Лабораторна робота №3

Моніторинг віддаленого Windows хосту за допомогою Icinga 2 Agent.

Мета: Набути практичних навичок налаштування агентського моніторингу у системі Icinga 2 для контролю стану серверних ресурсів у середовищі Windows Server 2022. Ознайомитися з принципами встановлення та конфігурації Icinga 2 Agent, забезпечити його інтеграцію з Icinga 2 Master, налаштувати перевірку ключових показників роботи локального хосту (CPU, RAM, дисковий простір, служби). Забезпечити відображення результатів моніторингу у веб-інтерфейсі Icinga Web 2 та перевірити коректність обміну даними між агентом і сервером моніторингу.

Інструменти: гіпервізор VirtualBox, модель комп'ютерної мережі.

Теоретичні відомості

У попередніх лабораторних роботах було створено віртуалізоване стендове середовище у NAT Network VirtualBox, що складається з трьох хостів:

Serv-G-N-1 (Windows Server 2022) – на якому розгорнуто контролер домену з ролями AD DS, DNS і DHCP;

Serv-G-N-3 (Ubuntu Server 24.04) – на якому встановлено та налаштовано систему моніторингу Icinga 2 разом із веб-інтерфейсом Icinga Web 2 і компонентами Icinga DB;

WS-G-N-1 (Windows 10) – робоча станція, включена до внутрішнього домену.

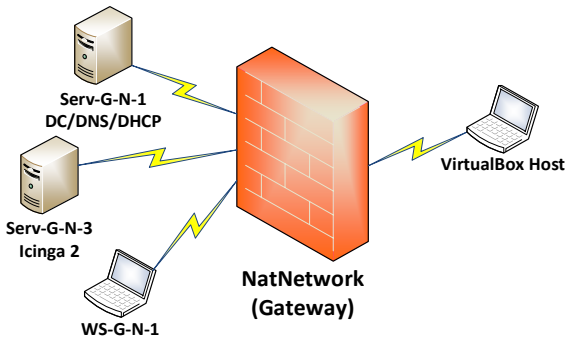

Рис. 3.1. Топологія мережі

Мережеве середовище забезпечує взаємодію між вузлами, а система Icinga 2 інтегрована з доменною інфраструктурою для подальшого моніторингу її елементів.

Встановлення Icinga 2 Agent на Windows Server 2022

Переходимо на офіційний сайт Icinga https://icinga.com/download/ та обираємо шлях до ICINGA Package Repository https://packages.icinga.com/windows/. Завантажуємо актуальний інсталяційний пакет Icinga 2 Agent for Windows (MSI). На момент написання цього документу це файл Icinga2-v2.15.0-x86_64.msi (агент версії 2.15.0). Запускаємо завантажений .msi інсталятор від імені адміністратора та виконуємо дії, показані на рис. 3.1.

