

LINUX GUARDIAN: NAGIOS-POWERED HOST MONITORING

Security Operations

CDAC, Noida

**CYBER GYAN VIRTUAL INTERNSHIP
PROGRAM**

Submitted By:

Ayush Roy

Project Trainee, (May-June) 2024

BONAFIDE CERTIFICATE

This is to certify that this project report entitled **Linux Guardian: Nagios-Powered Host Monitoring** submitted to CDAC Noida, is a Bonafede record of work done by **Ayush Roy** under my supervision from **01/06/2024** to **25/06/2024**.

HEAD OF THE DEPARTMENT

SUPERVISOR

Declaration by Author

This is to declare that this report has been written by me. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. I aver that if any part of the report is found to be plagiarized, I shall take full responsibility for it.

Name of Author: **Ayush Roy**

TABLE OF CONTENTS

Acknowledgement

1. Introduction	1
1.1 Problem statement	1
1.2 Learning Objectives	1
2. Theoretical Foundation	2
2.1 Approach	2
Tools and Technologies	2
Infrastructure Created	2
Infrastructure Diagram	3
3. Implementation	4
3.1 Nagios Core Installation	4
3.2 NCPA Installation	5
3.3 Configuration	5
3.4 Web Setup	5
3.5 Monitoring	7
4. Conclusion	9
4.1 Findings	9
4.2 Counter-measures	9

List of References

ACKNOWLEDGEMENT

My heartfelt appreciation to the Indian government for launching and supporting the "**Cyber Gyan Virtual Internship Program.**" I have had the wonderful opportunity to improve my cybersecurity and IT infrastructure management knowledge and abilities thanks to this program.

I want to express my sincere gratitude to **Jyoti Pathak**, my mentor, for her constant leadership, encouragement, and wise counsel during this endeavour. Their knowledge and support have been essential to my growth as a learner and the accomplishment of this assignment.

Additionally, I would like to thank **CDAC Noida** for offering the tools and assistance that I needed to complete my assignment. I would especially want to thank my classmates and the technical team for their help and collaboration throughout this internship. Their assistance has been invaluable to me on this journey.

Linux Guardian: Nagios-Powered Host Monitoring

PROBLEM STATEMENT:

Sustaining optimal performance, high availability, and strong security requires effective administration and monitoring of Linux servers. Administrators may find it difficult to recognize and resolve problems quickly in the absence of a dependable system, which might result in downtime, decreased performance, and security risks. By using Nagios for real-time monitoring and control of Linux hosts, this project seeks to address these issues by guaranteeing proactive issue detection and prompt resolution to preserve a stable and dependable infrastructure.

LEARNING OBJECTIVES:

1. **Understanding Nagios Architecture:** Gain a comprehensive understanding of the Nagios architecture and its components, including plugins, and configuration files.
2. **Installing and Configuring Nagios:** Learn how to install Nagios on a server and configure it to monitor various Linux system metrics, services, and applications.
3. **Real-time Monitoring:** Develop skills to set up real-time monitoring of Linux hosts to track performance, availability, and security metrics.
4. **Alerting and Notifications:** Learn to configure alerting mechanisms in Nagios to receive timely notifications about potential issues and system anomalies.
5. **Proactive Issue Detection:** Acquire the ability to use Nagios for proactive detection and diagnosis of issues to prevent downtime and ensure system stability.
6. **Performance Optimization:** Understand how to utilize Nagios data to optimize the performance and reliability of Linux-based infrastructure.
7. **Security Monitoring:** Gain insights into monitoring security aspects of Linux servers using Nagios, including detecting unauthorized access and potential vulnerabilities.
8. **Troubleshooting and Maintenance:** Develop troubleshooting skills to resolve issues in Nagios configurations and maintain the monitoring setup efficiently.

APPROACH:

Tools and Technologies Used

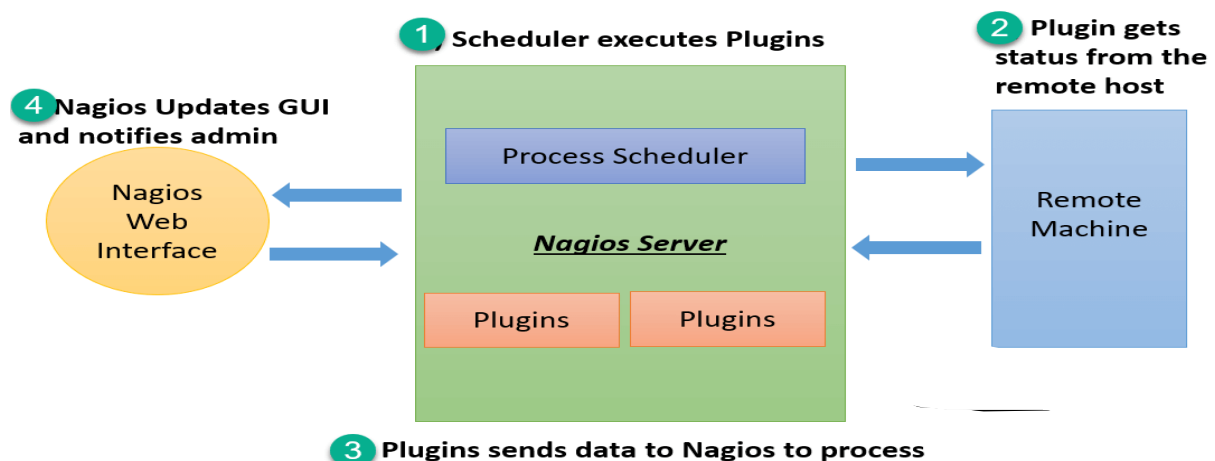
- **Nagios:** For monitoring system metrics, services, and applications.
- **Kali Linux VM:** The platform on which Nagios is installed and configured.
- **NCPA (Nagios Cross-Platform Agent):** To monitor specific metrics and services.
- **Apache:** Web server to host the Nagios web interface.
- **SSH:** Securely connecting to remote Linux hosts for configuration and monitoring.
- **Firewalls:** Configured to allow Nagios and NRPE traffic.

Infrastructure Created

The infrastructure setup involves the following components:

1. **Kali Linux Virtual Machine:**
 - **Operating System:** Kali Linux
 - **IP Address:** 10.0.2.15
 - **Roles:** Nagios server, Apache server
2. **Monitored Linux Hosts:**
 - **Host 1:**
 - **Operating System:** Kali Linux Server
 - **IP Address:** 127.0.0.1
 - **Roles:** NCPA agent installed for monitoring
3. **Network Components:**
 - **Firewall:** Configured to allow HTTP (port 80), HTTPS (port 443), and SSH (port 22) traffic to the Nagios server and monitored hosts.
 - **Router/Switch:** Manages network traffic between the Nagios server and monitored hosts.

Diagram Depicting Infrastructure



IMPLEMENTATION:

Step 1: Installing Nagios on Linux

- **Install Required Dependencies using the command:**

```
sudo apt install -y wget build-essential apache2 php openssl perl make gcc libc6  
libmcrypt-dev libssl-dev bc gawk dc build-essential snmp libnet-snmp-perl gettext
```

- **Create a Nagios User and Group using the command:**

```
sudo useradd nagios  
sudo usermod -a -G nagios www-data
```

- **Download and Install Nagios Core using the command:**

```
wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.6.tar.gz  
tar -xzf nagios-4.4.6.tar.gz  
cd nagios-4.4.6/  
./configure --with-httpd-conf=/etc/apache2/sites-enabled  
make all  
sudo make install  
sudo make install-init  
sudo make install-commandmode  
sudo make install-config  
sudo make install-webconf  
sudo a2enmod rewrite cgi
```

- **Set Nagios Web Interface Password using the command:**

```
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin  
sudo systemctl restart apache2
```

Option to type the password will appear
Type the password and hit Enter

Step 2: Installing Nagios Plugins

- **Download and Install Nagios Plugins using the command:**

```
wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
```

```
tar -xzf nagios-plugins-2.3.3.tar.gz
```

```
cd nagios-plugins-2.3.3/
```

```
./configure
```

```
make
```

```
sudo make install
```

Step 3: Installing NRPE (Nagios Remote Plugin Executor)

- **Download and Install NRPE**

```
wget https://github.com/NagiosEnterprises/nrpe/releases/download/nrpe-4.0.3/nrpe-4.0.3.tar.gz
```

```
tar -xzf nrpe-4.0.3.tar.gz
```

```
cd nrpe-4.0.3/
```

```
./configure --enable-command-args
```

```
make all
```

```
sudo make install-groups-users
```

```
sudo make install
```

```
sudo make install-config
```

```
sudo make install-init
```

- **Configure NRPE**

Edit the NRPE configuration file `/usr/local/nagios/etc/nrpe.cfg` to allow the Nagios server to communicate:

```
sudo vi /usr/local/nagios/etc/nrpe.cfg
```

Add the IP address of your Nagios server:

```
allowed_hosts=127.0.0.1,::1,10.0.2.15
```

Restart NRPE service:

```
sudo systemctl start nrpe.service
```

```
sudo systemctl enable nrpe.service
```

Step 4: Configuring Nagios for HTTP and SSH Monitoring

- Edit Nagios Configuration Files

Edit the `localhost.cfg` file to include HTTP and SSH monitoring:

```
sudo vi /usr/local/nagios/etc/objects/localhost.cfg
```

Add the following configurations:

```
define service{
    use            local-service
    host_name      localhost
    service_description HTTP
    check_command  check_http
}
```

```
define service{
    use            local-service
    host_name      localhost
    service_description SSH
    check_command  check_ssh
}
```

