Лабораторна робота №8

Додаткові налаштування системи моніторингу Nagios: користувачі, часові проміжки, контакти.

Мета: формування практичних навичок розширеного налаштування системи моніторингу Nagios, зокрема управління часовими періодами моніторингу, створення користувачів із обмеженими правами доступу та налаштування системи оповіщення про критичні події.

Інструменти: гіпервізор VirtualBox, модель комп'ютерної мережі.

Теоретичні відомості

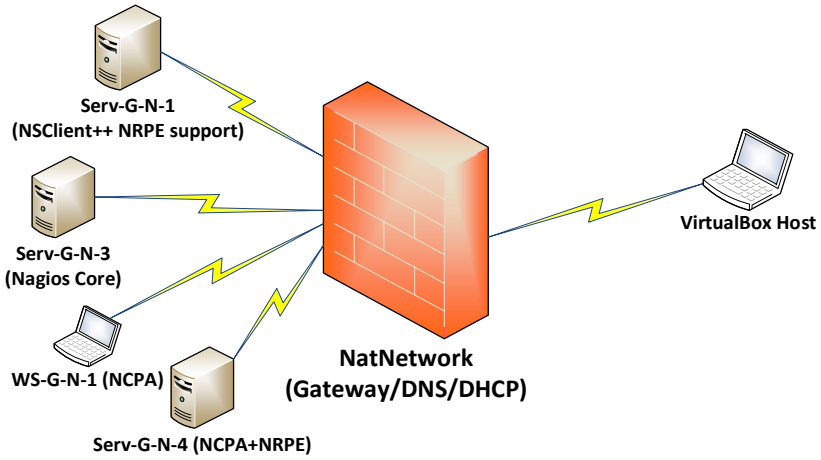

Рис. 8.1. Топологія мережі

На рис. 8.1 наведена модель комп'ютерної мережі, побудована під час виконання попередніх лабораторних робіт. До серверу Serv-G-N-2 налаштовано SSH доступ через NAT Network для VirtualBox Host.

На сервері Serv-G-N-2 розгорнуто систему моніторингу на базі Nagios 4.X. Налаштовано підключення з хосту NAT Network по протоколу HTTP до систему моніторингу під користувачем nagios.

Time Periods in Nagios Core.

Налаштування часових інтервалів моніторингу дозволяє контролювати, коли можуть працювати різні аспекти логіки моніторингу та оповіщення. Наприклад, можна обмежити виконання регулярних запланованих перевірок хосту та служби, надсилати сповіщення, використовувати ескалацію сповіщень, коли діють залежності.

Визначення хостів і служб мають необов'язкову директиву check_period , яка дозволяє вказати період часу, який слід використовувати для обмеження, коли можна виконувати регулярні активні перевірки хосту чи служби. Якщо директиву check_period не використовується для визначення періоду часу, Nagios Core зможе запланувати активні перевірки хосту чи служби в будь-який час. По суті, це сценарій моніторингу 24x7.

Зазначення періоду часу в директиві check_period дозволяє обмежити час, протягом якого Nagios Core виконує регулярні заплановані активні перевірки хосту або служби. Коли Nagios Core намагається перепланувати перевірку хосту або служби, він переконається, що наступна перевірка потрапляє в дійсний діапазон часу в межах визначеного періоду часу. Якщо цього не відбувається, Nagios Core налаштує час наступної перевірки так, щоб він збігся з наступним «дійсним» часом у вказаний період часу. Це означає, що хост або служба можуть не перевірятися знову через годину, день чи тиждень тощо.

Приклади директив для різних періодів часу можна знайти тут . Ми будуємо тестово-навчальну систему моніторингу, тому налаштуємо два часових проміжки моніторингу. Ці налаштування виконуються у файлі ***/usr/local/nagios/etc/objects/timeperiods.cfg***. Відкриваємо файл і одразу бачимо кілька вже визначених часових періодів та шаблонів до них. Наприклад стандарт моніторингу 24x7 виглядає наступним чином:

```
define timeperiod {
    name                    24x7
    timeperiod_name         24x7
    alias                   24 Hours A Day, 7 Days A Week
    sunday                  00:00-24:00
    monday                  00:00-24:00
    tuesday                 00:00-24:00
    wednesday               00:00-24:00
    thursday                00:00-24:00
    friday                  00:00-24:00
    saturday                00:00-24:00
}
```

