Лабораторна робота №7

Налаштування Nagios моніторингу Windows на базі NRPE (Nagios Remote Plugin Executor).

Мета: формування практичних навичок налаштування моніторингу Windows-серверів у системі Nagios за допомогою взаємодії NSClient++ з NRPE, а також моніторингу основних служб доменних контролерів.

Інструменти: гіпервізор VirtualBox, модель комп'ютерної мережі.

Теоретичні відомості

На рис.7.1. наведена модель комп'ютерної мережі, побудована під час виконання попередніх лабораторних робіт. До Nagios-серверу Serv-G-N-3 та дослідного серверу Ubuntu Serv-G-N-3 налаштовано SSH доступ через NAT Network для VirtualBox Host.

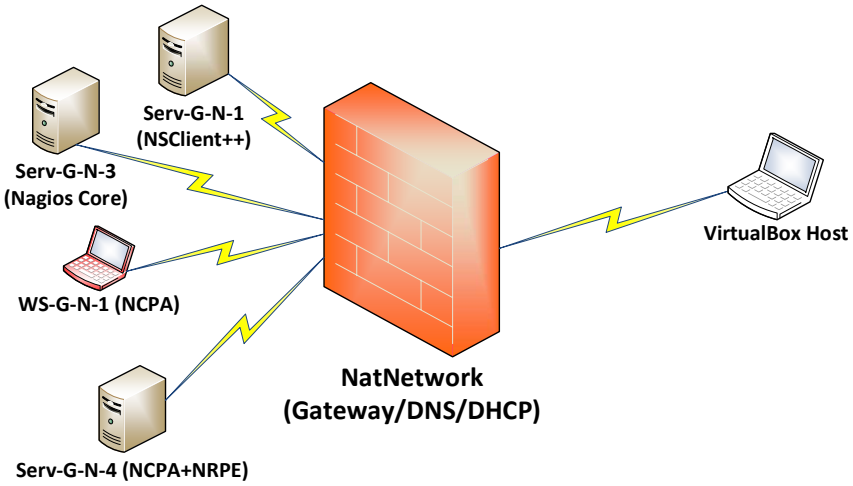

Рис. 7.1. Топологія мережі

На сервері Serv-G-N-3 розгорнуто систему моніторингу на базі Nagios 4.X. Моніторинг основних сервісів серверу Serv-G-N-1 виконується за допомогою NSClient++. Основні сервіси робочої станції WS-G-N-1 та Ubuntu-серверу Serv-G-N-4 відслідковуються за допомогою NCPA та NRPE. Налаштовано підключення з хосту NAT Network по протоколу HTTP до систему моніторингу під користувачем nagios.

NRPE розроблений, щоб дозволити запускати плагіни Nagios на віддалених машинах Linux/Unix, але успішно взаємодіє з NSClient++ на Windows.

<u>Windows сервер. Serv-G-N-1.</u>

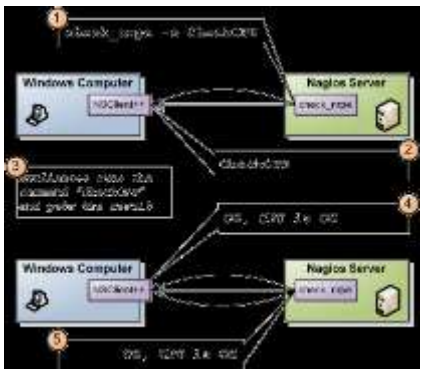

Рис. 7.2. Взаємодія Nagios з NSClient++ на Windows за допомогою check_nrpe.