Summary	localhost	Current Load	OK	06-19-2024 06:18:14	3d 18h 22m 14s	1/4	OK - load average: 0.12, 0.15, 0.17
# Define a service to check SSH on the local machine.				06-19-2024 06:18:58	3d 18h 28m 36s	1/4	USERS OK - 5 users currently logged in
# Disable notifications for this service by default, as not all users may have SSH enabled.				06-19-2024 06:19:41	3d 18h 29m 19s	1/4	TP OK: HTTP/1.1 200 OK - 10975 bytes in 0.002 second response time
define service {		Nagios Agent Version	OK	06-19-2024 06:16:47	0d 2h 12m 6s	1/4	OK: Agent_version was [3.1.0]
use	local-service						PING OK - Packet loss = 0%, RTA = 0.09 ms
host_name	localhost		OK	06-19-2024 06:16:12	0d 18h 25m 21s	1/4	
service_description	SSH						DISK OK - free space: / 60721 MB (79.85% inode=91%):
check_command	check_ssh		OK	06-19-2024 06:16:55	3d 18h 24m 44s	1/4	SSH OK - OpenSSH_9.7p1 Debian-5 (protocol 2.0)
notifications_enabled	0	SSH	OK	06-19-2024 06:17:41	0d 9h 1m 12s	1/4	SWAP OK - 73% free (747 MB out of 1023 MB)
							PROCS OK: 51 processes with STATE =
							REPORT
# Define a service to check HTTP on the local machine.				06-19-2024 06:17:30	3d 18h 22m 51s	1/4	
# Disable notifications for this service by default, as not all users may have HTTP enabled.							
define service {		Results 1 - 9 of 9 Matching Services					
use	local-service						; Name of service template to use
host_name	localhost						
service_description	HTTP						
check_command	check_http						
notifications_enabled	0						

- **Verify Nagios Configuration**

```
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

```

File Actions Edit View Help
(kali@kali)-[~]
└─$ sudo -i
[sudo] password for kali:
(kali@kali)-[~]
└─# sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

Nagios
Last Updated: Wed Jun 19 06:20:23 EDT 2024
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org

Nagios Core 4.4.6
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2020-04-28
License: GPL
Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay... 100%
Warning: Duplicate definition found for service 'NCPA Agent Version' on host 'localhost' (config file '/usr/local/nagios/etc/objects/localhost.cfg', starting on line 162)
  Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
  Checked 9 services.
  Checked 1 hosts.
  Checked 1 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 25 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 1 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check

(kali@kali)-[~]
└─#

```

If there are no errors, We proceed further

- **Restart Nagios Service**

```
sudo systemctl restart nagios.service
```

Step 5: Monitoring NCPA Agent Version

- Install NCPA on Client Machine

