ВІДЗИВ НАУКОВОГО КЕРІВНИКА НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ

Студента Пахлюка Максима Олеговича
представлену на здобуття освітнього ступеня бакалавр
зі спеціальності 123 «Комп'ютерна інженерія»

1. Робота виконана на тему „Проект підсистеми журналювання подій гетерогенної комп'ютерної мережі".

2. Актуальність роботи зумовлена потребою у розробці та впровадженні у практику експлуатації гетерогенних комп'ютерних мереж систем журналювання подій, засобів аналізу системних журналів, моніторингових систем .

3. Основний зміст пояснювальної записки роботи викладено на 57 сторінках тексту, записка містить 14 таблиць, 44 ілюстрації, 2 додатки. Список використаних літературних джерел налічує 20 найменувань. Невід'ємною складовою роботи є проект мережі, реалізований у середовищі моделювання GNS3.

4. Пояснювальна записка складається з вступу, трьох розділів основного тексту, висновків, списку використаних джерел, додатків.

Розділ 1 присвячений аналізу принципів, протоколів та технологій журналювання подій, зокрема, у розділі розглянуто: принципи журналювання подій, аналіз протоколів та технологій журналювання, організацію журналювання подій в ОС Unix/Linux та ОС Windows.

Розділ 2 присвячений питанням проектування комп'ютерної мережі організації, зокрема, у ньому здійснено проектування топології мережі, розроблено схему адресації мережі, розроблено підсистему журналювння подій.

Розділ 3 присвячений питанням вибору обладнання та програмного забезпечення для розробленого проекту мережі. У розділі наведено опис процесу налагодження та тестування побудованої підсистеми журналювання подій. Також у цьому розділі здійснено перевірку роботи підсистеми журналювання.

5. Висновки містять сумарні результати проведеної роботи.

6. Список використаних літературних джерел має достатній для даного виду робіт обсяг, є правильно оформленим та повністю опрацьованим у ході виконання роботи.

7. Додатки містять схему змодельованої мережі, лістинги конфігураційних файлів налагоджених пристроїв.

8. Таблиці та ілюстрації оформлені досить якісно, є інформативними та наведені у відповідних місцях роботи.

9. Якість оформлення роботи є досить високою.

10. Виявлені на етапі попереднього захисту недоліки оформлення пояснювальної записки та презентації усунені майже повністю. Рекомендації стосовно покращення проекту враховані.

11. Ставлення студента до виконання випускної кваліфікаційної роботи – відповідальне.

12. У цілому при написанні випускної кваліфікаційної роботи студент Пахлюк М.О. виявив здібності до оволодіння знаннями, уміннями і навичками з таких дисциплін, як «Операційні системи», «Комп'ютерні мережі», «Захист інформації в інформаційно-комунікаційних системах», «Адміністрування комп'ютерних систем та мереж». Також студент Пахлюк М.О. показав вміння підібрати необхідну літературу та інші інформаційні джерела, проаналізувати їх, виконати постановку задачі на розробку, розробити проект мережі та підсистеми журналювання, обрати відповідний інструментарій, створити закінчений цілісний продукт (у вигляді змодельованого проекту) і оформити свої розробки та напрацювання як випускну кваліфікаційну роботу.

Випускна кваліфікаційна робота студента Пахлюка М.О. написана на досить високому рівні, відповідає вимогам стандарту вищої освіти зі спеціальності 123 «Комп'ютерна інженерія» і може бути допущена до захисту в Екзаменаційній комісії.

Пропонована оцінка – «Добре».

Науковий керівник
к.т.н., доцент, доцент кафедри КІ та КБ Ю.М. Россінський
01.06.2020 р.

РЕЦЕНЗІЯ
на кваліфікаційну роботу
студента 4 курсу групи КІ-2 спеціальності 123 «Комп'ютерна інженерія»
факультету інформаційно-комп'ютерних технологій
Державного університету «Житомирська політехніка»
Пахлюка Максима Олеговича
на тему: „Проект підсистеми журналювання подій гетерогенної комп'ютерної мережі"

Кваліфікаційна робота студента Пахлюка М.О. „Проект підсистеми журналювання подій гетерогенної комп'ютерної мережі " присвячена актуальній тематиці – розробці та впровадженню у практику експлуатації мереж підприємств засобів фіксації та обліку подій різного характеру та походження, що відбуваються на кінцевих вузлах та комунікаційному обладнанні мережі.

Робота містить реалізований у середовищі моделювання GNS3 проект мережі на базі та пояснювальну записку наступного складу: вступ, три розділи основного тексту, висновки, список використаної літератури, два додатки.

У першому розділі пояснювальної записки проведено аналіз принципів, технологій та протоколів ведення системних журналів. Зокрема, у розділі розглянуто принципи організації журналювання подій, проведено аналіз функціоналу протоколів та технологій журналювання подій в Unix/Linux та Windows системах.

У другому розділі пояснювальної записки здійснено проектування комп'ютерної мережі організації, зокрема, у ньому здійснено проектування топології мережі, розроблено схему адресації пристроїв, розроблено підсистему журналювання подій..

У третьому розділі пояснювальної записки здійснено вибір обладнання та програмного забезпечення для розробленого проекту мережі. У розділі наведено опис процесу налагодження та тестування побудованої мережі. Також у цьому розділі здійснено перевірку функціоналу підсистеми журналювання подій.

Зміст роботи відповідає поставленому завданню. У цілому кваліфікаційна робота виконана на належному рівні. Тема роботи розкрита в достатній мірі. Студентом продемонстровані належні фахові компетенції, які отримані як під час навчання, так і у процесі виконання роботи. Матеріал пояснювальної записки викладено достатньо логічно і технічно грамотно. Ілюстративний матеріал (рисунки, таблиці), що наведений у роботі, оформлений досить якісно і наведений у відповідних місцях. Оформлення роботи у цілому відповідає вимогам стандартів. Функціонал реалізованого проекту мережі відповідає заявленому.

Позитивною стороною є те, що отримані результати мають практичну цінність – елементи розробленого проекту може бути впроваджені у процес експлуатації мережі підприємства. У роботі наявні такі недоліки: недостатньо глибоко проаналізовані питання централізованого ведення журналів гетерогенної мережі та питання опрацювання журнальних даних; в роботі наявні деякі відхилення від вимог щодо оформлення пояснювальної записки, деякі орфографічні та граматичні помилки. Зазначені недоліки у цілому не впливають на досить високу якість і актуальність роботи.

У цілому кваліфікаційна робота студента Пахлюка М.О. „Проект підсистеми журналювання подій гетерогенної комп'ютерної мережі" відповідає вимогам рівня вищої освіти «бакалавр» спеціальності 123 «Комп'ютерна інженерія» і рекомендується до захисту в ЕК.

Пропонована оцінка роботи – «Добре».

Рецензент,
Завідувач кафедри ІПЗ,
к.т.н., доцент
10 червня 2020 р.

І.В. Пулеко,

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп’ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра

на тему:

«Проект підсистеми журналювання подій гетерогенної мережі»

Виконав студент 4-го курсу, групи КІ-2
спеціальності 123 «Комп’ютерна інженерія»
_____ М.О. Пахлюк
Керівник доцент кафедри КІ та КБ,
к.т.н., доцент
_____ Ю.М. Россінський
Рецензент завідувач кафедри ІПЗ,
к.т.н., доцент
_____ І.В. Пулеко

Житомир – 2020

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ

СПЕЦІАЛЬНІСТЬ 123 «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

ЗАТВЕРДЖУЮ
Завідувач кафедри
комп'ютерної інженерії
та кібербезпеки
Єфіменко А. А.
«06» квітня 2020 р.