Створимо свій перший часовий період, що має працювати у режимі 24x7 з вимкненням моніторингу кожну неділю з 22:00 до 23:00 по Київському часу. Скажімо, що у цей час має виконуватися сервісне перезавантаження серверів. З огляду на приклади конфігурація такого часового періоду буде мати вигляд:

```
# 24/7 with downtime every Sunday from 22:00 to 23:00 EET
define timeperiod {
  timeperiod_name   24x7-no-sunday-night
  alias             24/7 (except Sunday night)
  sunday            00:00-23:00
  monday            00:00-24:00
  tuesday           00:00-24:00
  wednesday         00:00-24:00
  thursday          00:00-24:00
  friday            00:00-24:00
  saturday          00:00-24:00
}
```

Другий часовий період, протягом якого виконується моніторинг робочих станцій та їх периферії. Період має тривати лише робочий час персоналу - з понеділка по четвер – з 9:00 до 18:00, у п'ятницю – з 9:00 до 17:00 по Київському часу. Конфігурація такого часового періоду має наступний вигляд:

```
# Work hours (Mon-Fri 9:00-18:00, Fri 9:00-17:00)
define timeperiod {
  timeperiod_name   work-hours
  alias             Work hours
  monday            09:00-18:00
  tuesday           09:00-18:00
  wednesday         09:00-18:00
  thursday          09:00-18:00
  friday            09:00-17:00
  saturday          00:00-00:00
  sunday            00:00-00:00
}
```

Додаємо часові періоди до конфігураційного файлу ***/usr/local/nagios/etc/objects/timeperiods.cfg*** та виконуємо перевірку вірності внесених у конфігурацію змін з перезапуском сервісу Nagios:

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

sudo service nagios restart

Під час виконання попередньої лабораторної ми виконали для лінукс сервера Serv-G-N-4 налаштування перевірки доступності доменних імен ztu.edu.ua та тимчасово вимкнули їх моніторинг щоб запобігти можливому ефекту DDoS-атак через часті звернення.

```
define service {
    host_name                serv-22-40-3
    use                      generic-service
    service_description      Check domain ztu.edu.ua
    check_command            check_nrpe!check_dns -a ztu.edu.ua
    max_check_attempts       5
    check_interval           30
    retry_interval           1
    check_period             work-hours
    notification_interval    60
    notification_period      work-hours
}
```

Редагуємо секції сервісів перевірки доменів у файлі конфігурації хосту ***/usr/local/nagios/etc/objects/linux/serv-22-45-4.cfg*** Знімаємо коментарі з рядків, параметр check_interval збільшуємо з 5 хв до 30, у якості періоду перевірок check_period замість стандартного значення ***24x7*** задаємо назву визначеного робочого періоду ***work-hours***.

Так само змінюємо notification_period, що визначає періоди, коли Nagios буде відправляти повідомлення про недоступність сервісу. notification_period може відрізнятися від check_period.

Для зменшення кількості рядків і уникнення повторення одних і тих самих параметрів у файлі конфігурації ***/usr/local/nagios/etc/objects/linux/serv-22-45-4.cfg***, створюємо шаблон сервісу. У цьому шаблоні винесено всі типові параметри, що спільні для більшості сервісів. Секція шаблону має вигляд:

```
define service {
    name                     24x7-no-sunday-night-service
    register                 0
    active_checks_enabled    1
    max_check_attempts       5
    check_interval           5
    retry_interval           1
    check_period             24x7-no-sunday-night
    notification_interval    60
    notification_period      24x7-no-sunday-night
}
```

Як бачите, для значень ***check_period*** та ***notification_period*** у шаблоні вказано налаштований нами часовий період ***24x7-no-sunday-night.***

За потреби цей шаблон, як і будь який користувацький, можна винести до окремого файлу шаблонів сервісів. Однак, враховуючи невелику кількість хостів у навчальному стенді, цю секцію залишено безпосередньо у конфігураційному файлі відповідного хосту.

Для всіх сервісів даного хосту, за винятком перевірки доступності доменів, задається використання створеного шаблону 24x7-no-sunday-night-service. Це дозволяє уникнути дублювання параметрів у кожному сервісі. Відповідно, із секцій define service видаляються параметри, які вже визначені у шаблоні:

```
active_checks_enabled    1
max_check_attempts       5
check_interval           5
retry_interval           1
check_period             24x7-no-sunday-night
notification_interval    60
notification_period      24x7-no-sunday-night
```

Таким чином, типовий сервіс (не доступність домену) у файлі конфігурації хосту serv-22-45-4 отримує вигляд:

Зверніть увагу на лінк сервісу заданого як новий шаблон: ***use 24x7-no-sunday-night-service***

```
define service {
    host_name                serv-22-45-4
    use                      24x7-no-sunday-night-service
    service_description      Process Count
    check_command            check_ncpa!-t 'P@ssw0rd' -P
5693 -M processes -w 150 -c 200
}
```

Таким чином, всі сервіси даного серверу, крім перевірки доменів, працюють по шаблону ***24x7-no-sunday-night-service***.