```
sudo apt update
```

```
sudo apt install -y ncpa
```

If it doesn't work with this command, we can manually install NCPA from the website and then extract it

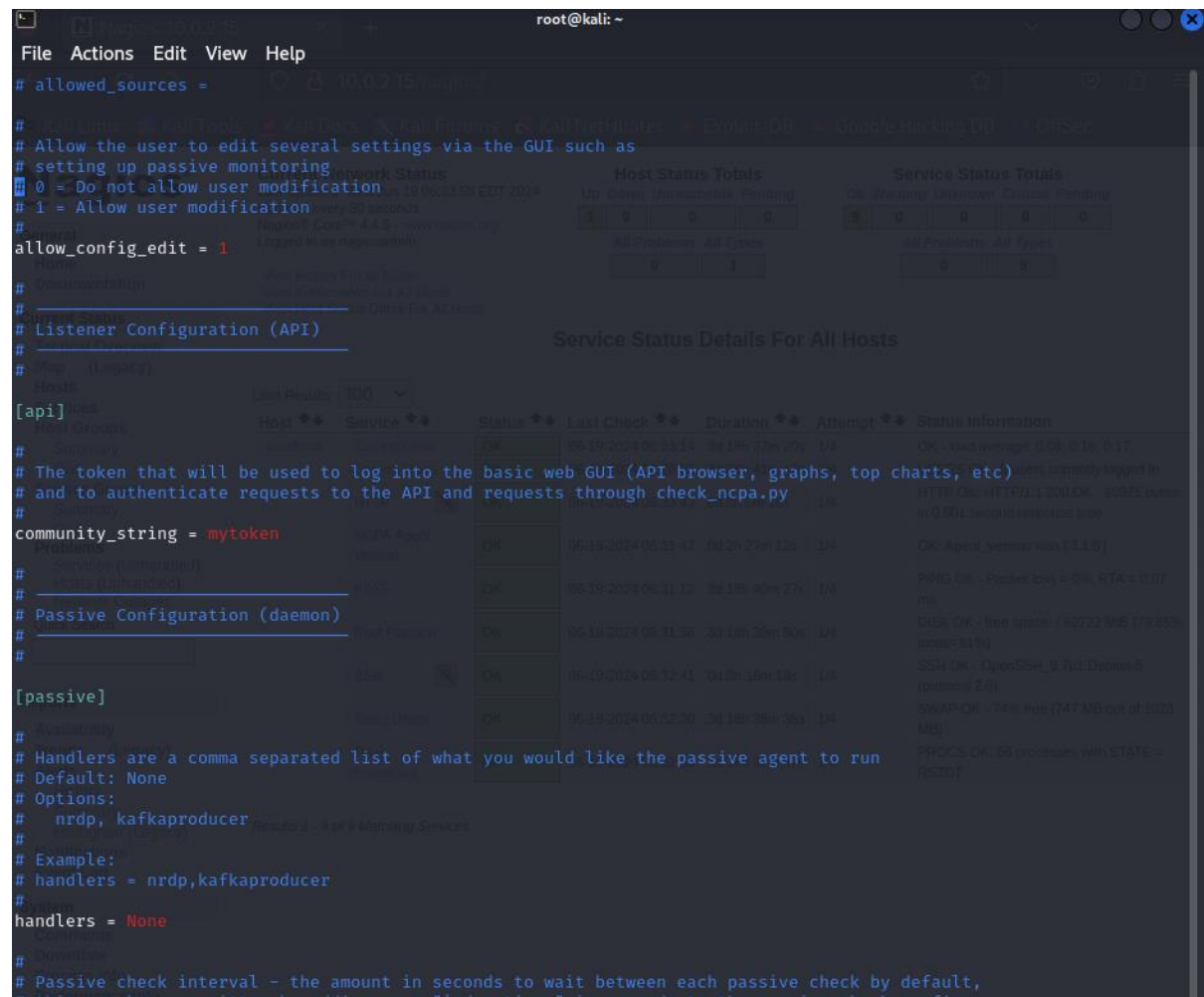
- Configure NCPA

```
sudo vi /usr/local/ncpa/etc/ncpa.cfg
```

Set the string for secure communication:

```
[api]
```

```
Community_string = mytoken
```



```
root@kali: ~  
File Actions Edit View Help  
# allowed_sources =  
#  
# Allow the user to edit several settings via the GUI such as  
# setting up passive monitoring  
# 0 = Do not allow user modification  
# 1 = Allow user modification  
allow_config_edit = 1  
#  
# Listener Configuration (API)  
#  
# Top: Disabled  
# Hosts: 100  
[api]  
# The token that will be used to log into the basic web GUI (API browser, graphs, top charts, etc) and to authenticate requests to the API and requests through check_ncpa.py  
community_string = mytoken  
#  
# Passive Configuration (daemon)  
#  
[passive]  
# Handlers are a comma separated list of what you would like the passive agent to run  
# Default: None  
# Options:  
#   nrdp, kafkaproducer  
# Example:  
# handlers = nrdp,kafkaproducer  
handlers = None  
#  
# Passive check interval - the amount in seconds to wait between each passive check by default,  
# this can be overridden by adding on a "duration" in seconds to the passive check config
```

Restart NCPA service:

```
sudo systemctl restart ncpa_listener.service
```

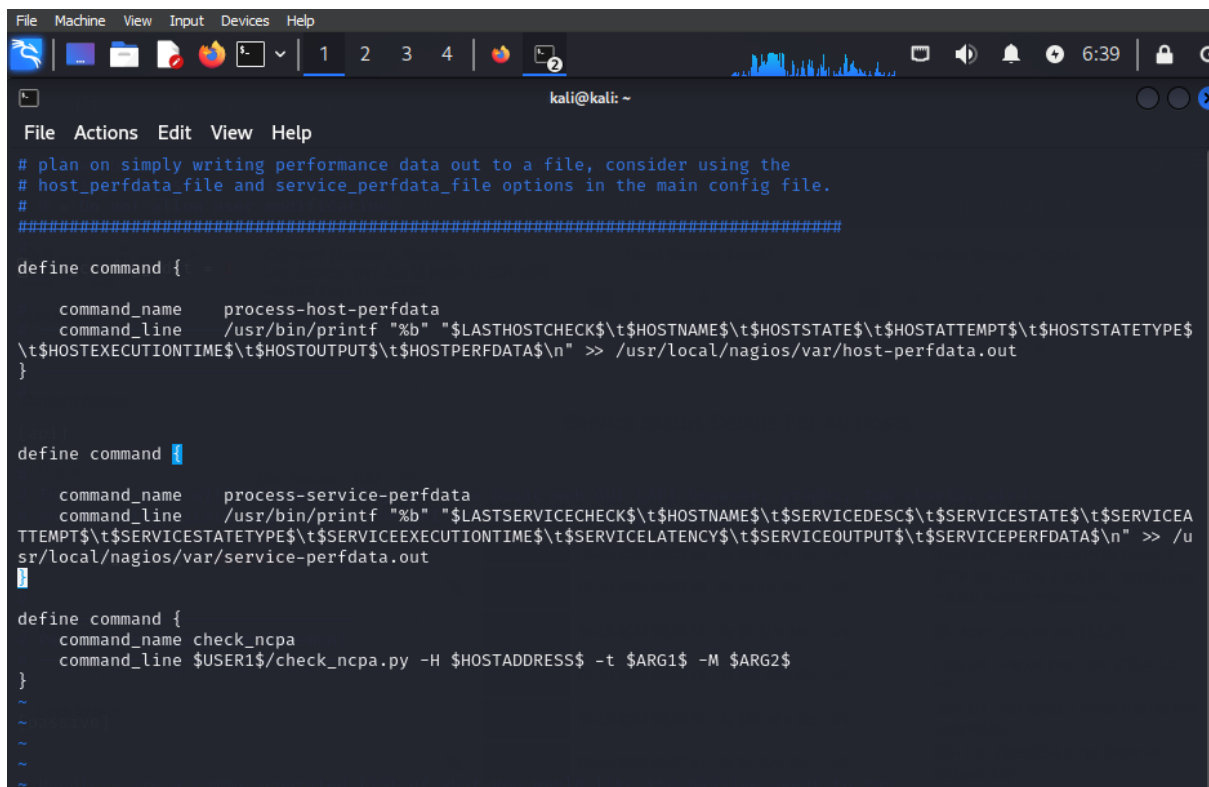
- **Configure Nagios to Monitor NCPA**

Edit `commands.cfg` to define the check command:

```
sudo vi /usr/local/nagios/etc/objects/commands.cfg
```

Add the following command definition:

```
define command{
    command_name check_ncpa
    command_line $USER1$/check_ncpa.py -H $HOSTADDRESS$ -t 'mytoken' -M $ARG1$
}
```



```
File Machine View Input Devices Help
kali@kali: ~
File Actions Edit View Help
# plan on simply writing performance data out to a file, consider using the
# host_perfdata_file and service_perfdata_file options in the main config file.
#
#####
define command {
    command_name    process-host-perfdata
    command_line    /usr/bin/printf "%b" "$LASTHOSTCHECK$\t$HOSTNAME$\t$HOSTSTATE$\t$HOSTATTEMPT$\t$HOSTSTATETYPE$\t$HOSTEXECUTIONTIME$\t$HOSTOUTPUT$\t$HOSTPERFDATA$\n" >> /usr/local/nagios/var/host-perfdata.out
}

define command {
    command_name    process-service-perfdata
    command_line    /usr/bin/printf "%b" "$LASTSERVICECHECK$\t$HOSTNAME$\t$SERVICEDESC$\t$SERVICESTATE$\t$SERVICEATTEMPT$\t$SERVICESTATETYPE$\t$SERVICEEXECUTIONTIME$\t$SERVICELATENCY$\t$SERVICEOUTPUT$\t$SERVICEPERFDATA$\n" >> /usr/local/nagios/var/service-perfdata.out
}