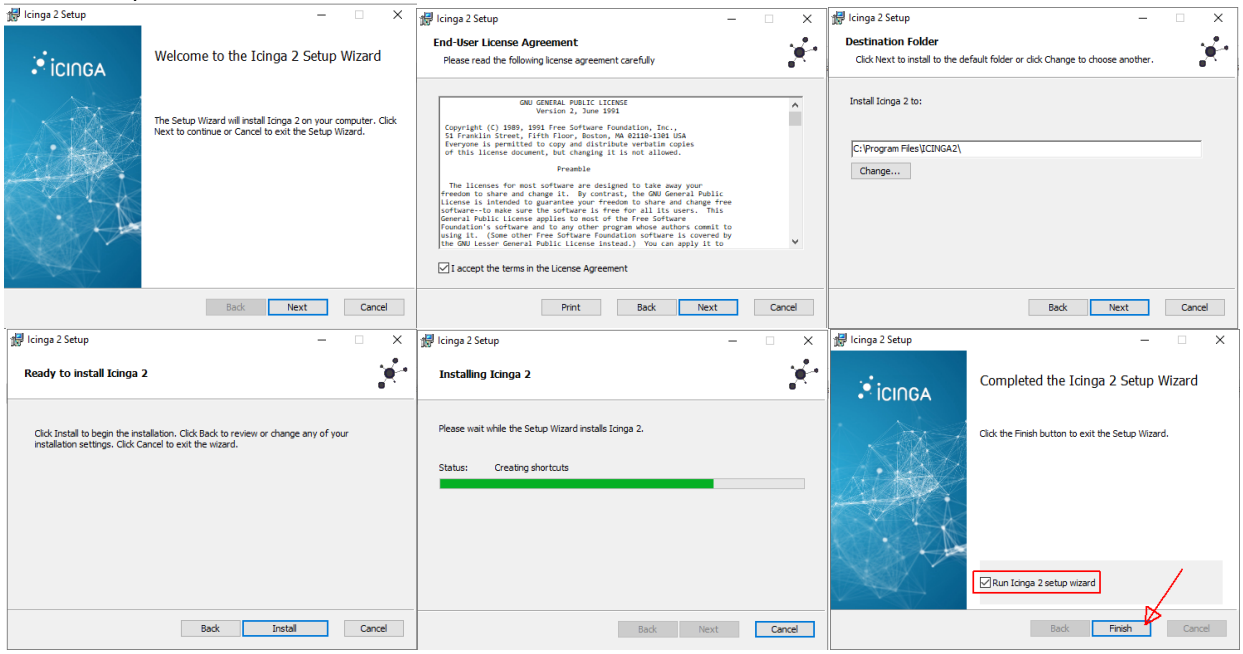

Рис. 3.2. Майстер інсталяції Icinga 2 Agent

На мал.6 рис.3.2 обов'язково відмічаємо Run Icinga 2 setup wizard, що ініціює виклик наступного майстра (рис. 3.3). На першому вікні заповнюємо Instance name (FQDN) – тут автоматично підставляється

повне доменне ім'я сервера, наприклад serv-G-N-1.surname.net. Це значення залишаємо без змін, оскільки воно використовується для генерації сертифіката та ідентифікації агента у середовищі Icinga.

Далі у полі CSR signing ticket (optional) необхідно вставити одноразовий квиток (ticket), згенерований на сервері Master (Serv-G-N-3).

Для генерації signing ticket відкриваємо SSH-сесію до серверу Ubuntu, де встановлено сервер Icinga, редагуємо файл ***/etc/icinga2/constants.conf***, підставляючи значення рядку з константою TicketSalt

const TicketSalt = "MySecretSalt123"

та виконуємо команди перезавантаження служби icinga2 та генерації сертифікату:

sudo systemctl restart icinga2

sudo icinga2 pki ticket --cn serv-G-N-1.surname.net

У результаті буде виданий унікальний рядок (напр. d5cbc48edfd4e725b8162e170be6f28af2e92831) (рис. 3.4), який копіюємо у поле CSR signing ticket. Це дозволить агенту автоматично отримати сертифікати від Master.

У полі Parent master/satellite instance(s) for this agent по кнопці Add вказуємо (мал.2 рис. 3.4) FQDN ім'я серверу Icinga 2, у якості Host – його IP-адресу, порт залишаємо без змін. Щоб переглянути FQDN Ubuntu сервера використайте команду

hostname -f

Поле Global Zone залишаємо порожнім, оскільки на цьому етапі спільні зони не використовуються.

У секції TCP Listener обираємо варіант Listen for connection from master/satellite instance(s) та залишаємо порт за замовчуванням 5665. Це відкриє канал для двосторонньої взаємодії з Master.

У блоці Advanced options обов'язково відмічаємо Accept command from master/satellite instance(s) – дозволяє виконання перевірок і команд з боку Master та Accept config update from master/satellite instance(s) – дозволяє отримувати оновлення конфігурації. Параметр Run Icinga 2 service as this user залишаємо за замовчуванням.

Опцію Disable include conf.d dir не активуємо, щоб зберегти приклади локальних перевірок (CPU, RAM, Disk), які входять до базової конфігурації.