NRPE працює так само, як SSH або telnet тощо. Він передає команду та очікує на результат. На наведеній вище діаграмі (рис 7.2) відбувається наступне:

1. Nagios виконує check_nrpe з відповідними аргументами.
2. NSClient++ отримує команду для виконання
3. NSClient++ виконує команду та отримує результат у формі за бажанням
4. NSClient++ надсилає результат назад до Nagios
5. Nagios отримує результат із check_nrpe (і використовує його, як і будь-який інший плагін)

Отже, по суті, NRPE — це просто транспортний механізм для надсилання результату команди перевірки через мережу. У якості зовнішніх скриптів NSClient++ можуть бути використані модулі з репозиторіїв Nagios. Завантажимо скрипт перевірки процесів Windows https://assets.nagios.com/downloads/nagiosxi/agents/nrds_win_plugins/check_winprocess.exe та помістимо його у каталозі scripts сценаріїв NSClient++. Команда визначається у файлі nsclient.ini та тестується з командного рядка на сервері Nagios:

```
[/settings/external scripts/scripts]
check_winprocess = scripts\check_winprocess.exe $ARG1$
```

Додаємо секції [/settings/external scripts/scripts], [/settings/external scripts]

```
[/settings/external scripts]
allow arguments = true
allow nasty characters = true
```

та визначаємо у ній команду check_winprocess, та дозволяє передачу аргументів і використання спеціальних символів (наприклад, круглих дужок (), лапок тощо).

Перевіряємо роботу завантаженого скрипта та виконаних налаштувань на сервері Serv-G-N-1 (рис.7.3):

cd "C:\Program Files\NSClient++"
scripts\check_winprocess.exe
scripts\check_winprocess.exe --warn 100 --critical 300

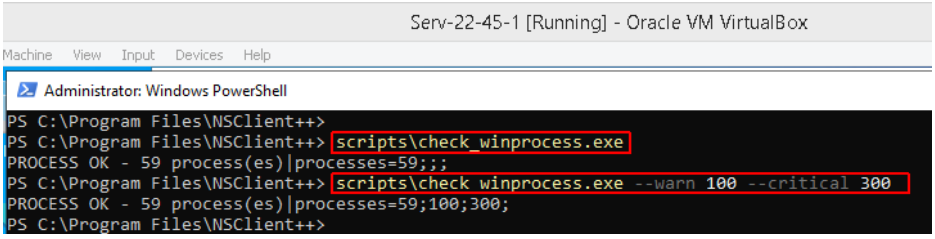

Рис. 7.3. Check_winprocess на сервері Serv-22-45-1

Nagios сервер. Serv-G-N-3.

Налаштуємо описаний процес. Сервер Nagios:

/usr/local/nagios/libexec/check_nrpe -H x.x.x.x -c check_winprocess -a '--warn 100 --critical 300' PROCESS OK - 99 process(es)|'processes'=99;100;300

Ця команда викличе помилку ***-bash: syntax error near unexpected token `('***, що виникає бо не налаштовано ключів шифрування nrpe. На Nagios сервері (Serv_G_N_3), у каталозі check_nrpe, створюємо DH SSL ключ для «спілкування» NSClient++ з NRPE.

cd /usr/local/nagios/libexec

sudo openssl dhparam -out nrpe_dh_2048.pem 2048

Ця операція займає кілька хвилин. Очікуйте її завершення та наберіться терпіння ☺

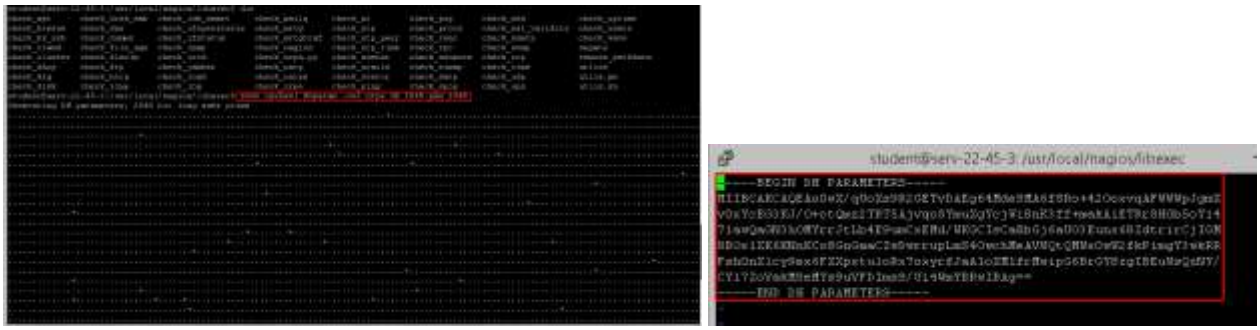

Рис. 7.4. Генерація DH SSL ключа на сервері Serv-22-45-3 (Nagios)

Windows сервер. Serv-G-N-1.