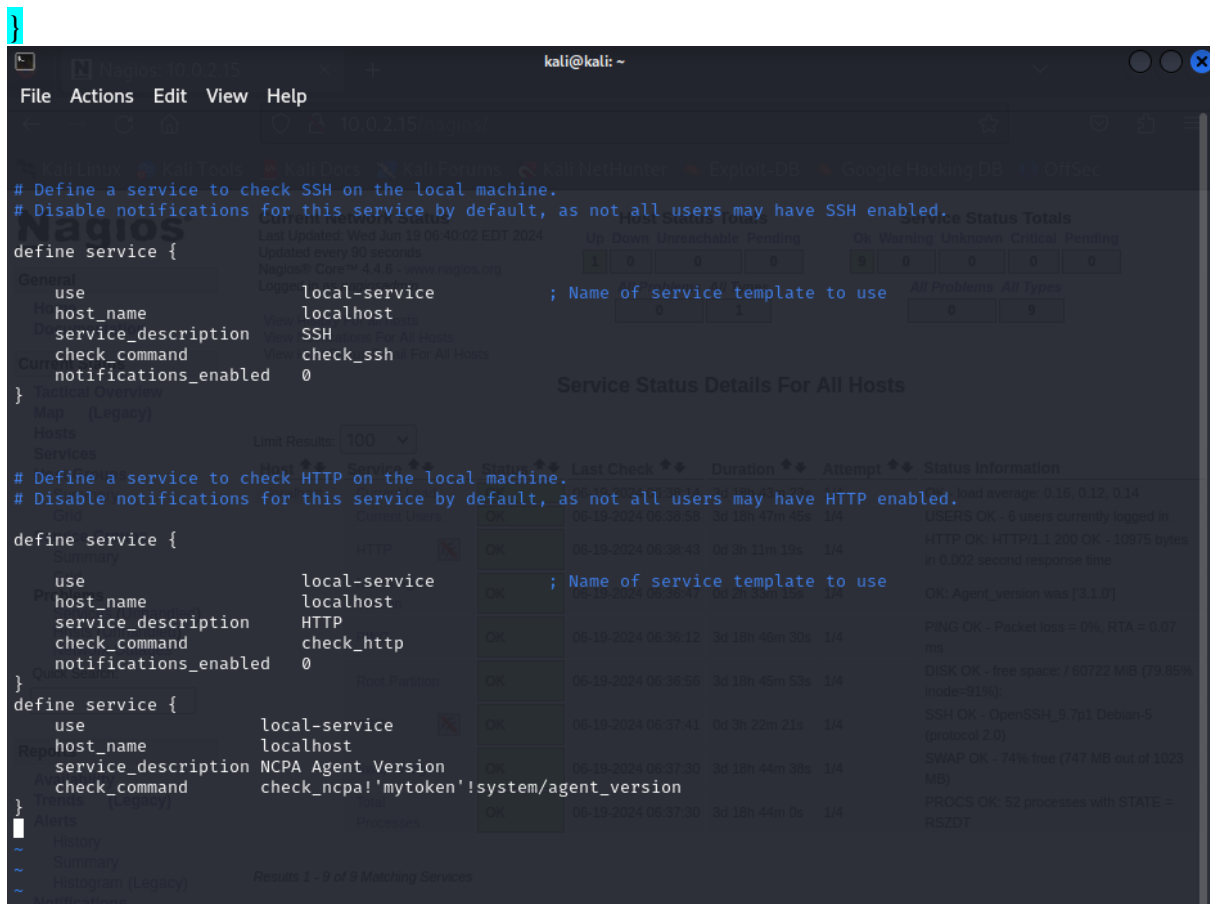
define command {
    command_name    check_ncpa
    command_line    $USER1$/check_ncpa.py -H $HOSTADDRESS$ -t $ARG1$ -M $ARG2$
}
~
~
~
~
```

Edit `localhost.cfg` to add NCPA service checks:

```
sudo vi /usr/local/nagios/etc/objects/localhost.cfg
```

Add the following service definitions:

```
define service{
    use                local-service
    host_name          localhost
    service_description NCPA Version
    check_command       check_ncpa!-M 'agent/plugin/version'
```

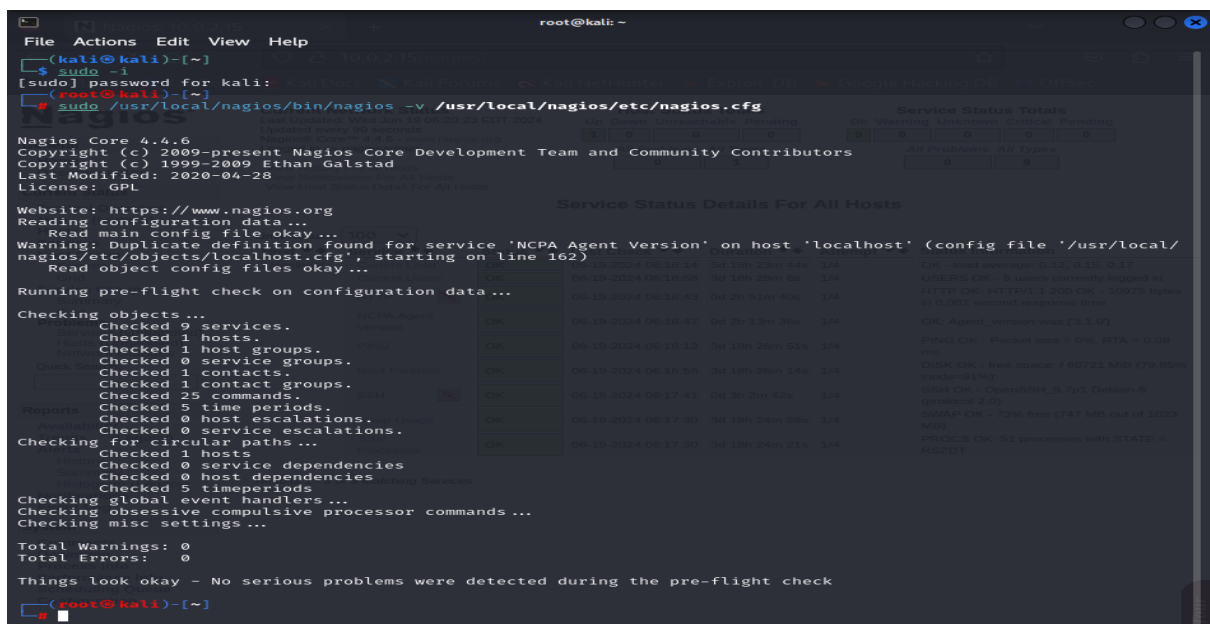


- Verify Nagios Configuration