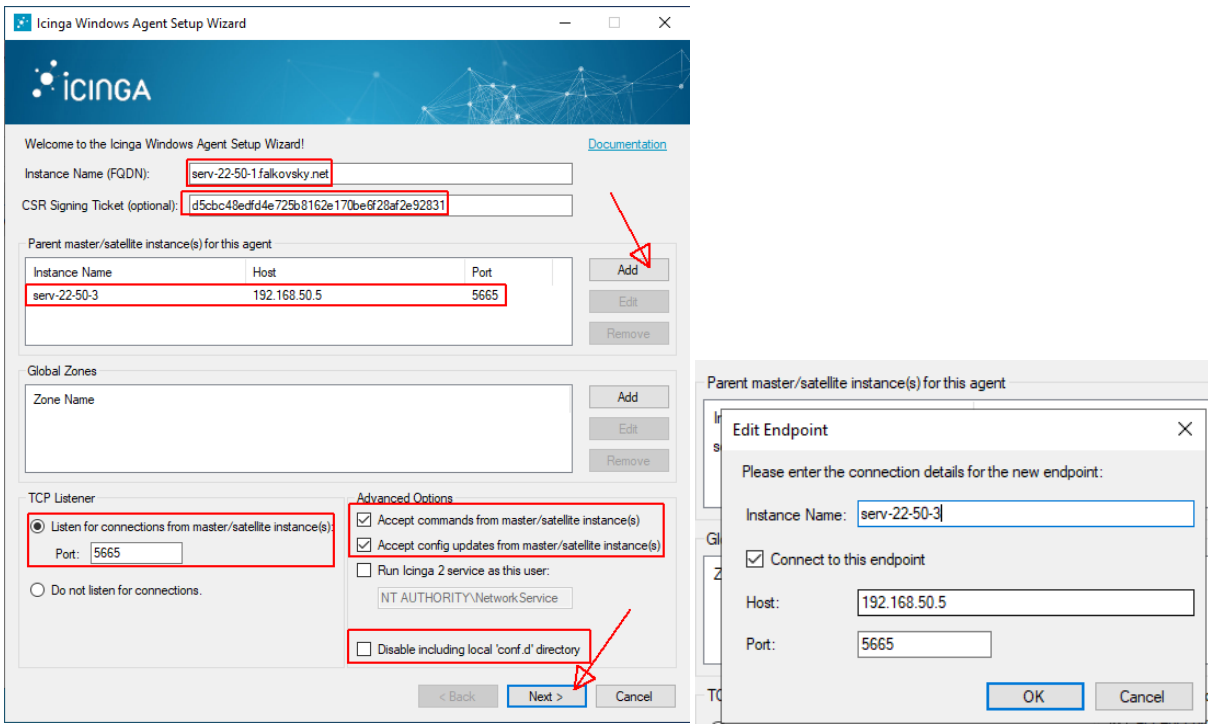

Рис. 3.3. Icinga Windows Agent Setup Wizard

Після заповнення всіх полів натискаємо Next для продовження роботи майстра.

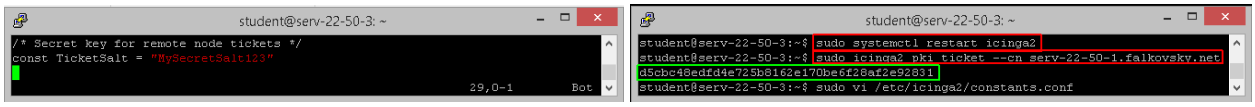

Рис. 3.4. Генерація signing ticket у SSH-сесії сервера Serv-22-50-3

Роботу майстра майже завершено. Буде виконано перевірку дійсності signing ticket, автоматичне створення файлів конфігурацій та перевірка зв'язку між серверами. На рис. 3.5 показані інформаційні вікна майстра, де необхідно обирати Next та Finish.

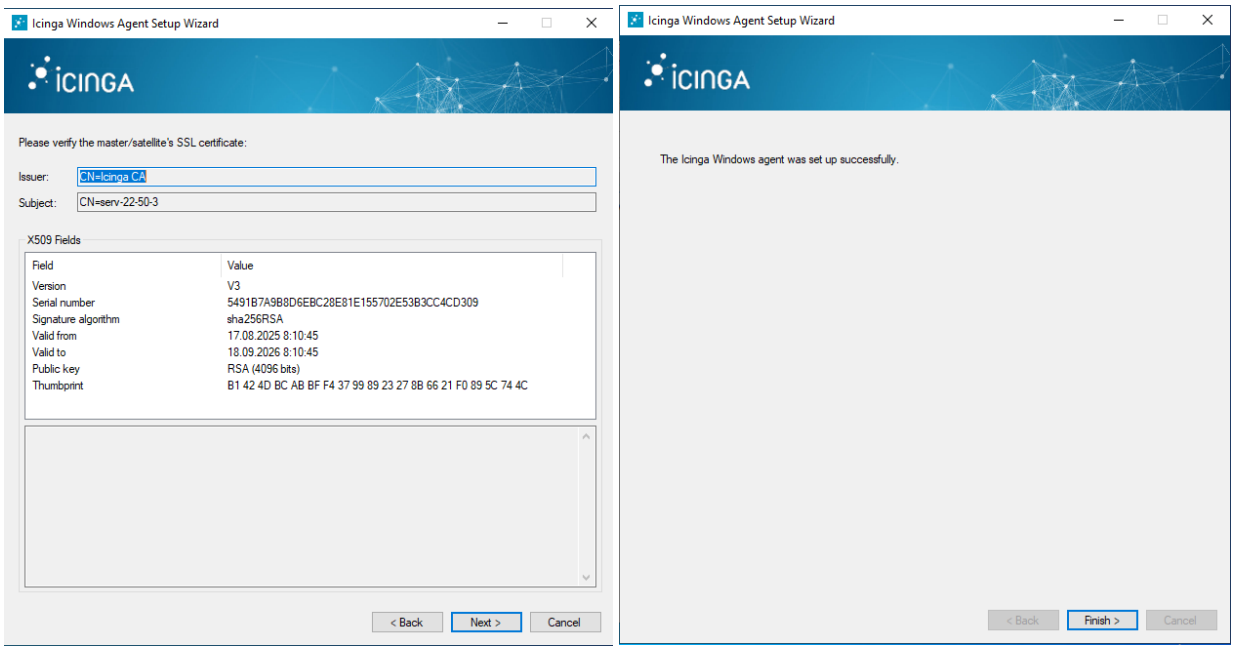

Рис. 3.5. Icinga Windows Agent Setup Wizard. Завершення.

