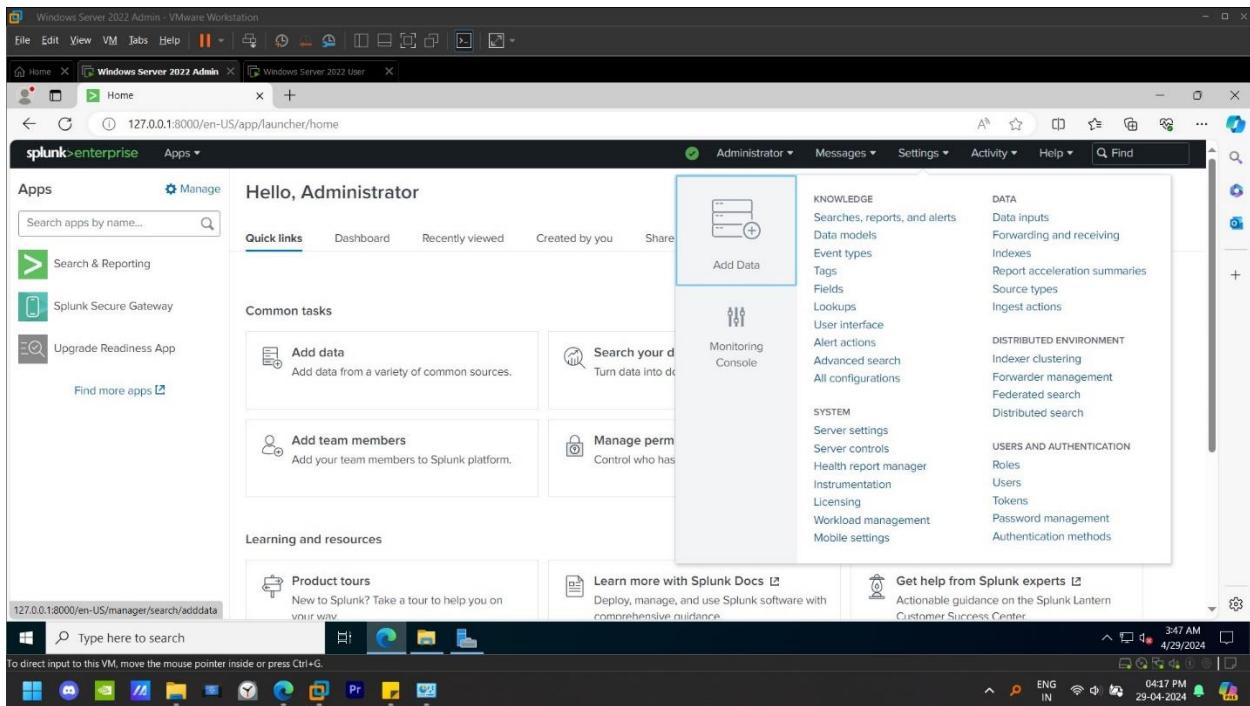
Enter the IP of Enterprise Server Machine and Port Number(9997) in Receiving Indexer –



## Installation Done –



Now Open Splunk and Goto Administrator option and add Forwarding and Receiving Data –