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:

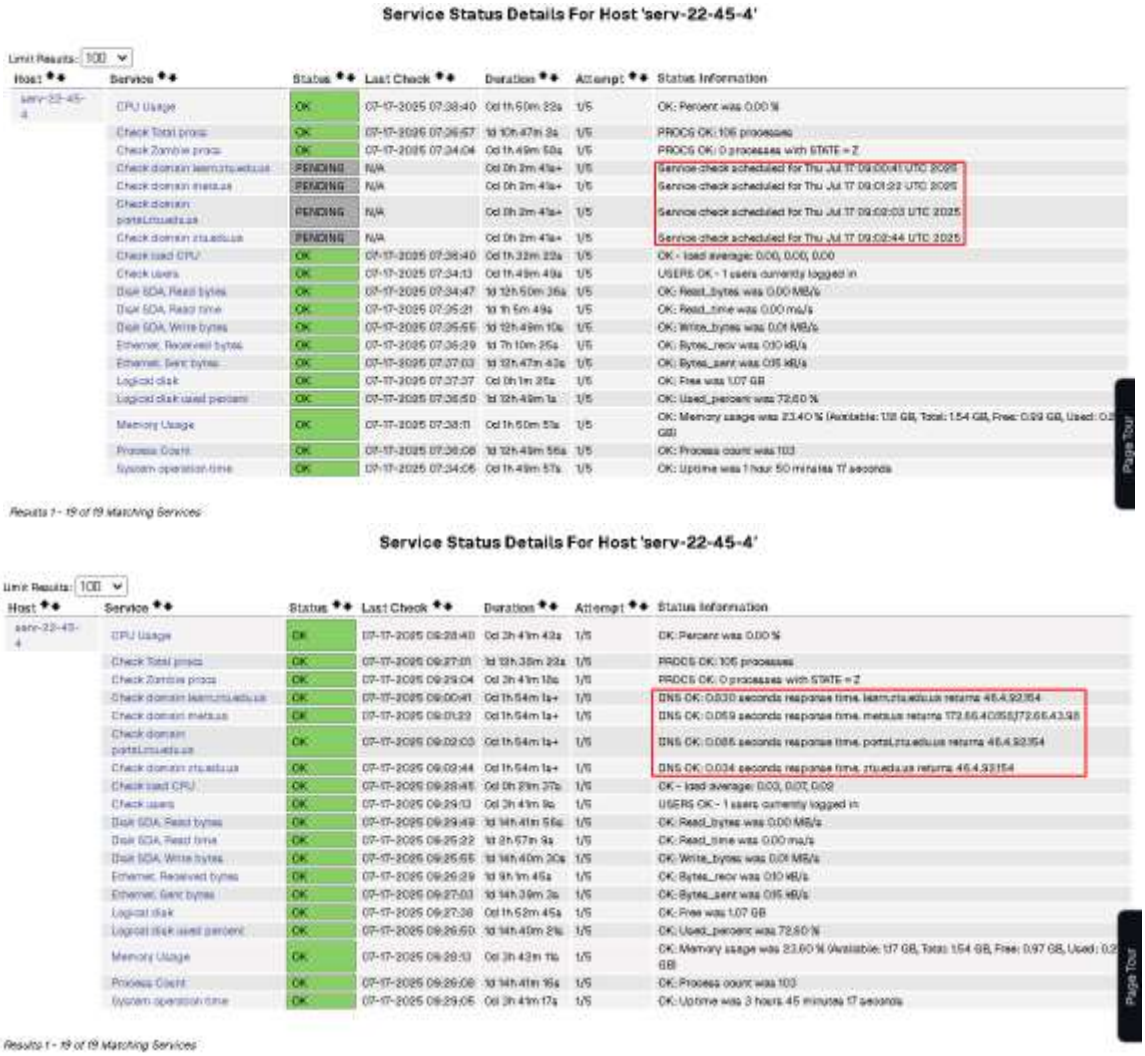

Рис. 8.2. Перегляд сервісів Serv-22-45-4

На рис. 8.2 показане очікування настання заданої 30-ти хвилинної затримки для перевірки доступності доменів. Зразок файлу конфігурації хосту Serv-22-45-4 з усіма внесеними змінами наведено у додатку 8.1.

Змінюємо часові налаштування моніторингу для робочої станції – редагуємо її конфігураційний файл ***/usr/local/nagios/etc/objects/workstation/ws-22-45-1.cfg***, змінюючи значення check_period та notification_period з 24x7 на work-hours, а параметр check_interval збільшуємо з 5 хв до 15.