Після завершення роботи майстра агент Serv-G-N-1 успішно підключений до сервера Master (Serv-G-N-3). На цьому етапі автоматично створюються необхідні файли конфігурацій, встановлюється зв'язок між агентом і Master, перевіряється дійсність signing ticket та активуються базові локальні перевірки. Це означає, що агент готовий надсилати дані про стан системи та приймати команди від Master. Наступним кроком є налаштування конкретних перевірок стану CPU, RAM, дискового простору та служб на Windows Server 2022, що дозволить здійснювати повноцінний моніторинг локального хосту у веб-інтерфейсі Icinga Web 2.

Налаштування перевірок стану локального хосту через Icinga 2 Agent

Перевіряємо, що Icinga 2 Agent на Serv-G-N-1 запущений. У службах стан service icinga2 має бути Running з автоматичним запуском.

Наступна перевірка – TCP-порт 5665 має бути відкритим. MSI інсталятор Icinga 2 зазвичай створює правило, але у нашому випадку воно не створилось. Можливо цей недолік буде виправлено у наступних версіях інсталятора.

Відкриваємо [Windows Defender Firewall with Advanced Security] – [Inbound Rules] Натисніть [New Rule…] та створюємо нове правило по шаблону [Port] – [TCP] – [Specific local port: 5665] – [Allow the connection] – [Domain, Private] – [Name: Icinga2] – [Finish].

На рис. 3.6 показані перевірка стану служби icinga2 та перегляд створеного «вручну» правила. На рис.3.7 показана перевірка «спілкування» між серверами по порту 5665.

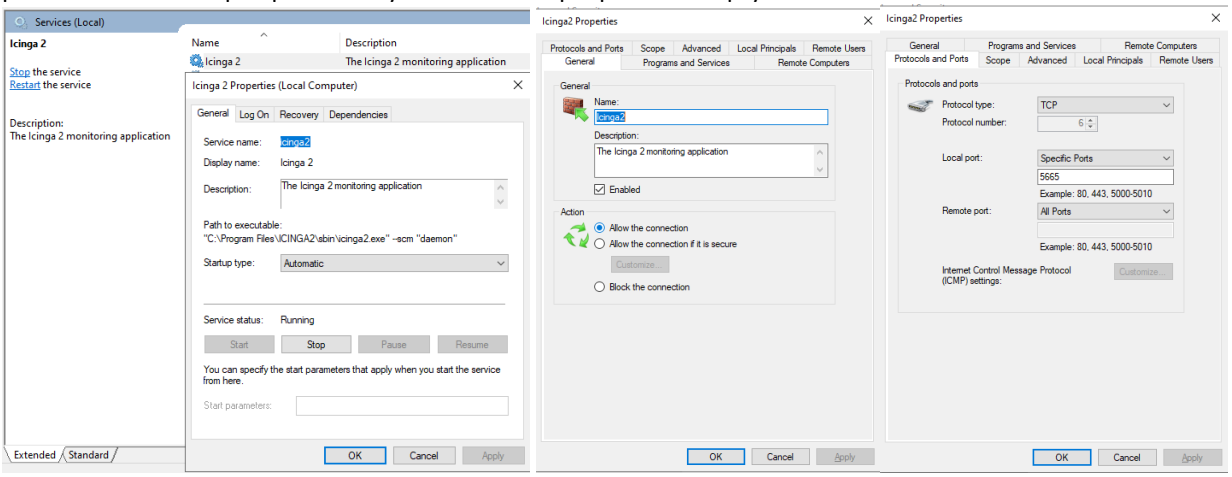

Рис. 3.6. Icinga2 service та правило Windows Defender Firewall.

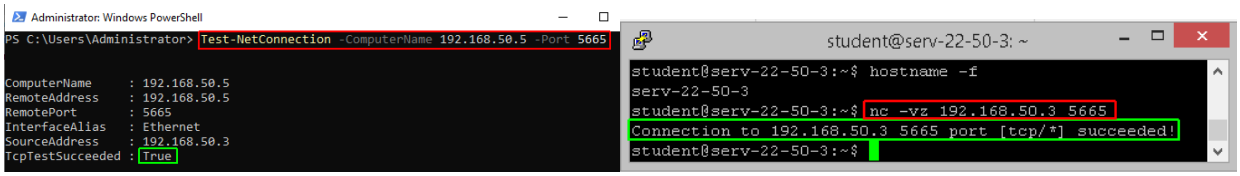

Рис. 3.7. Перевірка відкритості порту 5665.