Вміст файлу ключа /usr/local/nagios/libexec/dh_2048.pem зберігаємо на Windows сервері з NSClient++ у файлі C:\Program Files\NSClient++\security\nrpe_dh_2048.pem. Попередній вміст можливо видалити, або не генерувати новий ключ і зберегти вміст цього файлу у /usr/local/nagios/libexec/dh_2048.pem на сервері Nagios.

Основна вимога – вміст файлів має збігатись ☺

Редагуємо файл C:\Program Files\NSClient++\nsclient.ini дозволяючи зовнішні скрипти та додаючи відповідні команди, що описують конфігурацію взаємодії з nrpe

```
[/settings/NRPE/server]
ssl options =
```

```
allow arguments = true
allow nasty characters = true
use ssl = 1
port = 5666
extended response = 1
dh = C:\Program Files\NSClient++\security\nrpe_dh_2048.pem
[/modules]
…
NRPEListener = enabled
CheckServiceState = enabled
```

Перезавантажуємо сервіс “NSClient++ Monitoring Agent”, завершуючи налаштування Serv-G-N-1.

Nagios сервер. Serv-G-N-3.

Перевіряємо взаємодію NRPE на Nagios з NSClient++ на Serv-G-N-1. Ключ «-2» додається для ігнорування сумісності версій клієнта та сервера.

/usr/local/nagios/libexec/check_nrpe -H 192.168.45.135 -2

/usr/local/nagios/libexec/check_nrpe -H 192.168.45.135 -2 -c check_drivesize

/usr/local/nagios/libexec/check_nrpe -H 192.168.45.135 -2 -c CheckCPU -a warn=80 crit=90 time=20m time=10s time=4

/usr/local/nagios/libexec/check_nrpe -H 192.168.45.135 -c check_winprocess -a "--warn 100 --critical 300"

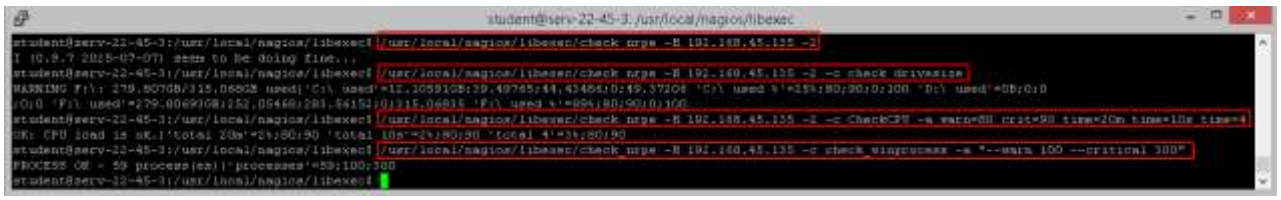

Рис. 7.5. Перевірка відгуку NSClient++ на Serv_22_45_1 на запити NRPE Nagios сервера.

У одній з попередніх лабораторних робіт здійснювався віддалений моніторинг служб на сервері з операційною системою Windows Server 2022, що не має встановлених серверних ролей. Основна мета полягала в перевірці доступності та стану базових служб Windows, таких як:

DHCP Client — служба, що забезпечує автоматичне отримання мережевих параметрів.

Data Sharing Service — служба, яка підтримує спільний доступ до даних між додатками.

Для цього використовувався інструмент NSClient++ у поєднанні з плагіном check_nt, що є частиною моніторингових систем типу Nagios.

Повторимо налаштування моніторингу деяких служб, але з використанням іншого підходу — NRPE (Nagios Remote Plugin Executor). На відміну від check_nt, NRPE надає більшу гнучкість, оскільки:

- підтримує детальніші локальні перевірки, які виконуються безпосередньо на сервері;
- дозволяє створювати власні команди для моніторингу;
- працює на основі конфігураційного файлу NSClient++, де адміністратор самостійно визначає, які саме команди будуть доступні для опитування.