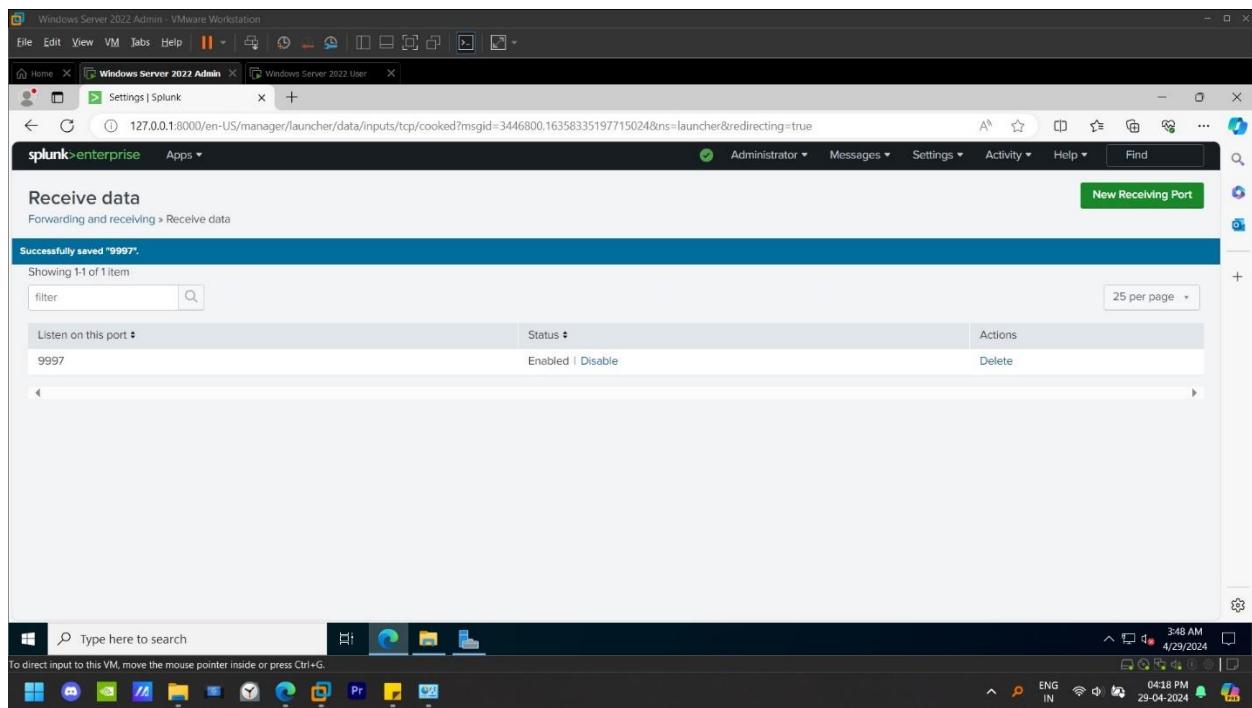


## Click on Add Data in Receive Data –

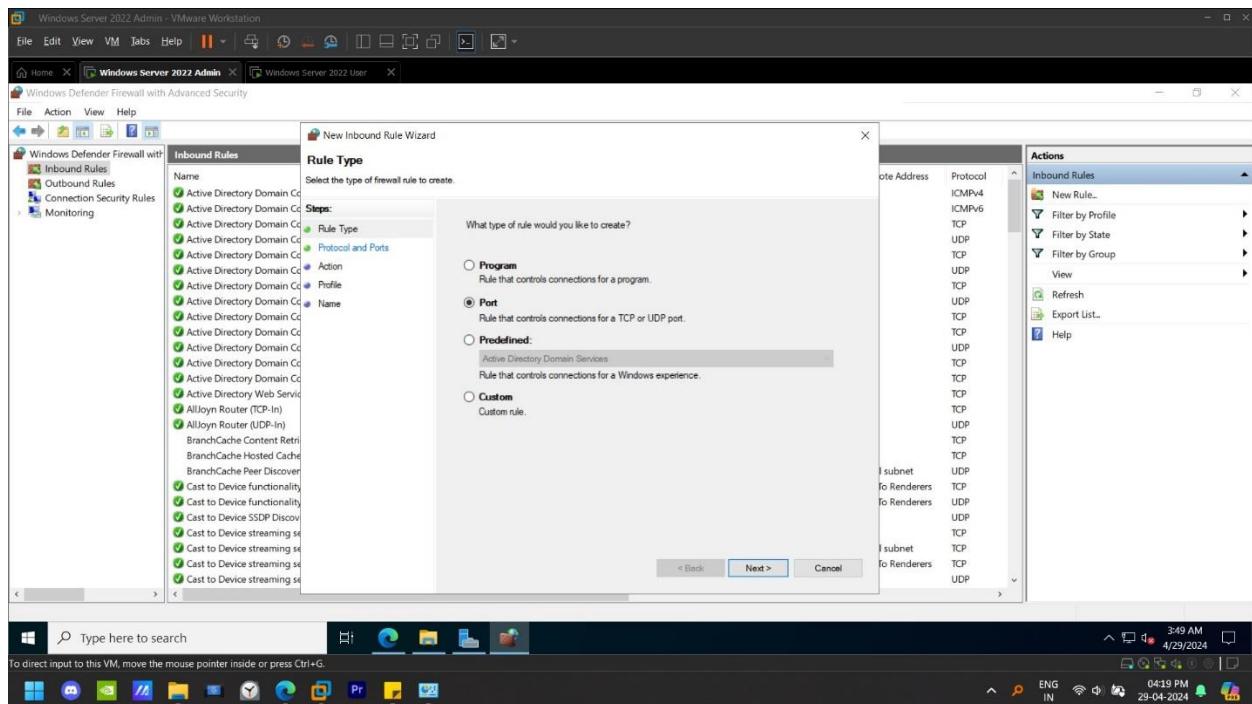
The screenshot shows the Splunk interface on a Windows Server 2022 Admin VM. The main window title is "Windows Server 2022 Admin - VMware Workstation". The URL in the address bar is "127.0.0.1:8000/en-US/manager/launcher/forwardreceive". The page displays the "Forwarding and receiving" configuration. Under the "Receive data" section, there is a table with one row labeled "Configure receiving". A " + Add new" button is located at the bottom right of this table. The taskbar at the bottom shows various application icons and the system tray.