Створюємо шаблон сервісу, подібно до попередніх кроків у файлі конфігурації робочої станції ***/usr/local/nagios/etc/objects/workstation/ws-22-45-1.cfg***. створюємо шаблон сервісу. У цьому шаблоні винесено всі типові параметри, що спільні для більшості сервісів.
Секція шаблону має вигляд:

```
define service {
    name                    worstation-service
    register                0
    active_checks_enabled   1
    max_check_attempts      5
    check_interval          15
    retry_interval          1
    check_period            work-hours
    notification_interval   60
    notification_period     work-hours
}
```

Перемикаємо всі сервіси робочої станції на використання створеного шаблону worstation-service та видаляємо зайві рядки з конфігурацій сервісів. Секція типового сервісу отримає насутпний вигляд:

```
define service {
    host_name               WS-22-45-1
    use                     worstation-service
    service_description     Process Count
    check_command           check_ncpa!-t 'P@ssw0rd' -P
5693 -M processes -w 150 -c 200
}
```

Вміст відредагованого файлу конфігурації ***/usr/local/nagios/etc/objects/workstation/ws-22-45-1.cfg*** наведено у додатку 8.2.

Виконуємо перевірку вірності внесених у конфігурацію змін з перезапуском сервісу Nagios:

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

sudo service nagios restart

Очікуємо ***check_interval*** і дивимось на картинку рис. 8.3.

Рис. 8.3. Перегляд сервісів WS-22-45-1

<u>Користувачі Nagios Core.</u>

У конфігураційному файлі ***/usr/local/nagios/etc/cgi.cfg*** є кілька глобальних змінних для налаштування прав користувачів:

Таблиця 9.1

authorized_for_system_information	SYSTEM/PROCESS INFORMATION ACCESS	Користувачі, які мають доступ до перегляду інформації про процес Nagios.
authorized_for_configuration_information	CONFIGURATION INFORMATION ACCESS	Користувачі, які можуть переглядати ВСЮ конфігураційну інформацію (хости, команди тощо). За замовчуванням користувачі можуть переглядати лише конфігураційну інформацію для хостів і служб, для яких вони є контактами.
authorized_for_system_commands	SYSTEM/PROCESS COMMAND ACCESS	Користувачі, які можуть видавати команди завершення роботи та перезапуску Nagios за допомогою команди CGI (cmd.cgi) і можуть змінювати режим програми на активний або очікування.
authorized_for_all_services authorized_for_all_hosts	GLOBAL HOST/SERVICE VIEW ACCESS	Користувачі, які можуть переглядати інформацію для всіх хостів і служб, які контролюються. За замовчуванням користувачі можуть переглядати лише інформацію про хости чи служби, для яких вони є контактами.
authorized_for_all_service_commands authorized_for_all_host_commands	GLOBAL HOST/SERVICE COMMAND ACCESS	Користувачі, які можуть видавати команди, пов'язані з хостом або службою, за допомогою команди CGI (cmd.cgi) для всіх хостів і служб, які контролюються. За замовчуванням користувачі можуть видавати команди лише для хостів або служб, для яких вони є контактами.
authorized_for_read_only	READ-ONLY USERS	Користувачі, які мають права лише на читання в CGI. Для них блоковані будь-які служби, коментарі чи команди хосту, які зазвичай відображаються на сторінках extinfo CGI.

Стандартна задача. Необхідно надати обмежений доступ виконавцю, який буде лише переглядати статуси хостів та сервісів і отримувати на пошту повідомлення про певні хости. Пам'ятаючи про «чутливість» Linux до регістру символів ☺, створюємо користувача системи моніторингу surname:

sudo htpasswd /usr/local/nagios/etc/htpasswd.users falkovsky

Додаємо користувача surname до файлу контактів Nagios

/usr/local/nagios/etc/objects/contacts.cfg

```
define contact {
    contact_name             falkovsky
    alias                    Nagios User
    email                    kkik_fiv@ztu.edu.ua
    service_notification_period 24x7
    host_notification_period    24x7
    service_notification_options w,u,c,r
    host_notification_options    d,r
    service_notification_commands notify-service-by-email
    host_notification_commands   notify-host-by-email
}
```

У конфігураційному файлі ***/usr/local/nagios/etc/cgi.cfg*** налаштовуємо наступні глобальні змінні

```
authorized_for_system_information=nagios,falkovsky
authorized_for_configuration_information=nagios,falkovsky
authorized_for_read_only=falkovsky
authorized_for_all_services=nagios,falkovsky
authorized_for_all_services=nagios,falkovsky
authorized_for_all_hosts= nagios,falkovsky
```