Виконаємо ці перевірки для описаних служб:

/usr/local/nagios/libexec/check_nrpe -H 192.168.45.135 -c check_service -a "service=DsSvc" "ok=state='running'" "critical=state='stopped'"

/usr/local/nagios/libexec/check_nrpe -H 192.168.45.135 -c check_service -a "service=Dhcp" "ok=state='running'" "critical=state='stopped'"

/usr/local/nagios/libexec/check_nrpe -H 192.168.45.135 -c check_service -a "service=EventLog" "ok=state='running'" "critical=state='stopped'"

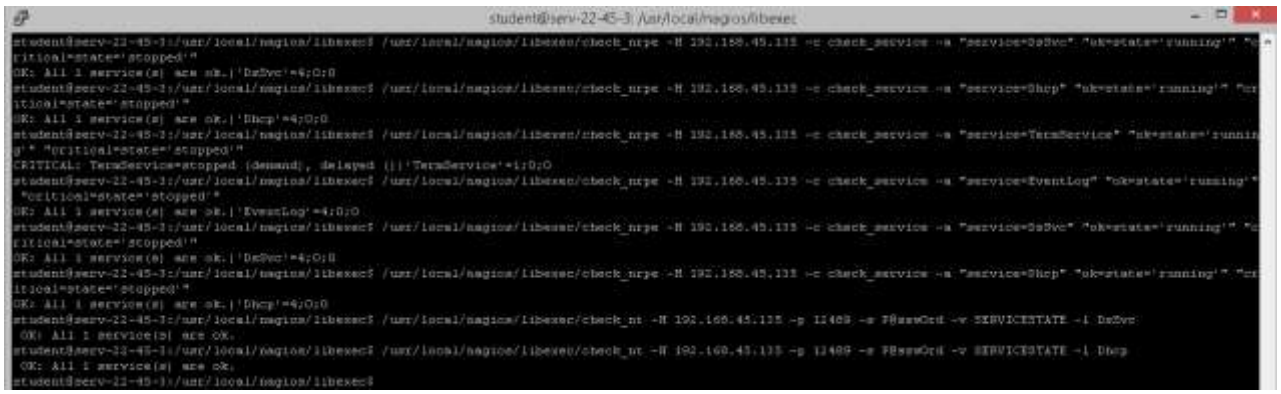

Рис. 7.6. Перевірка служб DC за допомогою NSClient++ за запитами NRPE Nagios сервера.

Додаємо налаштовані команди до конфігураційного файлу сервера

/usr/local/nagios/etc/objects/windows/serv-22-45-1.cfg

```
        define service {
            use                     generic-service
            host_name               serv-22-45-1
            service_description     NRPE Check Drive Size
            check_command           check_nrpe!check_drivesize
        }
        define service {
            use                     generic-service
            host_name               serv-22-45-1
            service_description     NRPE Check CPU
            check_command           check_nrpe!checkcpu -a warn=80 crit=90 time=20m time=10s time=4
        }
        define service {
            use                     generic-service
            host_name               serv-22-45-1
            service_description     NRPE Check DHCP Client
            check_command           check_nrpe!check_service -a "service=Dhcp" "ok=state='running'" "critical=state='stopped'"
        }
        define service {
            use                     generic-service
            host_name               serv-22-45-1
            service_description     NRPE Check Windows Event Log
            check_command           check_nrpe!check_service -a "service=EventLog" "ok=state='running'" "critical=state='stopped'"
        }
        define service {
            use                     generic-service
            host_name               serv-22-45-1
            service_description     NRPE Check WinProcess
            check_command           check_nrpe!check_winprocess!--warn 100 --critical 300
}
```

Перевірка вірності внесених у конфігурацію змін та перезапуск сервісу Nagios:

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

sudo service nagios restart

Переглядаємо роботу виконаних налаштувань:

Рис. 7.7. Перегляд сервісів Serv-22-45-1

Завдання до лабораторної роботи

1. Налаштуйте взаємодію NSClient++ на сервері Serv-G-N-1 з NRPE на сервері Serv-G-N-3.
2. Налаштуйте моніторинг мінімум 5 сервісів серверу Serv-G-N-1 за допомогою NRPE.