ЗАВДАННЯ

на випускну кваліфікаційну роботу

Студента **Пахлюка Максима Олеговича**

Тема роботи: **Проект підсистеми журналювання подій гетерогенної комп'ютерної мережі**
затверджена Наказом по університету від **«06» квітня 2020 р. № 98с**
Термін здачі студентом закінченої роботи **– 1 червня 2020 р.**
Вихідні дані для роботи:

1. Фізична топологія мережі
2. Маршрутизатори Cisco 7200
3. Комутатори Cisco 2960
4. Операційні системи Cisco IOS, Windows 7, Windows Server 2019, Kali Linux
5. Графічний симулятор мереж GNS3

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Зав. кафедри КІ та КБ, к.т.н. Єфіменко А. А.	06.04.2020 р.	06.04.2020 р.
2	Зав. кафедри КІ та КБ, к.т.н. Єфіменко А. А.	27.04.2020 р.	27.04.2020 р.
3	Зав. кафедри КІ та КБ, к.т.н. Єфіменко А. А.	18.05.2020 р.	18.05.2020 р.

Керівник _____ Россінський Ю.М.

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	6 квітня 2020 – 7 квітня 2020	Виконано
2	Пошук, огляд та аналіз літературних джерел щодо журналювання подій в гетерогенних комп'ютерних мережах	8 квітня 2020 – 11 квітня 2020	Виконано
3	Аналіз технологій журналювання подій	12 квітня 2020 – 16 квітня 2020	Виконано
4	Аналіз можливостей мережного обладнання Cisco для організації журналювання подій	17 квітня 2020 – 19 квітня 2020	Виконано
5	Розробка загального проекту мережі організації	20 квітня 2020 – 24 квітня 2020	Виконано
6	Проектування підсистеми журналювання подій та вибір обладнання для її реалізації	25 квітня 2020 – 30 квітня 2020	Виконано
7	Проектування та створення сценаріїв тестування функціоналу підсистеми	5 травня 2020 – 9 травня 2020	Виконано
8	Базове налагодження пристроїв мережі	10 травня 2020 – 13 травня 2020	Виконано
9	Налагодження підсистеми журналювання подій	14 травня 2020 – 18 травня 2020	Виконано
10	Тестування роботи підсистеми журналювання подій	19 травня 2020 – 22 травня 2020	Виконано
11	Аналіз результатів тестування	23 травня 2020 – 24 травня 2020	Виконано
12	Написання пояснювальної записки	25 травня 2020 – 29 травня 2020	Виконано
13	Підготовка презентації та виступу	30 травня 2020 – 31 травня 2020	Виконано
14	Отримання відгуку керівника та рецензії на кваліфікаційну роботу	4 червня 2020 – 8 червня 2020	Виконано
15	Захист	13 червня 2020	Виконано

Студент _____ Пахлюк М.О.

Керівник _____ Россінський Ю.М

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з проекту підсистеми журналювання подій гетерогенної комп'ютерної мережі та пояснювальної записки. Пояснювальна записка до випускної кваліфікаційної роботи містить 86 сторінок, 44 ілюстрацій та 14 таблиць.

Метою роботи є розробка проекту підсистеми журналювання подій гетерогенної комп'ютерної мережі.

В роботі визначено основні завдання на розробку проекту, проаналізовано аналоги розроблених проектів. Наведено поетапне налаштування проекту, загальний вигляд мережі. Також наводяться результати тестування мережі.

У першому розділі було обґрунтовано загальні принципи та аналіз програмного забезпечення пов'язані з Cisco system, IOS та підходи роботи журналювання подій.

У другому розділі було спроектовано топологію мережі та розроблено схему адресації.

У третьому розділі було підібрано та налагоджено оптимальне апаратне забезпечення для реалізації комп'ютерної мережі.

					КІК.КР.Б-123-20-ПЗ		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Пахлюк М.О.		10.12.20	Проект підсистеми журналювання подій гетерогенної комп'ютерної мережі	Літ.	Арк. 4 / Аркушів 78
Керівник		Россінський Ю.М.		10.12.20			
Н.контр		Єфіменко А.А.		10.12.20		ФІКТ, група КІ-2	
Рецензент		Пулеко І.В.		1.10.20			
Зав. каф.		Єфіменко А.А.		1.10.20			

ABSTRACT

The bachelor's thesis consists of a project of the event logging subsystem of a heterogeneous computer network and an explanatory note. The explanatory note to the qualification work contains 80 pages, 42 illustrations and 15 tables.

The aim of the work is to develop a project of the event logging subsystem of a heterogeneous computer network.

The main tasks for project development are defined in the work, analogues of the developed projects are analyzed. The step-by-step setup of the project, the general view of the network is given. The results of network testing are also given.

The first section substantiates the general principles and analysis of protocols and software products in Windows and Unix / Linux systems and approaches to event logging.

In the second section, the network topology was designed and an addressing scheme was developed in which attention was paid to address savings and divided into subnets with optimal masks.

In the third section, the hardware that best suited our goals and supported the technologies was selected, routers, switches, and end devices were configured, and network performance was tested by various teams.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Syslog – Стандарт відправи та реєстрації повідомлень.

WMI – Windows Management Instrumention (технологія для управління за роботою різних частин комп’ютеної інфраструктури)

DMZ – Demilitarized Zone (*з англ.* Демілітаризована зона)

PRTG – *Paessler Router Traffic Grapher* (*з англ.* предназначенная для мониторинга использования сети)

MIB – Management Information Base (з англ. виртуальная база данных)

ПК – персональний комп’ютер

ЗМІСТ

РЕФЕРАТ 4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 5
ВСТУП 7
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ 9
1.1. Принципи журналювання подій 9
1.2. Аналіз протоколів та технологій журналювання 11
1.3. Організація журналювання подій в Unix/Linux системах 31
1.4. Організація журналювання подій в Windows системах 35
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТОПОЛОГІЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ ТА РОЗРОБКА СИСТЕМИ АДРЕСАЦІЇ 45
2.1. Розробка топології комп'ютерної мережі 45
2.2. Розробка схеми адресації мережі 48
2.3. Проект підсистеми журналювання подій 51
РОЗДІЛ 3. АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНОЇ МЕРЕЖІ НАЛАГОДЖЕННЯ ОБЛАДНЕННЯ 52
3.1. Вибір обладнання для організації роботи мережі 52
3.2. Налаштування обладнання мережі. 57
3.3. Перевірка працездатності мережі 65
ВИСНОВКИ 69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 70
ДОДАТКИ 72

ВСТУП

В останні десятиріччя змінилося ставлення суспільства до інформації. Вона стала не тільки предметом ретельного вивчення і споживання на рівні з енергією, масою, простором, часом, узятими разом, але й найактуальнішим і найдефіцитнішим видом ресурсу у всіх сферах діяльності людини.

Напрямки застосування нових інформаційних технологій на базі електронно-обчислювальної техніки і розвинутих засобів комунікацій в управлінській діяльності підприємств країни різноманітні – від забезпечення найпростіших функцій службового листування до системного аналізу і підтримки складних задач прийняття рішень. Особливе місце займають і комп'ютерні мережі.

Дослідженням побудови ефективних комп'ютерних мереж людство займається досить довго. Принципи комутації пристроїв мережі та методи передачі даних є ключовими і сьогодні. Тому кваліфікаційна робота присвячена актуальній темі проектування комп'ютерної мережі та направлений на забезпечення взаємодії користувачів системи з належною якістю та мінімізацією затримки передачі даних.