Відкритість портів можливо перевірити наступними командами для Windows та Ubuntu

Test-NetConnection -ComputerName 192.168.50.5 -Port 5665

nc -vz 192.168.50.3 5665

Наступний крок – створення об'єкта Host у Master. На Ubuntu-сервері Serv-G-N-3 (Master) відкриваємо SSH-сесію. У цій роботі редагуватимуться конфігураційні файли Icinga 2. Аналогічно до Nagios, де для перевірки коректності розгорнутої конфігурації використовується команда:

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

В Icinga 2 існують власні інструменти перевірки. Перевірка коректності всієї конфігурації:

sudo icinga2 daemon --validate

або

sudo icinga2 daemon -C

Ця команда аналізує синтаксис та цілісність конфігурації, виявляє помилки у файлах та залежностях. Рекомендовано виконувати таку перевірку після кожного редагування конфігураційних файлів.

Також необхідно згадати про перевірку кешів конфігурації. Icinga 2 під час роботи створює кешовану конфігурацію, щоб швидше стартувати і не парсити заново усі .conf-файли при кожному запуску. Кеш конфігурації зберігається у каталозі ***/var/cache/icinga2/***. Для контролю її стану використовуються команди:

sudo icinga2 object list

(виводить актуальні об'єкти з кешу, що завантажені в пам'ять), та

sudo icinga2 daemon -C --dump-objects

(окрім перевірки синтаксису, ця команда додатково показує, які саме об'єкти і залежності будуть створені у кеші та зберігає їх). Таким чином, поєднання перевірки синтаксису та аналізу кешу дозволяє своєчасно виявити як синтаксичні помилки, так і логічні проблеми у структурі конфігурацій Icinga.

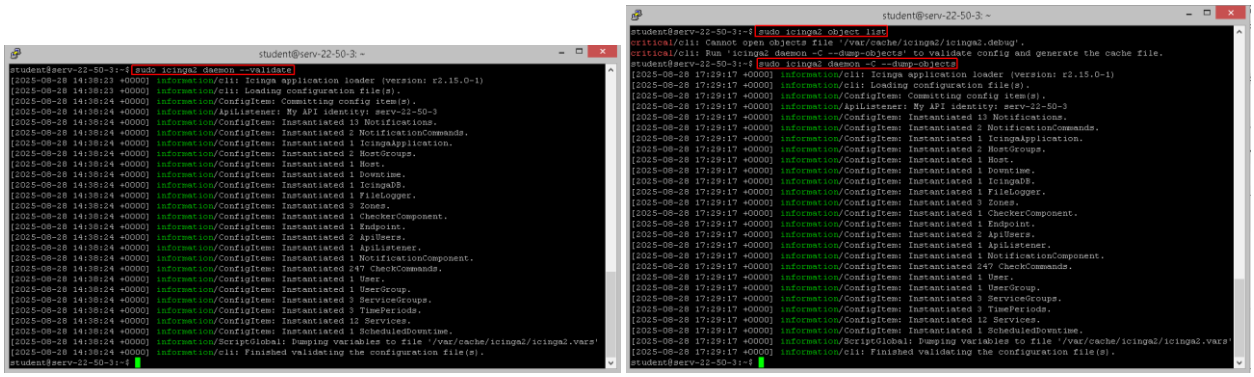

Рис. 3.8. Аналіз синтаксису та цілісності конфігурації та збереження перевіреної конфігурації у кеші.

В директорії конфігурацій Icinga 2 (звичайно /etc/icinga2/conf.d) створюємо файл для хосту, наприклад ***/etc/icinga2/conf.d/serv-22-50-1.conf*** та додаємо у нього базовий опис хоста:

```
object Host "serv-22-50-1.falkovsky.net" {
  import "generic-host"
  address = "192.168.50.3"  # IP Windows-сервера
  vars.os = "Windows"
}
```

Зберігаємо файл та виконуємо перевірку коректності конфігурації (аналог nagios -v ☺):

sudo icinga2 daemon -C

Якщо перевірка пройшла, «Finished validating the configuration file(s)», перезапускаємо Icinga 2

sudo systemctl restart icinga2

У веб-інтерфейсі Icinga Web 2 – Overview – Hosts має з'явитися новий хост serv-G-N-1.surname.net. Всі ці дії показані на рис.3.9.