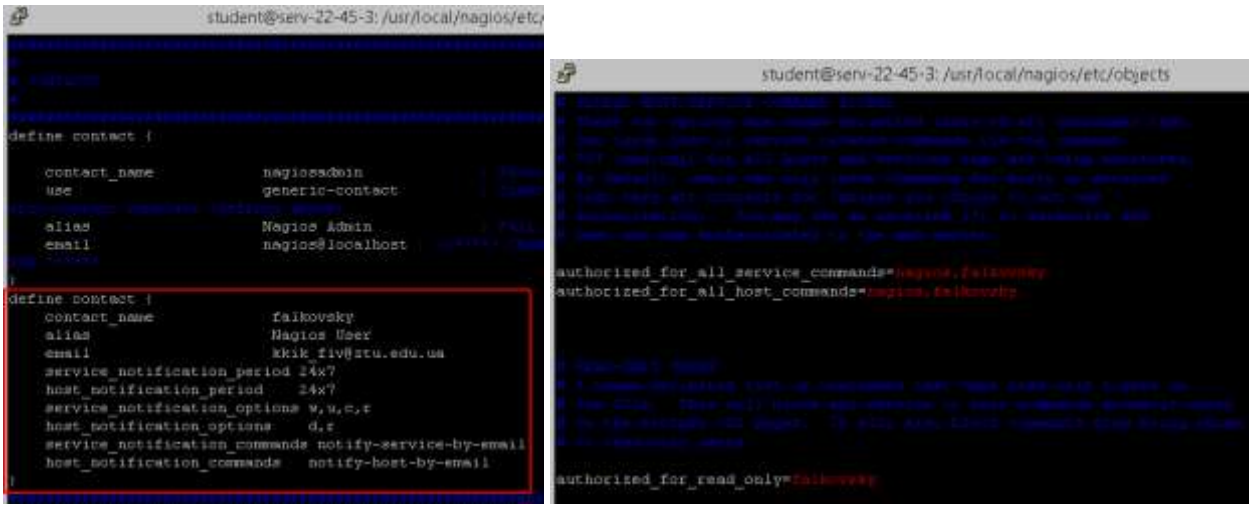

Рис. 8.4. Налаштування прав користувача surname

Виконуємо перевірку вірності внесених у конфігурацію змін. Перезапускаємо сервіси Nagios та Apache:

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

sudo service nagios restart

sudo systemctl restart apache2

Перезапуск сервісу Apache виконується через зміни у файлі ***/usr/local/nagios/etc/cgi.cfg***

<u>Налаштування відправки поштових повідомлень.</u>

У файлах конфігурації Nagios є налаштування відсилання повідомлень у випадку проблем з хостами або сервісами. Виконаємо додаткові налаштування для повноцінної роботи цього функціоналу.

Вище описано як у файлі ***/usr/local/nagios/etc/objects/contacts.cfg*** визначаються контакти за замовчуванням (рис. 8.3) та інші. Відповідно для кожного хосту або навіть сервісу можливо визначити свій контакт.

Визначення контакту для локального хосту у файлі ***/usr/local/nagios/etc/objects/linux/localhost.cfg*** виконується додаванням рядку contacts у відповідну секцію

```
define host {
    hostgroups              linux-servers
    use                     linux-server
    host_name               serv-22-45-3
    alias                   Serv-22-45-3
    address                 127.0.0.1
    contacts                falkovsky
}
```

Визначення контакту surname виконано у файлі ***/usr/local/nagios/etc/objects/contacts.cfg*** де в полі email прописується адреса одержувача, наприклад, surname@ukr.net або surname@ztu.edu.ua Також, в описі контакту є поля, в яких описано, коли відправляти повідомлення про стан хосту чи сервісу (рис.8.4).

```
define contact {
    contact_name                    falkovsky
    alias                           Nagios User
    email                           kkik_fiv@ztu.edu.ua
    service_notification_period 24x7
    host_notification_period    24x7
    service_notification_options w,u,c,r
    host_notification_options    d,r
    service_notification_commands notify-service-by-email
    host_notification_commands   notify-host-by-email
}
```

Пояснення параметрів:

contact_name: Це ім'я контакту, яке ідентифікує його в системі.
alias: Це псевдонім контакту, який може бути використаний для відображення в інтерфейсі Nagios.
email: Адреса електронної пошти, на яку будуть надсилатися повідомлення.
service_notification_period: Період, коли служби будуть сповіщати про стан.
host_notification_period: Період, коли будуть сповіщати про стан хосту.
service_notification_commands: Команда для сповіщень про стан служби.
host_notification_commands: Команда для сповіщень про стан хосту.