Журналювання подій – це певний процес запису інформації про події що відбуваються в системі.

Відомо, що будь-яка сучасна система для прискорення файлових операцій змушена використовувати кешування, в тому числі – кешування операцій запису. Так званий відкладений запис – принцип кешування, при якому дані, призначені для запису на диск, деякий час зберігаються в кеші і лише у вільний від інших занять час зберігаються фізично. Відкладений запис істотно підвищує ефективність дискових операцій, так як таке кешування групує безліч операцій в одну – це особливо ефективно, якщо запис проводиться в компактні ділянки диска. Ще один плюс відкладеного запису – не заважати більш потрібним операціями читання, і здійснювати запис тільки тоді, коли система вільна і їй не потрібно доступ до диска для інших потреб. Як узгодити відкладений запис з журналюванням? Це досить складне питання, тому що відкладений запис робить

можливим втрату тих даних, які перебували в черзі на фізичний запис, і не встигли записатися на диск до збою. Найнеприємніше тут навіть не втрата даних, а те, що відбувається неузгодженість часу запису: якісь службові області можуть бути оновлені, а якісь суміжні за змістом – ще ні, так як їх оновлення могло бути відкладено ще на пару секунд і не відбулося через збій.

NTFS справляється з цими проблемами за допомогою смислової інтеграції операцій відкладеного запису і ведення журналу. При спробі почати журнальовану операцію в лог тут же записується намір – наприклад, стерти файл. Це трапляється без затримок – на цьому етапі відкладений запис не працює: це плата за присутність журналювання, якої не можна уникнути. Але от всі інші операції вже йдуть в затриманому режимі – тобто вони можуть відбутися частково (можуть ще на додачу і не в тому порядку) або не відбудутися взагалі. Єдина операція з затримкою, робота якої дещо відрізняється від простого запису – запис в лог про вдале завершення попередніх транзакцій, так звана контрольна точка. Через певні проміжки часу – зазвичай через кожні кілька секунд – система в обов'язковому порядку скидає абсолютно всі затримані операції на диск. Після здійснення цієї операції в журнал записується контрольна точка, яка говорить про те, що всі попередні операції виконані коректно на всіх рівнях – як на логічному, так і на фізичному.

Такий режим роботи – за допомогою записів і контрольних точок – з одного боку, як і раніше гарантує повністю коректну роботу журналювання, а з іншого боку практично зовсім не приводить до уповільнення роботи: простановка контрольних точок проводиться, вважай, миттєво, а запис в журнал про початок операції відповідає за об’ємні записи самих даних без відкладеного кешування. Реальна ж запис, здійснювана пізніше, в переважній кількості випадків не заважає ніяким операціям і не йде на шкоду продуктивності системи.

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

1.1. Принципи журналювання подій

Журнальована файлова система – система (ФС), в якій здійснюється ведення журналу, що зберігає список змін і, в тій чи іншій мірі, допомагає зберегти цілісність файлової системи при збоях.

Журнальована файлова система зберігає список змін, які вона буде проводити з файлової системою, перед фактичним їх здійсненням. Ці записи зберігаються в окремій частині файлової системи, яку називають журналом (англ. Journal або Log). Як тільки зміни файлової системи внесені до журналу, вона застосовує ці зміни до файлів або метаданих, а потім видаляє ці записи з журналу. Записи журналу організовані в набори пов'язаних змін файлової системи.

При перезавантаженні комп'ютера програма монтування може гарантувати цілісність журнальованої файлової системи простою перевіркою лог-файлу на наявність очікуваних, але не застосовувати різні модифікації і подальшим записом їх в файлову систему. Тобто, при наявності журналу в більшості випадків системі не потрібно проводити перевірку цілісності файлової системи. Відповідно, шанси втрати даних у зв'язку з проблемами в файлової системі значно знижуються.

За типом внесення в журнал ФС поділяються на:

- в режимі зворотного зв'язку (журналюються тільки метадані): XFS, ext3fs;
- впорядковані (журналюються тільки метадані синхронно щодо даних): JFS2, ext3fs (за замовчуванням), ReiserFS (основний);
- в режимі даних (журналюються як метадані, так і дані): ext3fs.

Компанія Microsoft запропонувала ряд простих схем для оцінки та розробки безпечного програмного забезпечення, для оцінки загроз та атак і оцінки наслідків атаки.

Схема (формула) SD3C визначає основні принципи розробки безпечного програмного забезпечення:

– Secure in Design – застосування принципів безпечного проектування; облік можливих атак; реалізація способів їх відображення;

– Secure by Default – включення установок безпеки за замовчуванням;

– Secure in Deployment – безпечне розгортання і інсталяція програмного забезпечення;

– Communication – постійна взаємодія групи супроводу продукту з користувачами, швидкий випуск нових версій продукту з виправленими помилками безпеки; рекомендації по налаштуванню безпеки.

Принципи:

– зменшувати «площу» додатку, відкриту для нападів;

– створювати систему безпеки з захистом на всіх рівнях;

– використовувати мінімальні привілеї;

– в конфігурації за замовчуванням призначати безпечні параметри;

– пам'ятати, що зворотна сумісність завжди несе за собою проблеми;

– завжди припускати незахищеність зовнішніх систем;

– передбачати план дій при збоях і відмовах;

– подбати, щоб при будь-яких збоях система зберігала конфіденційність інформації;

– пам'ятати, що можливості підсистеми безпеки – це не те ж саме, що безпечні можливості системи;

– ніколи не покладатися на захист, засновану тільки на обмеженості інформації;

– не змішувати код і дані; коректно виправляти помилки в підсистемі безпеки.

1.2. Аналіз протоколів та технологій журналювання

На даний час існують знайомі усім традиційні підходи до журналювання подій, які в основному пов'язані зі збором повідомлень з різних джерел на центральному сервері з метою їх архівації. Такий тип журналювання – це краще, ніж нічого, але сучасна архітектура журналювання може запропонувати набагато більше. Рівень збору передає журнали від додатків, систем, мережевого обладнання та сторонніх сервісів і направляє їх в центральне розташування. Далі ми докладно обговоримо збір журналів і перерахуємо їх типи, які варто збирати. Після збору повідомлення журнали передаються на рівень потокової передачі, який зазвичай реалізується у вигляді брокера повідомлень, такого як RabbitMQ або Apache Kafka. Призначення цього рівня – концентрація журналів в одному конвеєрі, в якому можна здійснювати перенаправлення.

Обробка журналів відбувається на рівні аналізу. Саме тут сучасний конвеєр журналювання починає відрізнятися від традиційних технік. Цей рівень складається з невеликих програм, призначених для поглинання повідомлень журналів і виконання над ними певних маніпуляцій. Одні аналізатори зберігають журнали в базі даних, інші обчислюють статистику, а треті, найбільш цікаві для нас, можна налаштувати на виявлення аномалій, вторгнень і шаблонів атак.

Далі йде рівень зберігання, і, хоча його концепція на перший погляд може здатися простою, зберігання великої кількості даних, що містяться в журналах, перетворить цю задачу в захоплююче випробування. У минулому журнали зберігалися в файлах журналів до того часу, коли від них потрібно було позбавлятися. Сьогодні поширеною практикою є завантаження журналів в базу даних для швидкого доступу і приміщення старих журналів в сховище з більш повільним і складним доступом.