```
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

- Restart Nagios Service

```
sudo systemctl restart nagios.service
```



THE HOST:

```

root@kali: ~
File Actions Edit View Help
# NOTE: This config file is intended to serve as an *extremely* simple
# example of how you can create configuration entries to monitor the local
# the local (Linux) machine.

##### Current Network Status #####
Updated every 60 seconds
Nagios Core 4.4.0 - www.nagios.org
Licensed as per requirements

##### Host Status Totals #####
3 0 0 0
All Problems All Types

##### Service Status Totals #####
0 0 0 0 0 0
All Problems All Types

##### HOST DEFINITION #####
# Define a host for the local machine
define host {
    use linux-server
    host_name localhost
    alias Local Linux Host
    address 127.0.0.1
    max_check_attempts 5
    check_period 24x7
    notification_interval 30
    notification_period 24x7
}

##### Service Status Details For All Hosts #####
# Define a service for the local machine
define service {
    use linux-server
    service_name localhost
    check_command exec
    max_check_attempts 5
    check_period 24x7
    notification_interval 30
    notification_period 24x7
}

##### Host Group Definition #####
define hostgroup {
    hostgroup_name linux-servers
    alias Linux Servers
    members localhost
}

```

NAGIOS MONITORING SYSTEM WEB SERVER:

The screenshot shows the Nagios Core 4.4.6 web interface. The browser's address bar indicates the URL is 10.0.2.15/nagios/. The page layout includes a left-hand navigation menu with categories such as General, Current Status, Reports, and System. The main content area prominently displays the Nagios Core logo and version number (4.4.6), along with the status 'Daemon running with PID 71580'. A blue banner highlights that a new version (4.5.3) is available for download. Below the banner, there are four main sections: 'Get Started' with a list of bullet points, 'Quick Links' with a list of links, 'Latest News', and 'Don't Miss...'. The footer contains copyright information for 2010-2020 and a license statement.

Summary of Nagios Monitoring Status:

1. Host Status:
- Up: 1 host is up and running.

Down/Unreachable/Pending: 0 hosts in these states, indicating no issues with host availability.
2. Service Status:
- OK: 9 services are running without issues.

Warning/Unknown/Critical/Pending: 0 services in these states, indicating no problems with the monitored services.

Detailed Service Status:

- Current Load: OK - Load average is within acceptable limits.
- Current Users: OK - 6 users are currently logged in.
- HTTP: OK - The HTTP service is responding correctly with a 200 OK status.
- NCPA Agent Version: OK - The NCPA agent is running version 3.1.0.
- PING: OK - No packet loss and minimal round-trip time.
- Root Partition: OK - Sufficient free space on the root partition.
- SSH: OK - The SSH service is responding correctly.
- Swap Usage: OK - 76% of swap space is free.
- Total Processes: OK - 53 processes are running with acceptable states.

Nagios: 10.0.2.15

10.0.2.15/nagios/

Kali LinuxKali ToolsKali DocsKali ForumsKali NetHunterExploit-DBGoogle Hacking DBOffSec

Nagios®

General

HomeDocumentation

Current Status

Tactical OverviewMap (Legacy)HostsServicesHost GroupsSummaryGridService GroupsSummaryGridProblemsServices (Unhandled)Hosts (Unhandled)Network OutagesQuick Search:

Reports

AvailabilityTrends (Legacy)AlertsHistorySummaryHistogram (Legacy)NotificationsEvent Log

System

CommentsDowntimeProcess InfoPerformance InfoScheduling QueueConfiguration

Current Network Status

Last Updated: Wed Jun 19 06:05:16 EDT 2024
Updated every 90 seconds
Nagios® Core™ 4.4.6 - www.nagios.org
Logged in as nagiosadmin

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

UpDownUnreachablePending

100000

All ProblemsAll Types

01

Service Status Totals

OkWarningUnknownCriticalPending

90000

All ProblemsAll Types

09

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	06-19-2024 06:03:14	3d 18h 8m 37s	1/4	OK - load average: 0.37, 0.29, 0.21
	Current Users	OK	06-19-2024 06:03:58	3d 18h 12m 59s	1/4	USERS OK - 5 users currently logged in
	HTTP	OK	06-19-2024 06:03:43	0d 2h 36m 33s	1/4	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.000 second response time
	NCPA Agent Version	OK	06-19-2024 06:01:47	0d 1h 58m 29s	1/4	OK: Agent_version was [3.1.0]
	PING	OK	06-19-2024 06:01:12	3d 18h 11m 44s	1/4	PING OK - Packet loss = 0%, RTA = 0.11 ms
	Root Partition	OK	06-19-2024 06:01:56	3d 18h 11m 7s	1/4	DISK OK - free space: / 60754 MiB (79.89% inode=91%)
	SSH	OK	06-19-2024 06:02:41	0d 2h 47m 35s	1/4	SSH OK - OpenSSH_9.7p1 Debian-5 (protocol 2.0)
	Swap Usage	OK	06-19-2024 06:02:30	3d 18h 9m 52s	1/4	SWAP OK - 73% free (746 MB out of 1023 MB)
	Total Processes	OK	06-19-2024 06:02:30	3d 18h 9m 14s	1/4	PROCS OK: 51 processes with STATE = RSZDT