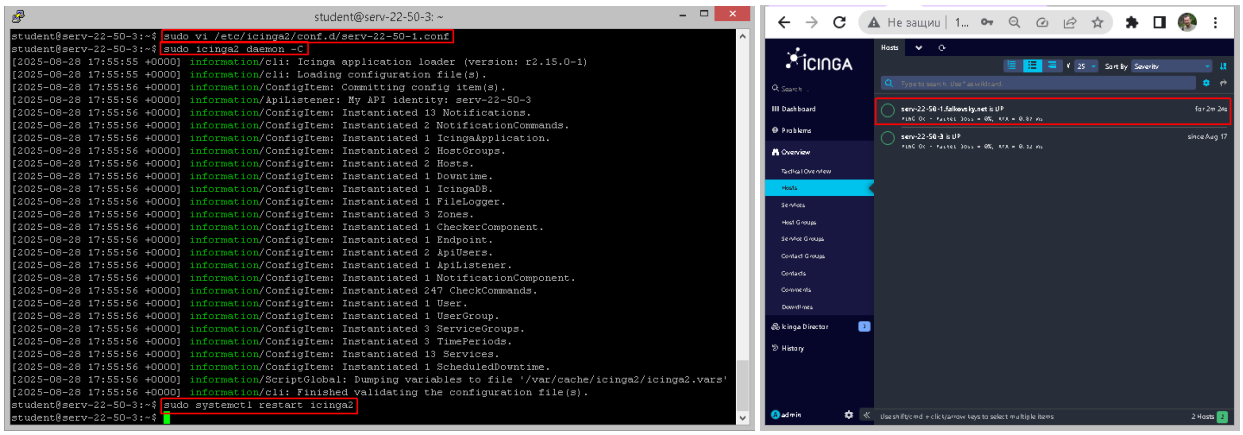

Рис. 3.9. Створення конфігурації хосту serv-22-50-1.falkovsky.net та його перегляд у WEB GUI.

Активуємо базові перевірки серверу. У тій же папці створюємо файл конфігурації сервісів ***/etc/icinga2/conf.d/serv-22-50-1-services.conf*** та додаємо у нього сервіси для перевірки CPU, RAM і диску:

```
apply Service "CPU Load" {
 import "generic-service"
 check_command = "perfmon-windows"
 vars.perfmon_counter = "\\Processor(_Total)\\% Processor Time"
 assign where host.name == "serv-22-50-1.falkovsky.net"
}
apply Service "Memory Usage" {
 import "generic-service"
 check_command = "perfmon-windows"
 vars.perfmon_counter = "\\Memory\\% Committed Bytes In Use"
 assign where host.name == "serv-22-50-1.falkovsky.net"
}
apply Service "C: Disk Usage" {
 import "generic-service"
 check_command = "perfmon-windows"
 vars.perfmon_counter = "\\LogicalDisk(C:)\\% Free Space"
 assign where host.name == "serv-22-50-1.falkovsky.net"
}
apply Service "DFS Namespace Service" {
 import "generic-service"
 check_command = "windows-service "
 vars.service_name = "Dfs"
 assign where host.name == "serv-22-50-1.falkovsky.net"
}
```

У цій конфігурації наведено приклад перевірки служби Windows ”DFS Namespace”. Додаємо ще кілька критичних для серверу служб:

```
apply Service "Active Directory Domain Services" {
 import "generic-service"
 check_command = "service-windows"
 vars.service_name = "NTDS"
 assign where host.name == "serv-22-50-1.falkovsky.net"
}
apply Service "DNS Server" {
 import "generic-service"
 check_command = "service-windows"
 vars.service_name = "DNS"
 assign where host.name == "serv-22-50-1.falkovsky.net"
}
apply Service "DHCP Server" {
 import "generic-service"
 check_command = "service-windows"
 vars.service_name = "DHCPServer"
 assign where host.name == "serv-22-50-1.falkovsky.net"
}
apply Service "Windows Time" {
```

```
  import "generic-service"
  check_command = "service-windows"
  vars.service_name = "W32Time"
  assign where host.name == "serv-22-50-1.falkovsky.net"
 }
 apply Service "Windows Remote Management" {
  import "generic-service"
  check_command = "service-windows"
  vars.service_name = "WinRM"
  assign where host.name == "serv-22-50-1.falkovsky.net"
 }
```

Після редагування конфігурації виконуємо класичну перевірку конфігурації, переконуємось, що повідомлень про помилки немає та перезапускаємо Icinga 2

```
sudo icinga2 daemon -C
sudo systemctl restart icinga2
```

Перемикаємося у веб-інтерфейс Icinga Web 2 та перевіряємо вкладку Hosts та Services для serv-G-N-1.surname.net. Мають відобразитись актуальні значення CPU, RAM, диску та служб.