І нарешті, рівень доступу забезпечує адміністраторів інтерфейсом для розгляду журналів під різними кутами. Якщо все в порядку, панель спостереження відображає те, що люди хочуть бачити, але важливість хорошого інтерфейсу для доступу до необроблених даних не варто недооцінювати:

кращими друзями досліджує найчастіше стають прості Unix-інструменти, такі як grep, sed, awk, і кілька рядків bash -скріптов. Така архітектура складна, її реалізація потребує багато часу і ресурсів.

Під час розробки програмного забезпечення чи навіть невеликої програми, розробники додають логування, протоколювання до своїх систем задля моніторингу роботи, відловлення помилок, отримання інформації про використану пам'ять, тощо. Результатом може накоплюватися велика кількість розкиданих текстових і бінарних файлів зі зрозумілими і незрозумілими розширеннями, незрозумілого формату і змісту. Зрозуміло, що це все є наслідками неправильної організації.

Для вирішення цієї проблеми операційна система Windows надає такий сервіс і програмний інтерфейс, як Eventlog. Цей інструментарій належить до базових сервісів Windows, тобто поставляється з самою системою і система сама ж його використовує. Варто зауважити, що ця можливість є тільки у систем сімейств WinNT/XP, тому що додаток для протоколювання подій є сервісом (рис.1.1).

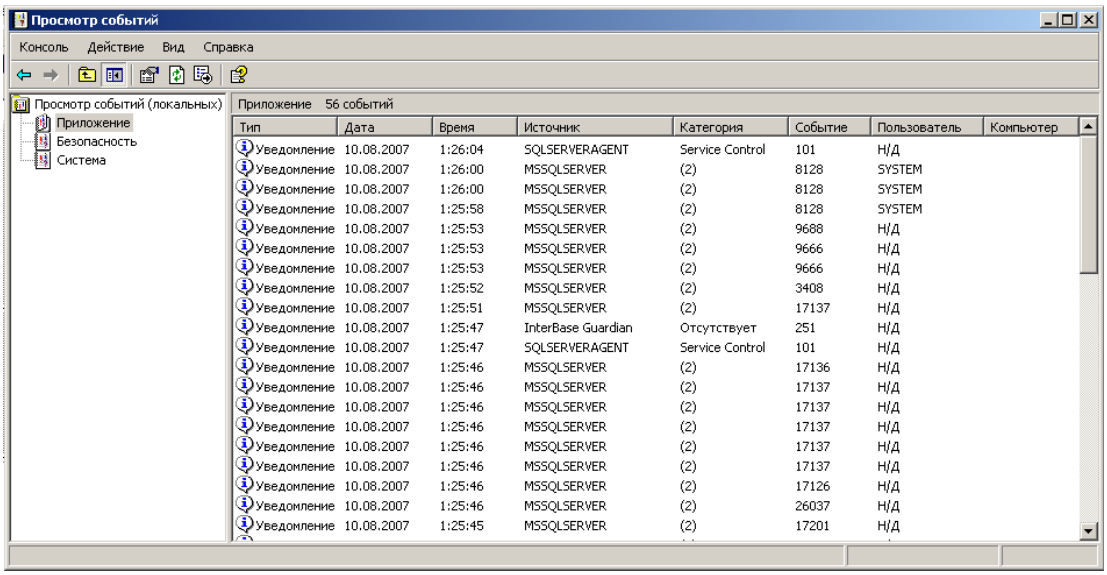


Рис.1.1. Програмний інтерфейс Eventlog

Напевно, майже всі розробники використовують в своїх програмах протоколювання подій при виявленні помилок, налагодженні і діагностуванні додатків. Але навіть після успішного збору і перегляду статистики іноді буває складно проаналізувати, що ж все-таки сталося і в якому місці. Так ось, сервіс

«Журнал подій (Evenlog)» є стандартним, централізованим способом збору статистики і перегляду повідомлень про події, що надходять від додатків, сервісів операційної системи і апаратних пристроїв. Для моніторингу цих подій використовується оснащенням Microsoft Management Console.

При описі подій в журналі важливо дотримуватися певних рекомендацій, що допоможе зробити журнали подій легкими для читання та охайними.

Повідомлення в журналі подій – це, перш за все, інформація, здатна допомогти розробнику, адміністратору і навіть користувачеві зрозуміти, яка проблема виникла в додатку і як її усунути. Розглянемо наступний приклад повідомлення: «Процес А не зміг прочитати 0x05 байт 0x2-ого сектора дисковода В». Даний результат є незрозумілим не тільки для користувача, але й для самого розробника чи системного адміністратора.

Тому ідеальне повідомлення повинно допомогти користувачеві відповісти на наступні питання:

- Що трапилося і чому?;
- Що йому (користувачеві) робити далі?;
- Що він (користувач) може зробити, щоб цього більше не повторилося?;

Можуть стати в нагоді і наступні рекомендації:

- Уникайте умовних помилок. Якщо ви можете спрогнозувати помилку, яка може трапитися в разі виконання певної дії, перепишіть свій код так, щоб користувач не отримував повідомлення про помилку;
- Напишіть текстове повідомлення для кожної відомої помилки;
- Використовуйте системні коди і повідомлення про помилки;
- Надайте користувачеві хоча б один варіант вирішення проблеми, що виникла внаслідок вашої помилки;
- Не використовуйте технічні терміни, жаргон, сленг і абревіатури;
- У діалогових повідомленнях використовуйте саме необхідні кнопки (такі як Yes, No, Cancel).

Угоди про стилі змісту повідомлення:

– Використовуйте докладні, але прості речення;

– Не використовуйте слова, що складаються з одних великих букв;

– Використовуйте точну семантику. Наприклад, замість повідомлення "Bad size" краще все-таки вказати користувачеві правильний розмір;

В журнал можна записувати п'ять типів подій. Всі типи подій досить зрозуміло класифіковані, визначені і можуть включати багато додаткової інформації. Кожна з подій, яку ми посилаємо зі свого додатку (системи), може мати тільки один тип.

Таблиця 1.1

Типи подій

Тип події	Пояснення
Помилка (Error)	Цим типом зазвичай визначається серйозна помилка програми. Наприклад, виконання програми перервалося через брак ресурсів.
Попередження (Warning)	Цим типом додаток зазвичай інформує про те, що скоро може виникнути проблема, наприклад, закінчиться дисковий простір.
Інформація (Information)	Цим типом додаток зазвичай інформує про успіх будь-якої важливої операції, наприклад, при старті сервісу.
Успішний звіт (Success Audit)	Цей тип події зазвичай означає про успіх будь-якої операції доступу, наприклад, користувач увійшов в систему.
Чи не успішний звіт (Fault Audit)	Цей тип події зазвичай означає, що сталася якась помилка при доступі до ресурсу, наприклад, користувач не зміг звернутися до мережного диска.

Оскільки запис в журнал може відбуватися нечасто, в тексті повідомлення слід досить повно описувати кожну подію, що відбулася. Слід досить зрозуміло описувати подію, так як інформацію про неї, до того як вона потрапить до розробника, може переглянути, наприклад, адміністратор системи і не надавши їй особливого значення, просто проігнорувати.

У тексті повідомлення про подію краще не застосовувати символи табуляції і крапку з комою, так як журнал може експортуватися в текстовий файл з роздільниками. Багато організацій імпортують журнали подій в свої бази даних

для діагностики своїми засобами. Рядки в форматі UNC краще укладати в трикутні дужки, наприклад <\\ sharename \ servername>.

Всю інформацію про атрибути журналу сервіс бере з реєстру:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog.

Кожна гілка в цьому ключі – це журнал. Спочатку їх всього три, але можна створювати свої журнали, і реєструвати свої додатки як джерела подій для цих журналів.