Таблиця 8.2

host_notification_options	service_notification_options
значення за замовчуванням «d, r»	значення за замовчуванням «w,u,c,r»
Можливі варіанти значень:	Можливі варіанти значень:
• d — повідомляти про статус DOWN • u — повідомляти про статус UNREACHABLE • r — повідомляти про підняття хоста (перший UP)	• w — повідомляти про статус WARNING service states, • u — повідомляти про статус UNKNOWN service states, • c — повідомляти про статус CRITICAL service states,

• f — повідомляти про початок і закінчення «блимаючого» стану • s — повідомляти про початок і закінчення запланованого вимкнення • n — не відсилати жодних повідомлень	• r — повідомляти про підняття сервісу (перша поява статусу OK) • f — повідомляти про початок і закінчення «блимаючого» стану • n — не відсилати жодних повідомлень

Входимо до Nagios під користувачем surname

Для підключення до WEB-інтерфейсу Nagios можливо використовувати анонімні вкладки у браузерах: Google Chrome (комбінація клавіш: Ctrl + Shift + N), Internet Explorer (IE 11) (комбінація клавіш: Ctrl + Shift + P).

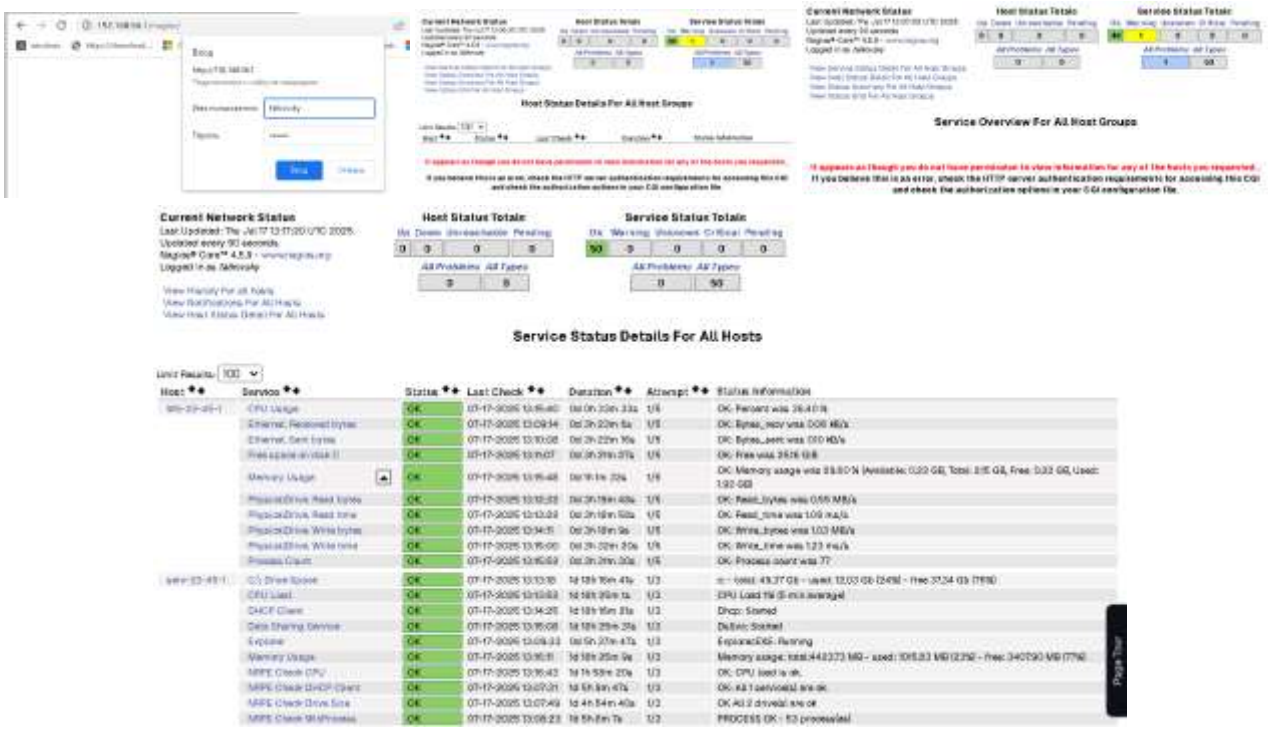

Рис. 8.5. Перегляд WEB GUI Nagios під користувачем falkovsky.