## Add Port 9997 –

The screenshot shows the "Configure receiving" dialog box from the Splunk interface. The title of the dialog is "Add new" and it is part of the "Forwarding and receiving > Receive data > Add new" process. The dialog contains a field labeled "Listen on this port \*" with the value "9997" highlighted. Below the field is a note: "For example, 9997 will receive data on TCP port 9997." At the bottom of the dialog are "Cancel" and "Save" buttons. The taskbar at the bottom shows various application icons and the system tray.

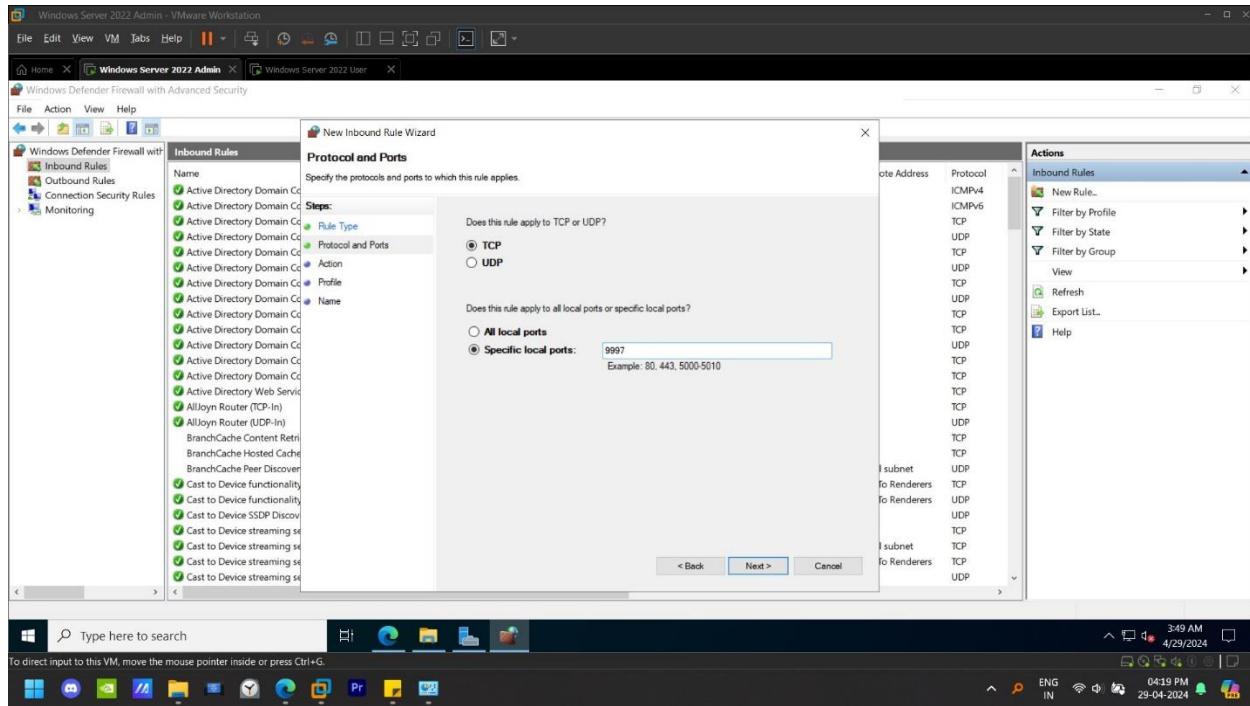


Now goto windows firewall and and create inbound and outbound rule –

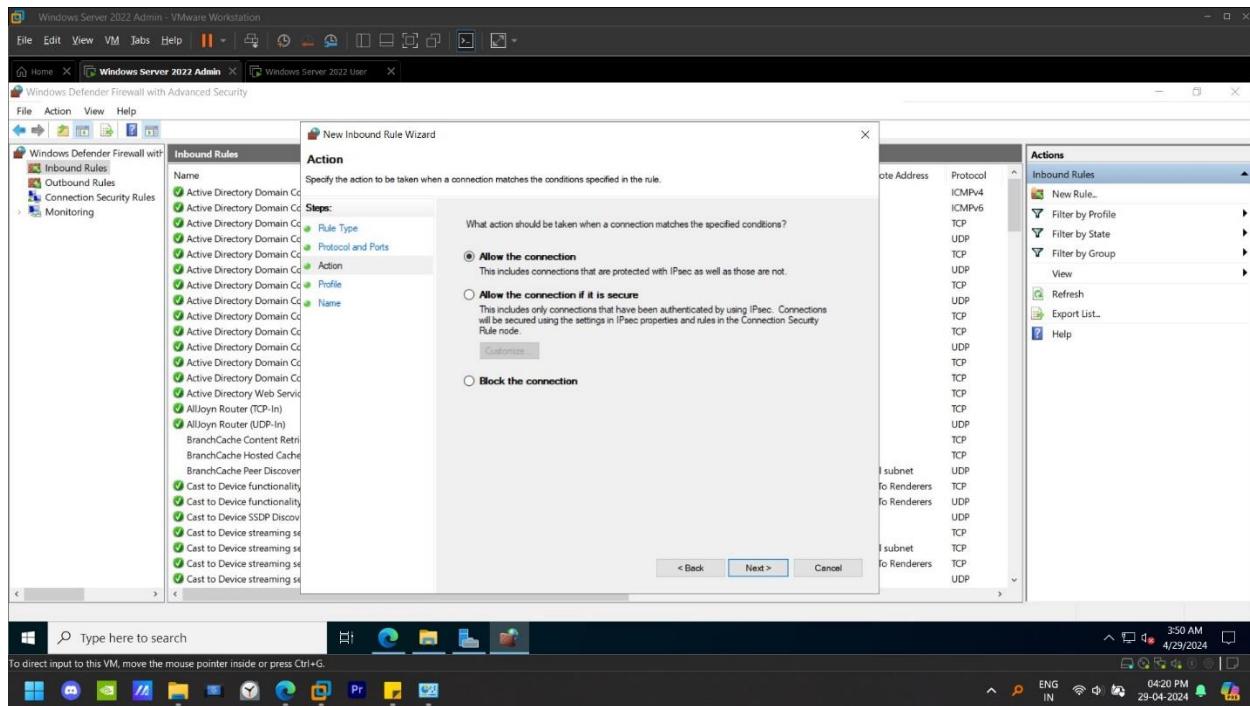


Select Port

Specify the port number 9997.



Allow the connection –



Do same for Outbound Rule in both the Admin Account and user Account.

## Logs of User Account –

The screenshot shows the Splunk 9.2.1 search interface. The search bar at the top contains the query "index='main'". Below the search bar, it says "175 events (4/28/24 4:00:00.000 AM to 4/29/24 4:01:22.000 AM)" and "No Event Sampling". The main pane displays a list of 175 events. The first two events are expanded:

Time	Event
4/29/24 4:01:17.677	dcName=LDAP://WIN-ABFB0HMPB.mydc.local/ adminEventType=schema className=rIDSet classN=RID-Set Show all 142 lines host = WIN-UVNQKJ0H3C7   source = ActiveDirectory   sourcetype = ActiveDirectory
4/29/24 4:01:17.677	dcName=LDAP://WIN-ABFB0HMPB.mydc.local/ adminEventType=schema className=rIDManager

The bottom right corner of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time as 4/29/2024 04:31 PM.