Таблиця 1.2

Типи журналів

Журнал (Log)	Пояснення
Додаток (Application)	Цей журнал містить записи від додатків. Наприклад, якщо ми зареєструємо своє джерело подій (тобто додаток) і не вкажемо журнал, за замовчуванням записи будуть надходити сюди.
Система (System)	Цей журнал містить записи, що надходять від системних служб. Але писати в нього може будь-який додаток.
Безпека (Security)	Цей журнал призначений для аудиту, наприклад, подій входу користувача в систему.
Інший (Custom)	Можливість створити свій власний журнал.

Контролер домену має ще два додаткових журнали: Directory і File Replication. DNS-сервер також має додатковий журнал, DNS.

Для кожного журнала визначаються ключі. Практично всі вони є обов'язковими і мають значення за замовчуванням, але рекомендується все ж.

Таблиця 1.3

Ключі в журналах

Ключ в реєстрі	Пояснення
MaxSize	Максимальний розмір журналу в байтах. За замовчуванням 512K. Параметр DWORD.
DisplayNameID	Ідентифікатор рядка в ресурсній DLL. Тип параметра DWORD.
MaxSize	Максимальний розмір журналу в байтах. За замовчуванням 512K. Параметр DWORD.
PrimaryModule	Найменування ключа, де зберігаються настройки за замовчуванням. Зазвичай збігається з найменуванням журналу. Строковий параметр.

Продовження таблиці. 1.3

Retention	Інтервал в секундах, протягом якого записи будуть зберігатися без перезапису. Якщо встановлено в нуль, записи в журналі завжди будуть перезаписуватися, якщо не нуль або 0xFFFFFFFF, то записи ніколи не перезаписуватимуться. При досягненні максимального розміру журнал необхідно очистити вручну, інакше записи будуть втрачені. Перед тим, як змінювати це значення, журнал необхідно очистити. Параметр DWORD. За замовчуванням – 0.
AutoBackupLogFiles	Якщо значення параметра – 0, журнал не зберігає резервну копію. За замовчуванням – 0.
RestrictGuestAccess	Якщо значення – 1, то користувачі під обліковим записом Guest і Anonymous не мають доступу до журнала. За замовчуванням – 0.
Isolation	Не використовується.
Sources	Список додатків, сервісів, які можуть писати в журнал. Тільки для читання. Цей список створює сам сервіс. Назви додатків беруться з поточної гілки журналу і поділяються null-terminated. Параметр багатостроковий.
DisplayNameFile	Файл, в якому міститься локалізований рядок з назвою журналу, тобто рядок, який покаже Event Viewer. Якщо параметр не вказано, Event Viewer, як рядок, покаже найменування підключа, в якому визначено параметр. За замовчуванням всі локалізовані ресурси знаходяться в %SystemRoot%\system32\ELS.DLL. Строковий параметр.
File	Шлях до папки, де Event Viewer буде зберігати файли журналів. За замовчуванням це %SystemRoot%\system32\config\MyLogName, де MyLogName – ім'я журналу. При створенні нового файлу журналу сервіс повинен мати права на повний доступ до файла. Якщо значення цього параметра буде невірним, всі записи будуть перенаправлятися в журнал Application. В дорозі не можна використовувати ім'я віддаленого комп'ютера, DOS-пристрої, дисководи, іменовані канали. Не можна використовувати змінні оточення, які не можна розкрити в контексті сервісу.

Кожна внутрішня гілка в основній гілці Eventlog – це джерело подій.

Наприклад, для додатка MySuperApp.EXE, яке буде записувати події в журнал Application, необхідно створити таку внутрішню гілку:

HKEY_LOCAL_MACHINE \ SYSTEM \ CurrentControlSet \ Services \ Eventlog \ Application \ MySuperApp

Тут MySuperApp – це довільне ім'я, по якому сервіс (журнал) буде впізнавати події, що надходять від нашого застосування. Яких-небудь угод про ім'я в документації не зазначено, але ім'я має бути унікальним в межах однієї

внутрішньої гілки. Зазвичай використовується назва програми чи модуля. Так відбувається реєстрація додатка в сервісах Event Logging та Event Viewer.

Саме це ім'я буде передаватися функції RegisterEventSource, яка поверне обробник (handle) журналу.

Користувальницькі додатки і сервіси повинні або реєструвати себе в журналі Application, або створювати свій журнал. Журнал Security використовується тільки системою. Драйвери пристроїв повинні використовувати журнал System.

Кожне джерело подій (додаток) повинно у своїй гілці визначити перераховані ключі. Вони допомагають Event Viewer зіставляти повідомлення і підставляти параметри з ресурсної DLL-бібліотеки з ідентифікаторами в додатку.

Таблиця 1.4.

Ідентифікатори подій

Ключ в реєстрі	Пояснення
CategoryCount	Число використовуваних категорій. Тип DWORD.
CategoryMessageFile	Шлях до файла з локалізованими рядками, визначаеними категоріями. Строковий параметр.
EventMessageFile	Шлях до файла з локалізованими рядками повідомлень. Можна перерахувати декілька файлів через кому. Наприклад, EVT_ENG.DLL, EVT_RUS.DLL. Строковий параметр.
ParameterMessageFile	Шлях до файла з локалізованими рядками параметрів, підставляється в текст повідомлення. Строковий параметр.
TypeSupported	Параметр, що визначає типи підтримуваних повідомлень. Наприклад, тільки помилки і попередження: EVENTLOG_ERROR_TYPE | EVENTLOG_WARNING_TYPE. Параметр DWORD визначає маску з наступних типів: EVENTLOG_ERROR_TYPE, EVENTLOG_WARNING_TYPE, EVENTLOG_INFORMATION_TYPE, EVENTLOG_AUDIT_SUCCESS, EVENTLOG_AUDIT_FAILURE

Отже, коли додаток викличе функції RegisterEventSource або OpenEventLog, сервіс Eventlog буде шукати в реєстрі гілку додатку, щоб повернути описувач журналу. Якщо він не знайде гілку з ім'ям додатка, за замовчуванням буде використаний журнал Application.

Однак якщо гілка була знайдена, але не були визначені файли повідомлень для знайденого джерела подій, то при відкритті Event Viewer повідомить про помилку. Тому в даному випадку необхідно не тільки зареєструвати джерело подій, а й надати файли повідомлень.

Syslog означає протокол системного журналу і являє собою стандартний протокол, який використовується для надсилання системного журналу або повідомлень про події на певний сервер, який називається сервером syslog. Він в основному використовується для збору різних журналів пристроїв з декількох різних машин у центральному місці для моніторингу та огляду.

На будь-якому даному пристрої система генерує різні події у відповідь на зміни умов. Ці події зазвичай реєструються локально, де їх можна переглянути та проаналізувати адміністратором. Однак моніторинг численних журналів за однаковою кількістю маршрутизаторів, комутаторів та систем буде трудомістким та непрактичним. Syslog допомагає вирішити цю проблему, пересилаючи ці події на централізований сервер.

Традиційно Syslog використовує протокол UDP на порт 514, але може бути налаштований на використання будь-якого порту. Крім того, деякі пристрої використовуватимуть TCP 1468 для надсилання даних системного журналу для отримання підтвердженої доставки повідомлень.

Передача пакетів Syslog є асинхронною. Що змушує генерувати повідомлення syslog, налаштовується всередині маршрутизатора, комутатора чи сервера. На відміну від інших протоколів моніторингу, таких як SNMP, не існує механізму опитування даних syslog. У деяких реалізаціях SNMP може використовуватися для віддаленого встановлення або зміни параметрів syslog.