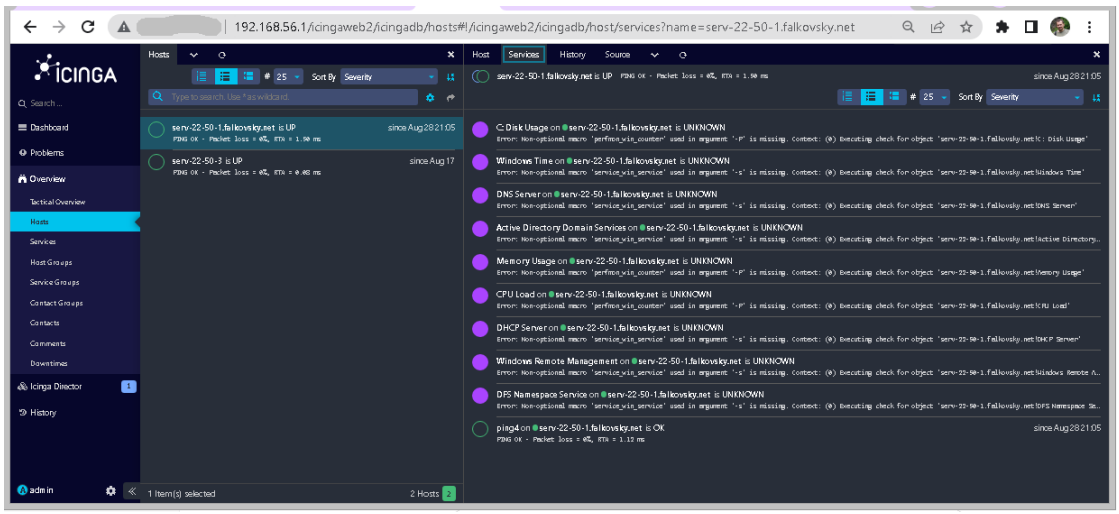

Рис. 3.10. Перегляд сервісів хосту serv-22-50-1.falkovsky.net у WEB GUI.

Завдання до лабораторної роботи

1. **Встановіть Icinga 2 Agent на сервері Windows Server 2022 (Serv-G-N-1)** і підключіть його до Master-сервера (Serv-G-N-3).
   Перевірте коректність встановлення та авторизації агента через CSR signing ticket.
2. **Налаштуйте базові хости у Icinga 2** для моніторингу сервера контролера домену (AD DS).
   Створіть файл конфігурації хоста з FQDN та IP-адресою.
3. **Додайте сервіси для перевірки ключових параметрів Windows-сервера**: CPU, RAM, диску та критичних служб (AD DS, DNS, DHCP).
   Використовуйте відповідні CheckCommand з пакету windows-plugins.
4. **Перевірте конфігурацію та зв'язок між Master і агентом**, виконавши команду icinga2 daemon -C.
   Виправте помилки конфігурації за потреби.
5. **Контролюйте результати моніторингу через Icinga Web 2**, переконайтесь, що дані з агента відображаються у Dashboard.

Звіт має містити:

- Опис встановлення та налаштувань Icinga 2 Agent на Windows Server 2022 і підключення до Master;
- Команди генерації CSR signing ticket та підключення агента;
- Конфігурації хоста та сервісів (CPU, RAM, диск, AD DS, DNS, DHCP);
- Перевірку зв'язку між Master та агентом та стан сервісів (icinga2 daemon -C, Dashboard);
- Скріншоти веб-інтерфейсу Icinga Web 2 із даними агента;
- Короткий опис виконаних етапів та усунення помилок (якщо вони були ☺).

Додаток 1.

Методи визначення сервісів у Icinga 2

В Nagios ми робили купу define service { ... host_name X } або hostgroup_name, і це було досить жорстко. В Icinga 2 через apply rules і vars ми отримуємо дуже гнучку систему.

✓ Один і той самий сервіс для всіх хостів. Приклад - ping для всіх:

```
apply Service "ping4" {
 import "generic-service"
 check_command = "ping4"
 assign where host.address
}
```

Тепер будь-який хост із полем address автоматично отримає цей сервіс.

✓ Сервіси тільки для певних груп/OS. Наприклад, CPU-тести тільки для Windows:

```
apply Service "windows-cpu" {
 import "generic-service"
 check_command = "nscp-local-cpu"
 assign where host.vars.os == "Windows"
}
```

Або диски тільки для Linux:

```
apply Service "linux-disk" {
 import "generic-service"
 check_command = "disk"
 assign where host.vars.os == "Linux"
}
```

✓ Унікальні сервіси для окремих хостів (наприклад, додаткові диски). Використовуємо змінні (vars) прямо в описі хоста:

```
object Host "serv-22-50-1.falkovsky.net" {
 import "generic-host"
 address = "192.168.50.3"
 vars.os = "Windows"
 // оголошуємо, які диски моніторити
 vars.disks = [ "C:", "D:" ]
}
```

А в сервісах пишемо універсальне правило:

```
apply Service "disk" for (disk => config in host.vars.disks) {
 import "generic-service"
 check_command = "disk"
 vars.disk_partitions = [ disk ]
}
```

Результат - якщо у хоста є тільки "C:", то створиться лише перевірка диску C, якщо "C:" і "D:" — будуть два сервіси, якщо змінної vars.disks немає — сервіс взагалі не створиться.