Звіт має містити:

- лістинг використаних команд;
- скріншоти отриманих результатів моніторингу у Nagios 4;
- короткий опис редагування файлів конфігурації Nagios 4.

Додаток 7.1.

Перелік базових командних рядків check_nrpe для роботи з NSClient

- У додатку наведено повні командні рядки check_nrpe, що використані у вигляді команд при побудові конфігурації хосту з NSClient++.
- Після -H - IP-хосту, де встановлено NSClient++ 192.168.45.135

/usr/local/nagios/libexec/check_nrpe -H 192.168.45.135 -2

I (0.9.7 2025-07-07) seem to be doing fine...

/usr/local/nagios/libexec/check_nrpe -H 192.168.45.135 -2 -c check_drivesize

OK All 2 drive(s) are ok|'C:\ used'=12.12471GB;39.49765;44.43486;0;49.37206 'C:\ used %'=25%;80;90;0;100 'D:\ used'=0B;0;0;0;0

/usr/local/nagios/libexec/check_nrpe -H 192.168.45.135 -2 -c CheckCPU -a warn=80 crit=90 time=20m time=10s time=4

OK: CPU load is ok.|'total 20m'=2%;80;90 'total 10s'=2%;80;90 'total 4'=4%;80;90

/usr/local/nagios/libexec/check_nrpe -H 192.168.45.135 -c check_winprocess -a "--warn 100 --critical 300"

PROCESS OK - 62 process(es)|'processes'=62;100;300

student@serv-22-45-3:/usr/local/nagios/libexec$ /usr/local/nagios/libexec/check_nrpe -H 192.168.45.135 -c check_service -a "service=DsSvc" "ok=state='running'" "critical=state='stopped'"

OK: All 1 service(s) are ok.|'DsSvc'=4;0;0

student@serv-22-45-3:/usr/local/nagios/libexec$ /usr/local/nagios/libexec/check_nrpe -H 192.168.45.135 -c check_service -a "service=Dhcp" "ok=state='running'" "critical=state='stopped'"

OK: All 1 service(s) are ok.|'Dhcp'=4;0;0

student@serv-22-45-3:/usr/local/nagios/libexec$ /usr/local/nagios/libexec/check_nrpe -H 192.168.45.135 -c check_service -a "service=EventLog" "ok=state='running'" "critical=state='stopped'"

OK: All 1 service(s) are ok.|'EventLog'=4;0;0

Додаткові налаштування C:\Program Files\NSClient++\nsclient.ini

```
[/settings/NRPE/server]
ssl options =
allow arguments = true
allow nasty characters = true
use ssl = 1
port = 5666
extended response = 1
dh = C:\Program Files\NSClient++\security\nrpe_dh_2048.pem

[/modules]
...
NRPEServer = enabled
NRPEListener = enabled
CheckServiceState = enabled
```

Корисні посилання

- Nagios Add-Ons Projects
https://www.nagios.org/downloads/nagios-core-addons/

- NRPE - How To Install NRPE v4 From Source
https://support.nagios.com/kb/article/nrpe-how-to-install-nrpe-v4-from-source-515.html

- NRPE - How to install NRPE
https://support.nagios.com/kb/article/nrpe-how-to-install-nrpe-8.html

- Index of /downloads/nagiosxi/agents
https://assets.nagios.com/downloads/nagiosxi/agents/

- Exchange Nagios. NRPE - Nagios Remote Plugin Executor
https://exchange.nagios.org/directory/Addons/Monitoring-Agents/NRPE--2D-Nagios-Remote-Plugin-Executor/details

- Using NSClient++ with check_nrpe
https://nsclient.org/docs/howto/nrpe/

- The Nagios Plugins. Category: Operating Systems