Повідомлення syslog складається з трьох частин: PRI (обчислене значення пріоритету), HEADER (з ідентифікаційною інформацією) та MSG (саме повідомлення).

Дані PRI, що надсилаються через протокол syslog, походять з двох числових значень, які допомагають класифікувати повідомлення. Перший – значення фонду. Це значення є одним із 15 заздалегідь визначених значень або різних

локально визначених значень у випадку від 16 до 23. Ці значення класифікують тип повідомлення або систему, яка генерувала подію.

Таблиця 1.5.

Числові значення пріоритету

Номер	Опис об'єкта
0	Повідомлення ядра
1	Повідомлення на рівні користувача
2	Поштова система
3	Системні демони
4	Повідомлення безпеки / авторизації
5	Повідомлення, згенеровані syslogd
6	Підсистема лінійного принтера
7	Мережева підсистема новин
8	Підсистема UUCP
9	Годинник Демон
10	Повідомлення безпеки / авторизації
11	FTP Daemon
12	Підсистема NTP
13	Аудит журналу
14	Попередження журналу
15	Годинник Демон
16 - 23	Місцеве використання 0 - 7

Друга мітка повідомлення syslog класифікує важливість або серйозність повідомлення в числовому коді від 0 до 7.

Таблиця 1.6

Класифікація важливості повідомлень

Код	Суворість	Опис
0	Надзвичайна ситуація	Система непридатна
1	Сповіщення	Дії потрібно вжити негайно
2	Критичний	Критичні умови
3	Помилка	Умови помилки
4	Увага	Умови попередження
5	Зауважте	Нормальна, але значна умова
6	Інформаційні	Інформаційні повідомлення
7	Відлагодження	Повідомлення на рівні налагодження

Значення обох міток не мають жорстких визначень. Таким чином, система чи додаток визначають, як зареєструвати подію (наприклад, як попередження, повідомлення чи щось інше) та на якому об'єкті. У межах однієї програми або

послуги менші числа повинні відповідати більш серйозним питанням щодо конкретного процесу.

Два значення поєднуються, щоб отримати значення пріоритету, надіслане разом із повідомленням. Значення пріоритету обчислюється шляхом множення значення Фонду на вісім, а потім додавання до значення Значення серйозності. Чим нижчий показник PRI, тим вище пріоритет.

(Вартість об'єкта * 8) + Значення серйозності = PRI

Таким чином повідомлення ядра отримує нижче значення (більш високий пріоритет), ніж попередження журналу, незалежно від строгості попередження про журнал. Додаткові ідентифікатори в пакеті включають ім'я хоста, IP-адресу, ідентифікатор процесу, ім'я програми та часову позначку повідомлення.

Фактична багатослівність або зміст повідомлення syslog не визначені протоколом. Деякі повідомлення – це простий текст, читабельний, інші – лише машинне для читання.

Повідомлення Syslog зазвичай не перевищують 1024 байт.

<165> 1 2017-05-11T21:14:15.003Z mymachine.example.com appname [su] - ID47 [exampleSDID @ 32473 iut = "3" eventSource ="eventID="1011"] BOMB запис журналу програми.

Розглянемо дане повідомлення окремо по частинам у вигляді наступної таблиці:

Таблиця 1.7

Частини повідомлення syslog

Частина	Значення	Інформація
PRI	165	Об'єкт = 20, тяжкість = 5
ВЕРСІЯ	1	Версія 1
TIMESTAMP	2017-05-11T21:14:15.003Z	Повідомлення створено 11 травня 2017 року о 09:14:15, через 3 мілісекунди наступної секунди
HOSTNAME	mymachine.example.com	Повідомлення походить від хоста "mymachine.example.com"
APP-NAME	Su	Ім'я програми: "su"
PROCEED	-	PROCEED невідомий
MSGID	ID47	ID повідомлення: 47

Продовження таблиці 1.7

Частина	Значення	Інформація
СТРУКТУРОВАНІ ДАНІ	[exampleSDID @32473 iut = "3" eventSource = "eventID=" 1011 "]	Структурований елемент даних з неконтрольованим IANA SD-ідентифікатором типу "exampleSDID @32473 ", який має три параметри
MSG	BOM запис журналу програми ...	BOM вказує кодування UTF-8, саме повідомлення "Запис журналу програми"

Сервер Syslog також відомий як колектор syslog або приймач.

Повідомлення Syslog надсилаються з генераторного пристрою в колектор. IP-адреса сервера цільового системного журналу повинна бути налаштована на самому пристрої, або за допомогою командного рядка, або через конфігураційний файл. Після налаштування всі дані системного журналу будуть відправлені на цей сервер. Немає механізму в протоколі syslog для іншого сервера для запиту даних syslog.

У той час як більшість реалізацій Unix та постачальників мереж, як і Cisco, мають власні колектори syslog barebones, також є кілька інших.

Paessler в PRTG програмного забезпечення моніторингу пропонує вбудований Syslog приймач датчика. Одержувач збирає всі доставлені повідомлення Syslog. Для використання функції адміністратору потрібно додати приймач Syslog, а потім налаштувати IP-адресу цього сервера як сервер призначення для даних даних syslog на всіх пристроях, які слід контролювати.

Після зібрання на інформаційній панелі показано:

- Кількість прийнятих системних повідомлень в секунду;
- Кількість повідомлень, віднесених до категорії "попередження" за секунду;
- Кількість повідомлень, віднесених до категорії "помилка" в секунду;
- Кількість скинутих пакетів за секунду.

Протокол syslog може генерувати безліч повідомлень. Syslog просто пересилає повідомлення так само швидко, як і генерує їх. Як результат, найважливішою здатністю для сервера syslog є можливість належного фільтрування та реагування на вхідні дані системного журналу.

Датчик приймача Syslog PRTG пропонує можливість встановлювати правила фільтрації. Ці правила дозволяють включати або виключати повідомлення системного журналу як попередження або помилки, незалежно від того, як вони були спочатку створені на пристрої. Ця фільтрація забезпечує адміністраторам повідомлення про всі помилки, про які вони хочуть знати, не перевантажуючи їх менш важливими помилками.

Протокол syslog не пропонує механізму захисту. Немає вбудованої автентифікації, яка б гарантувала, що повідомлення надходять від пристрою, який вимагає їх надсилання. Немає шифрування, щоб приховати, яку інформацію надсилає на сервер. Це особливо сприйнятливе до так званих «атак відтворення», коли зловмисник генерує попередній потік попереджень для незаконного реагування.

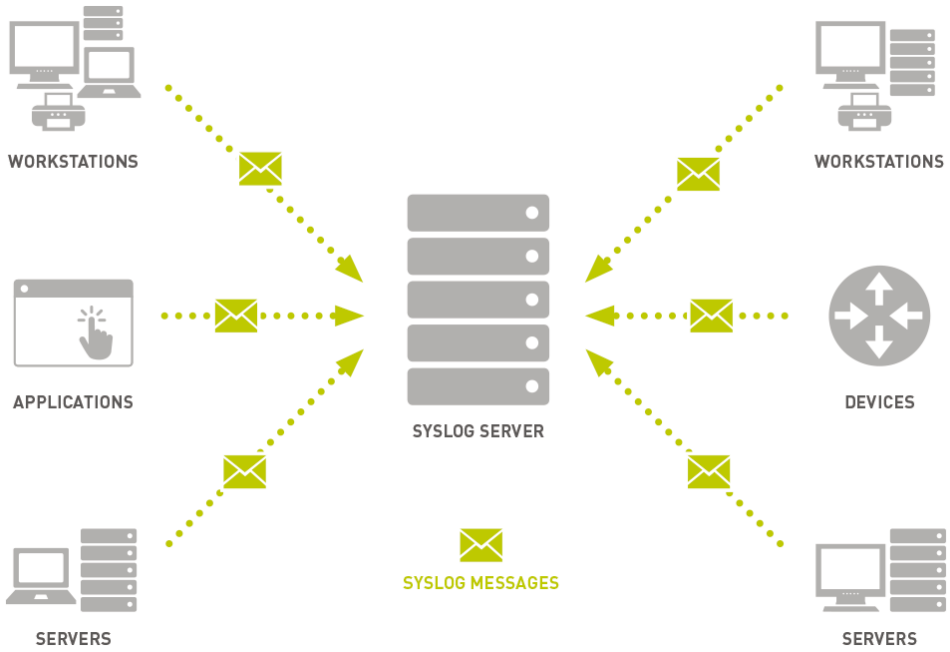


Рис. 1.2. Повідомлення системного журналу

Протокол syslog не пропонує механізму захисту. Немає вбудованої автентифікації, яка б гарантувала, що повідомлення надходять від пристрою,