Results 1 - 9 of 9 Matching Services

Page Tour

CONCLUSION & RECOMMENDATIONS:

Findings:

The "Linux Guardian: Nagios-Powered Host Monitoring" project successfully demonstrated the implementation and configuration of Nagios to monitor Linux hosts. The key findings include:

1. **Effective Monitoring:**
 - Nagios provided comprehensive real-time monitoring of various system metrics, services, and applications on the Linux hosts.
 - The configured services (HTTP, SSH, NCPA Version) were monitored effectively, with alerts generated for any anomalies.
2. **Proactive Issue Detection:**
 - The monitoring setup facilitated early detection of potential issues, allowing for prompt intervention and resolution.
 - System administrators could monitor the health and performance of the infrastructure continuously.
3. **Enhanced Security:**
 - The use of Nagios and NCPA agents helped in maintaining a robust security posture by monitoring critical services and system metrics.
 - SSH was used for secure remote management, ensuring secure access to the monitored hosts.
4. **Scalability and Flexibility:**
 - The setup can be easily scaled to include additional hosts and services.
 - The flexibility of Nagios allows for customization and extension to meet specific monitoring needs.

Countermeasures for Cyber Attacks or Vulnerabilities

1. **Regular Updates and Patching:**
 - Ensure that all components (Nagios, NCPA, OS) are regularly updated to the latest versions to mitigate vulnerabilities.
2. **Secure Configuration:**
 - Configure Nagios and NCPA with secure settings, including strong community strings and access controls.
 - Use firewalls to restrict access to critical ports (HTTP, HTTPS, SSH) only to trusted IP addresses.
3. **Strong Authentication:**
 - Implement strong authentication mechanisms for accessing the Nagios web interface and SSH.
 - Use complex passwords and consider multi-factor authentication (MFA) where possible.
4. **Monitoring and Logging:**
 - Continuously monitor system logs and Nagios alerts for any signs of suspicious activity.
 - Set up alerting mechanisms to notify administrators of critical issues promptly.
5. **Backup and Recovery:**
 - Regularly back up Nagios configuration files and other critical data to ensure quick recovery in case of a system failure or cyber-attack.

6. Network Segmentation:

- Segment the network to isolate critical systems and limit the impact of potential breaches.

CONCLUSION:

The project effectively illustrated how Nagios can offer reliable administration and monitoring for Linux systems. Implementing proactive issue identification, effective alerting, and real-time monitoring greatly improved the Linux-based infrastructure's performance, security, and stability. The project's countermeasures made clear how crucial it is to keep an eye on things constantly and act quickly to preserve a safe and reliable IT environment. In real-world situations, the knowledge and abilities acquired via this project will be crucial for administering and safeguarding Linux systems.

LIST OF REFERENCES:

• Nagios Documentation:

- Nagios Core Documentation (<https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/>)
- Nagios Plugins Documentation (<https://nagios-plugins.org/doc/>)
- Nagios QuickStart Guide([https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/quickstart.h
tml](https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/quickstart.html))

☐ **NCPA (Nagios Cross-Platform Agent):**

- NCPA Download (<https://www.nagios.org/ncpa/>)
- NCPA Documentation (<https://www.nagios.org/ncpa/help/>)

☐ **Kali Linux:**

- [Kali Linux Official Website](https://www.kali.org/) (<https://www.kali.org/>)
- Kali Linux Documentation (<https://www.kali.org/docs/>)

☐ **OpenSSH:**

- OpenSSH Documentation (<https://www.openssh.com/manual.html>)

☐ **Linux Networking:**

- Linux Network Configuration (<https://www.linux.com/training-tutorials/linux-network-configuration/>)

☐ **Firewall Configuration:**

- [UFW \(Uncomplicated Firewall\) Documentation](https://help.ubuntu.com/community/UFW) (<https://help.ubuntu.com/community/UFW>)
- iptables Documentation (<https://linux.die.net/man/8/iptables>)

☐ **General Linux Resources:**

- [The Linux Documentation Project](https://www.tldp.org/) (<https://www.tldp.org/>)
- [Linux Command](https://linuxcommand.org/) (<https://linuxcommand.org/>)