✓ "Hardware" відмінності. Тут теж “рулить” vars. Наприклад, можна в хост додати:

```
vars.is_db_server = true
А правило:
apply Service "mssql" {
 check_command = "mssql"
 assign where host.vars.is_db_server
}
```

Отже, робиться одна універсальна конфігурація, а поведінка підлаштовується під хости завдяки vars.

Додаток 2.

Приклад структурованої конфігурації icinga 2

В Icinga2 дуже важливо правильно структурувати файли конфігурації, щоб потім не заплутатися. У цьому додатку приведено приклад логічного поділу (аналогічний Nagios — хости, сервіси, групи).

Структура конфігурацій у каталозі /etc/icinga2/conf.d

```
/etc/icinga2/conf.d/
├── templates.conf          # шаблони хостів і сервісів
├── hosts/
│   ├── dc1.conf            # опис Domain Controller
│   ├── dns1.conf           # опис DNS сервера
│   └── dhcp1.conf          # опис DHCP сервера
├── services/
│   ├── base_services.conf  # універсальні сервіси (ping, cpu, memory)
│   └── role_services.conf  # сервіси за ролями (AD, DNS, DHCP)
└── groups.conf             # групи хостів і сервісів (опціонально)
```

✓ templates.conf (шаблони хостів і сервісів)

```
template Host "windows-host" {
 import "generic-host"
 vars.os = "Windows"
}
```

```
template Service "generic-windows-service" {
 import "generic-service"
}
```

✓ sts/dc1.conf (опис контролера домену)

```
object Host "dc1.surname.net" {
 import "windows-host"
 address = "192.168.50.10"
 vars.is_domain_controller = true
 vars.is_dns_server = true
 vars.is_dhcp_server = false
}
```

✓ hosts/dhcp1.conf (опис DHCP сервера)

```
object Host "dhcp1.surname.net" {
 import "windows-host"
 address = "192.168.50.11"
 vars.is_domain_controller = false
 vars.is_dns_server = false
 vars.is_dhcp_server = true
}
```

✓ services/base_services.conf (універсальні сервіси для всіх Windows-хостів)

```
apply Service "ping4" {
 import "generic-service"
 check_command = "ping4"
 assign where host.address
}
```

```
apply Service "cpu" {
 import "generic-service"
 check_command = "nscp-local-cpu"
 assign where host.vars.os == "Windows"
}
```

```
apply Service "memory" {
 import "generic-service"
 check_command = "nscp-local-memory"
 assign where host.vars.os == "Windows"
}
```

✓ services/role_services.conf (сервіси за ролями)

```
apply Service "Active Directory Replication" {
 import "generic-service"
 check_command = "nscp-local-service"
 vars.nscp_service = "NTDS"
 assign where host.vars.is_domain_controller
}
DNS
apply Service "DNS Service" {
 import "generic-service"
 check_command = "nscp-local-service"
 vars.nscp_service = "DNS"
 assign where host.vars.is_dns_server
}
DHCP
apply Service "DHCP Service" {
 import "generic-service"
 check_command = "nscp-local-service"
 vars.nscp_service = "DHCPServer"
 assign where host.vars.is_dhcp_server
}
```

✓ (опціонально) groups.conf (групи хостів і сервісів)

```
object HostGroup "domain-controllers" {
 display_name = "Active Directory Controllers"
 assign where host.vars.is_domain_controller
}
object HostGroup "dns-servers" {
 display_name = "DNS Servers"
 assign where host.vars.is_dns_server
}
object HostGroup "dhcp-servers" {
 display_name = "DHCP Servers"
 assign where host.vars.is_dhcp_server
}
```

Така структура дозволяє чітко розділити хости, шаблони і сервіси, додавати новий сервер простим створенням файлу hosts/ім'я.conf та включати сервіси автоматично через прапорці ролей.

Корисні посилання

- Install/Update Icinga Agent.

https://icinga.com/docs/icinga-for-windows/latest/doc/frameworkusage/35-Install-Update-Icinga-Agent/

- ICINGA Package Repository. Windows.

https://packages.icinga.com/windows/

- Icinga. Quickstart.

https://icinga.com/docs/get-started/latest/