який вимагає їх надсилання. Немає шифрування, щоб приховати, яку інформацію надсилає на сервер. Це особливо сприйнятливе до так званих «атак відтворення», коли зловмисник генерує попередній потік попереджень для незаконного реагування.

Більшість реалізацій syslog налаштовуються щодо того, які засоби та числа вираженості створюватимуть події syslog, які передаються на сервер syslog. Важливо правильно налаштувати це, щоб уникнути затоплення сервера (та мережі) непотрібним трафіком. Наприклад, налагодження ніколи не повинно встановлюватись для надсилання повідомлень, за винятком тестування.

Одним з головних обмежень протоколу syslog є те, що пристрій, який контролюється, повинен бути запущений та підключений до мережі для створення та надсилання події syslog. Критична помилка з боку ядра може взагалі ніколи не надсилати помилку, коли система переходить в офлайн. Іншими словами, syslog – це не гарний спосіб контролювати стан пристроїв вгору та вниз.

Хоча syslog не є хорошим способом контролю стану мережевих пристроїв, він може бути хорошим способом моніторингу загального стану роботи мережевого обладнання. Хоча програмне забезпечення для моніторингу мережі, як PRTG, пропонує набір утиліт для спостереження за мережею, ніщо не говорить адміністратору, що існує проблема, швидше, ніж журнал подій, заповнений попередженнями. Правильно налаштований моніторинг системного журналу виявить раптове збільшення обсягу та тяжкості події, повідомлення буде отримано до появи проблеми, виявленої користувачем.

Середня корпоративна мережа містить численні пристрої, до яких ніхто не повинен намагатися отримати доступ. Якщо віддалений комутатор, який входить у систему лише один раз за цикл аудиту, раптово має щоденні спроби входу (успішні чи в інший спосіб), він перевіряє. На цих типах пристроїв syslog можна встановити для пересилання подій аутентифікації на сервер syslog, без накладних витрат на встановлення та налаштування повного агента моніторингу.

Syslog також пропонує спосіб переконатися, що критичні події реєструються та зберігаються на вихідному сервері. Першими зусиллями

зловмисника після компрометування системи є покриття слідів, залишених у журналі. Події, передані через syslog, будуть недоступними.

Існує маса способів відстежувати, як програма працює на сервері. Однак ці монітори можуть не помітити, як програма впливає на інші процеси на сервері. Незважаючи на те, що високий рівень використання процесора або пам'яті досить легко виявити за допомогою інших моніторів, зафіксовані події можуть допомогти показати більше можливих проблем. Чи програма постійно намагається отримати доступ до закритого файлу? Чи є спроба запису в базу даних, що генерує помилку? Такі події можуть не помітитись, коли їх спричиняють додатки, які добре вирішують помилки, але їх не слід ігнорувати.

Повний моніторинг мережі вимагає використання декількох інструментів. Найкраща практика – використання програмного забезпечення, яке поєднує всі інструменти, щоб завжди мати огляд того, що відбувається в мережі.

Простий протокол управління мережею (SNMP) – протокол рівня додатків, визначений Інститутом архітектури Інтернету (IAB) в RFC1157 для обміну інформацією про управління між мережевими пристроями. Він є частиною набору протоколів управління передачею⁄ протоколу Інтернет-протоколу (TCP⁄IP).

SNMP – один із широко прийнятих протоколів управління та моніторингу мережевих елементів. Більшість мережевих елементів професійного класу поставляються з комплектом SNMP-агента. Ці агенти повинні бути включені та налаштовані на зв'язок із системою управління мережею (NMS).

SNMP складається з наступних компонентів:

- SNMP Manager.
- Керовані пристрої.
- SNMP агент.
- База даних управління управління інакше називається інформаційною базою управління (MIB).

Менеджер або система управління – це окрема особа, яка відповідає за зв'язок із мережевими пристроями, реалізованими агентом SNMP. Зазвичай це комп'ютер, який використовується для запуску однієї або декількох систем управління мережею.

Основні функції менеджера SNMP:

- Агенти запитів.
- Отримує відповіді від агентів.
- Встановлює змінні агенти.
- Визнає асинхронні події від агентів.

Керований пристрій або мережевий елемент – це частина мережі, яка вимагає певної форми моніторингу та управління, наприклад, маршрутизатори, комутатори, сервери, робочі станції, принтери, ДБЖ тощо.

Агент – це програма, упакована в мережевий елемент. Увімкнення агента дозволяє йому збирати локальну базу даних управління з пристрою і надає доступ до менеджера SNMP, коли він запитується. Ці агенти можуть бути стандартними (наприклад, Net-SNMP) або специфічними для постачальника (наприклад, агентом огляду HP).

Основні функції агента SNMP:

- Збирає інформацію про управління про своє місцеве середовище.
- Зберігає та витягує інформацію про управління, як визначено в MIB.
- Сигналізує подію менеджеру.
- Виступає як проксі-сервер для деякого мережевого вузла, що не керується SNMP.

Кожен агент SNMP підтримує інформаційну базу даних, що описує параметри керованого пристрою. Менеджер SNMP використовує цю базу даних для запиту агента конкретної інформації та надалі перекладає інформацію, необхідну для системи управління мережею (NMS). Ця загальнодоступна база

даних між агентом та менеджером називається інформаційною базою управління (MIB).

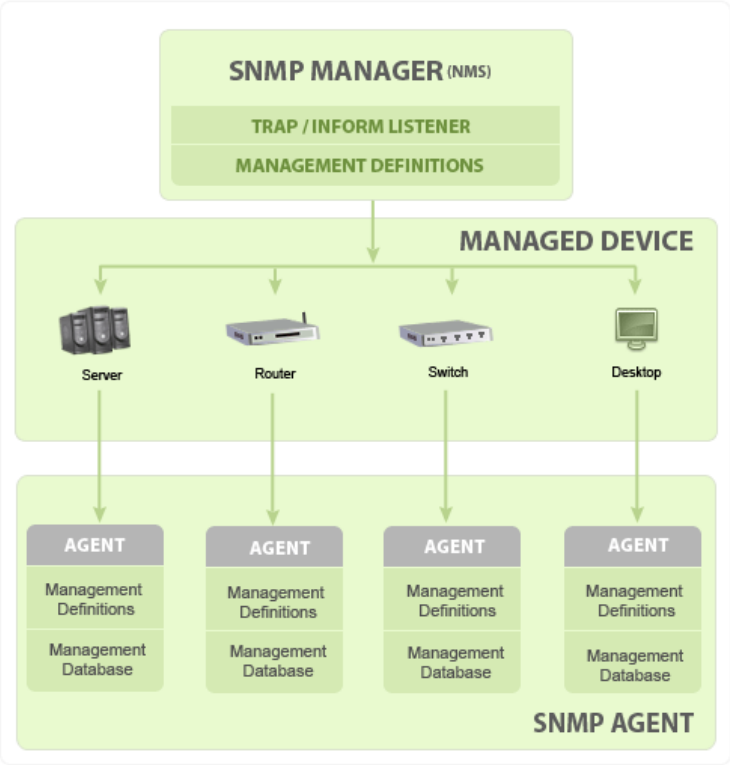


Рис. 1.3. Основна схема зв'язку SNMP

Зазвичай ці MIB містять стандартний набір статистичних та контрольних значень, визначених для апаратних вузлів у мережі. SNMP також дозволяє розширити ці стандартні значення зі значеннями, характерними для конкретного агента за допомогою приватних MIB.