Завдання до лабораторної роботи

1. Налаштуйте новий часовий період системи моніторингу таким чином щоб моніторинг відбувався не більш ніж 5 днів на тиждень та 7 годин на день, з періодичністю не частіше 1 раз на 3 години. Змініть налаштування моніторингу на цей період для лінукс-сервера Serv-G-N-3. Зніміть коменти з сервісів моніторингу зовнішніх доменів, що були налаштовані у одній з попередніх робіт.
2. Створіть додаткового, гостьового користувача у системі моніторингу таким чином, щоб він не мав доступу до системних команд, команд управління сервісами та команд управління пристроями. Ім'я користувача – довільне.
3. Увійдіть до веб-інтерфейсу системи моніторингу Nagios під створеним користувачем surname, де surname — ваше прізвище. Перевірте, які хости та сервіси відображаються для цього користувача, відповідно до його ролі та налаштувань доступу.
   До звіту додайте скріншоти GUI-інтерфейсу Nagios, доступного користувачу surname, зокрема:
   - головна сторінка (Home – Tactical Overview);
   - список хостів (Hosts);
   - список сервісів (Services);
   - приклад інформації про конкретний хост або сервіс.

Звіт має містити:

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

Додаток 8.1.

Зразок конфігурації Serv-22-45-4

```
define host {
    host_name               serv-22-45-4
    address                 192.168.45.132
    use                     linux-server
    hostgroups              linux-servers
    check_command           check_ncpa!-t 'P@ssw0rd' -P 5693 -M system/agent_version
    max_check_attempts      5
    check_interval          5
    retry_interval          1
    check_period            24x7-no-sunday-night
    notification_interval   60
    notification_period     24x7-no-sunday-night
    notifications_enabled   1
}
define service {
    name                    24x7-no-sunday-night-service
    register                0
    active_checks_enabled   1
    max_check_attempts      5
    check_interval          5
    retry_interval          1
    check_period            24x7-no-sunday-night
    notification_interval   60
    notification_period     24x7-no-sunday-night
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     CPU Usage
    check_command         check_ncpa!-t 'P@ssw0rd' -P 5693 -M cpu/percent -w 20 -c 40 -q 'aggregate=avg'
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     Memory Usage
    check_command           check_ncpa!-t 'P@ssw0rd' -P 5693 -M memory/virtual -w 50 -c 80 -u G
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     Process Count
    check_command           check_ncpa!-t 'P@ssw0rd' -P 5693 -M processes -w 150 -c 200
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     Ethernet. Sent bytes
    check_command           check_ncpa!-t 'P@ssw0rd' -P 5693 -M interface/enp0s3/bytes_sent -d -u k -w 10 -c 100
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     Ethernet. Received bytes
    check_command          check_ncpa!-t 'P@ssw0rd' -P 5693 -M interface/enp0s3/bytes_recv -d -u k -w 10 -c 100
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     System operation time
    check_command           check_ncpa!-t 'P@ssw0rd' -P 5693 -M 'system/uptime'
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     Logical disk
    check_command        check_ncpa!-t 'P@ssw0rd' -P 5693 -M 'disk/logical/|/free' --warning 0.8: --critical 0.3:
-u G
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     Logical disk used percent
    check_command             check_ncpa!-t 'P@ssw0rd' -P 5693 -M 'disk/logical/|/used_percent' --warning 90 --
critical 95
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     Disk SDA. Read bytes
    check_command            check_ncpa!-t 'P@ssw0rd' -P 5693 -M 'disk/physical/sda/read_bytes' -d -u M -w 50 -c
100
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     Disk SDA. Write bytes
```

```
    check_command           check_ncpa!-t 'P@ssw0rd' -P 5693 -M 'disk/physical/sda/write_bytes' -d -u M -w 50 -c
100
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     Disk SDA. Read time
    check_command           check_ncpa!-t 'P@ssw0rd' -P 5693 -M 'disk/physical/sda/read_time' -d -w 50 -c 100
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     Check users
    check_command           check_nrpe!check_users
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     Check load CPU
    check_command           check_nrpe!check_load
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     Check Total procs
    check_command           check_nrpe!check_total_procs
}
define service {
    host_name               serv-22-45-4
    use                     24x7-no-sunday-night-service
    service_description     Check Zombie procs
    check_command           check_nrpe!check_zombie_procs
}
define service {
    host_name               serv-22-45-4
    use                     generic-service
    service_description     Check domain meta.ua
    check_command           check_nrpe!check_dns -a meta.ua
    max_check_attempts      5
    check_interval          30
    retry_interval          1
    check_period            work-hours
    notification_interval   60
    notification_period     work-hours
}
define service {
    host_name               serv-22-45-4
    use                     generic-service
    service_description     Check domain ztu.edu.ua
    check_command           check_nrpe!check_dns -a ztu.edu.ua
    max_check_attempts      5
    check_interval          30
    retry_interval          1
    check_period            work-hours
    notification_interval   60
    notification_period     work-hours
}
define service {
    host_name               serv-22-45-4
    use                     generic-service
    service_description     Check domain learn.ztu.edu.ua
    check_command           check_nrpe!check_dns -a learn.ztu.edu.ua
    max_check_attempts      5
    check_interval          30
    retry_interval          1
    check_period            work-hours
    notification_interval   60
    notification_period     work-hours
}
define service {
    host_name               serv-22-45-4
    use                     generic-service
    service_description     Check domain portal.ztu.edu.ua
    check_command           check_nrpe!check_dns -a portal.ztu.edu.ua
    max_check_attempts      5
    check_interval          30
    retry_interval          1
    check_period            work-hours
    notification_interval   60
    notification_period     work-hours
}
```