## **CHAPTER: 12 CONCLUSION**

## **CHAPTER: 12 CONCLUSION**

The implementation of Wazuh, coupled with the successful Proof of Concept (POC) and the integration of Shuffle SOAR, marks a significant advancement in our cybersecurity infrastructure. By leveraging multiple operating systems within Wazuh, we have enhanced our ability to detect and respond to a diverse range of threats effectively. Furthermore, the integration of VirusTotal enriches our threat intelligence capabilities, enabling more comprehensive analysis and informed decision-making.

This comprehensive approach not only strengthens our defensive posture but also streamlines incident response processes. The successful integration of these tools showcases our commitment to proactive cybersecurity measures and underscores our readiness to mitigate emerging threats effectively.

Moving forward, continual refinement and optimization of these integrations will be essential to ensure ongoing efficacy and adaptability in the face of evolving cyber threats. Additionally, exploring further synergies between Wazuh and other complementary security solutions could yield additional benefits, enhancing our overall resilience against cyberattacks.

Overall, the successful deployment and integration of Wazuh, coupled with the strategic utilization of additional tools and resources, position us well to navigate the dynamic landscape of cybersecurity threats confidently.

## **CHAPTER: 13 REFRENCE**

## **CHAPTER: 13 REFERENCES**

1. <https://docs.splunk.com/Documentation/Splunk/latest/Admin/Inputsconf>
2. <https://documentation.wazuh.com/current/getting-started/index.html>
3. <https://learn.microsoft.com/en-us/windows-server/>
4. <https://www.youtube.com/watch?v=6ig5vTzME20&pp=ygUVc2V0dGluZyB1cCB3aW5kb3dzIGFk>