Додаток 8.2.

Зразок конфігурації WS-22-45-1

```
define host {
    host_name                WS-22-45-1
    address                  192.168.45.133
    hostgroups               win-workstations
    check_command            check_ncpa!-t 'P@ssw0rd' -P 5693 -M system/agent_version
    contact_groups           admins
    max_check_attempts       5
    check_interval           15
    retry_interval           1
    check_period             work-hours
    notification_interval    60
    notification_period      work-hours
    notifications_enabled    1
}
define service {
    name                     worstation-service
    register                 0
    active_checks_enabled    1
    max_check_attempts       5
    check_interval           15
    retry_interval           1
    check_period             work-hours
    notification_interval    60
    notification_period      work-hours
}

define service {
    host_name                WS-22-45-1
    use                      worstation-service
    service_description      CPU Usage
    check_command            check_ncpa!-t 'P@ssw0rd' -P 5693 -M cpu/percent -w 50 -c 750 -q 'aggregate=avg'
}
define service {
    host_name                WS-22-45-1
    use                      worstation-service
    service_description      Memory Usage
    check_command            check_ncpa!-t 'P@ssw0rd' -P 5693 -M memory/virtual -w 80 -c 90 -u G
}
define service {
    host_name                WS-22-45-1
    use                      worstation-service
    service_description      Process Count
    check_command            check_ncpa!-t 'P@ssw0rd' -P 5693 -M processes -w 150 -c 200
}
define service {
    host_name                WS-22-45-1
    use                      worstation-service
    service_description      Free space on disk C
    check_command            check_ncpa!-t 'P@ssw0rd' -P 5693 -M 'disk/logical/C:|/free' -w 15: -c 10: -u Gi
}
define service {
    host_name                WS-22-45-1
    use                      worstation-service
    service_description      PhysicalDrive. Read bytes
    check_command             check_ncpa!-t 'P@ssw0rd' -P 5693 -M 'disk/physical/PhysicalDrive0/read_bytes' -d -u M
-w 50 -c 100
}
define service {
    host_name                WS-22-45-1
    use                      worstation-service
    service_description      PhysicalDrive. Write bytes
    check_command             check_ncpa!-t 'P@ssw0rd' -P 5693 -M 'disk/physical/PhysicalDrive0/write_bytes' -d -u
M -w 50 -c 100
}
define service {
    host_name                WS-22-45-1
    use                      worstation-service
    service_description      PhysicalDrive. Read time
    check_command             check_ncpa!-t 'P@ssw0rd' -P 5693 -M 'disk/physical/PhysicalDrive0/read_time' -d -w 50
-c 100
}
define service {
    host_name                WS-22-45-1
    use                      worstation-service
    service_description      PhysicalDrive. Write time
    check_command              check_ncpa!-t 'P@ssw0rd' -P 5693 -M 'disk/physical/PhysicalDrive0/write_time' -d -w
50 -c 100
}
define service {
    host_name                WS-22-45-1
    use                      worstation-service
    service_description      Ethernet. Sent bytes
    check_command             check_ncpa!-t 'P@ssw0rd' -P 5693 -M 'interface/Ethernet/bytes_sent' -d -u k -w 10 -c
100
}
define service {
```

```
    host_name               WS-22-45-1
    use                     worstation-service
    service_description     Ethernet. Received bytes
    check_command            check_ncpa!-t 'P@ssw0rd' -P 5693 -M 'interface/Ethernet/bytes_recv' -d -u k -w 10 -c
100
}
```

Корисні посилання

- Nagios Core. Time Periods.

https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/timeperiods.html

- Nagios Core. Time Period Definition

https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/objectdefinitions.html#timeperiod

- Nagios Core. CGI Configuration File Options

https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/configcgi.html

- How to Install and Use SendEmail on Linux

https://tecadmin.net/how-to-install-sendemail-in-linux/

- NagiosQL - Nagios configuration tool Files

https://sourceforge.net/projects/nagiosql/files/nagiosql/