Отже, файли MIB – це набір питань, які менеджер SNMP може задати агенту. Агент збирає ці дані локально та зберігає їх, як визначено в MIB. Отже, менеджеру SNMP слід знати про ці стандартні та приватні питання для кожного типу агентів.

Комплект безкоштовних інструментів ManageEngine включає браузер SNMP MIB, який допомагає завантажувати/вивантажувати MIB та отримувати MIB дані SNMP (v1, v2c, v3) агентів. SNMP MIB Browser – це повний інструмент для моніторингу пристроїв та серверів, що підтримують SNMP. Ви можете завантажувати, переглядати декілька модулів MIB та виконувати операції GET,

GETNEXT та SET SNMP. Цей інструмент простий у використанні та дозволяє переглядати, налаштовувати та аналізувати пастки SNMP. Ви також можете виконувати операції SNMP з пристроїв Windows та Linux.

Інформаційна база управління (MIB) – це сукупність інформації для управління мережевим елементом. MIB включає керовані об'єкти, ідентифіковані іменем ідентифікатора об'єкта (ідентифікатор об'єкта або OID).

Кожен Ідентифікатор є унікальним і позначає конкретні характеристики керованого пристрою. За запитом, повернене значення кожного ідентифікатора може бути різним, наприклад, текст, число, лічильник тощо ...

Існує два типи керованого об’єкта або ідентифікатор об'єкта:

- скалярний;
- табличний.

Вони могли бути краще зрозумілі на прикладі

Скаляр: ім’я постачальника пристрою, результат може бути лише одним. (Як говорить визначення: "Скалярний об'єкт визначає один екземпляр об'єкта")

Таблиця: Використання процесором Quad-процесора, це дасть мені результат для кожного процесора окремо, значить, для цього конкретного ідентифікатора об’єкта буде 4 результати. (Як випливає з визначення: "Табличний об'єкт визначає кілька пов'язаних екземплярів об'єктів, які об'єднані в таблиці MIB")

Кожен ідентифікатор об'єкта організовано ієрархічно в MIB. Ієрархія MIB може бути представлена у структурі дерева з індивідуальним ідентифікатором змінної.

Простота в обміні інформацією зробила SNMP широко поширеним протоколом. Основною причиною є стислий набір команд, ось вони перераховані нижче:

GET: Операція GET – це запит, який менеджер надсилає керованому пристрою. Він виконується для отримання одного або декількох значень з керованого пристрою.

Отримати наступну: ця операція схожа на GET. Суттєва різниця полягає в тому, що операція GET NEXT отримує значення наступного OID у дереві MIB.

Отримати BULK: Операція GETBULK використовується для отримання об'ємних даних з великої таблиці MIB.

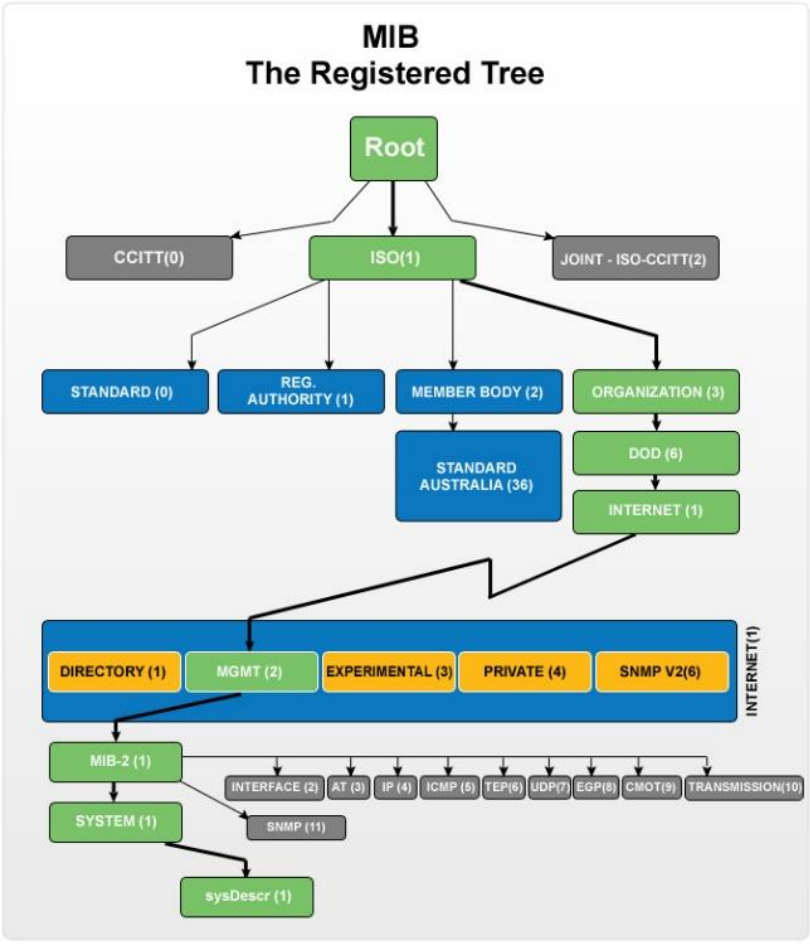


Рис.1.4. Діаграма MIB дерева

SET: Ця операція використовується менеджерами для зміни або призначення значення керованого пристрою.

TRAPS: На відміну від вищезазначених команд, які ініціюються з SNMP Manager, TRAPS ініціюються агентами. Це сигнал агенту менеджера SNMP про настання події.

ІНФОРМАЦІЯ: Ця команда схожа на TRAP, ініційований Агентом, крім того INFORM включає підтвердження від менеджера SNMP про отримання повідомлення.

ВІДПОВІДЬ: Це команда, яка використовується для передачі значення (ів) або сигналу дій, керованих менеджером SNMP.

SNMP-пастки дозволяють агенту повідомляти менеджера SNMP про значні події за допомогою небажаного повідомлення SNMP. Протоколи SNMP Trap включають поточне значення sysUpTime, OID, що ідентифікує тип пастки та необов'язкові змінні прив’язки. Адресація адресатів для пасток SNMP визначається специфічно для додатка, як правило, через змінні конфігурації ловушки в MIB. Формат повідомлення ловушки було змінено в SNMPv2, а одиниці даних протоколу перейменовано в SNMPv2-Trap.

Будучи частиною набору протоколів TCP/IP-повідомлень, SNMP-повідомлення перетворюються на протокол User Datagram Protocol (UDP) та стажуються, обертаються та передаються в Інтернет-протокол. Наступна схема буде ілюструвати чотиришарову модель, розроблену Міністерством оборони (DoD).

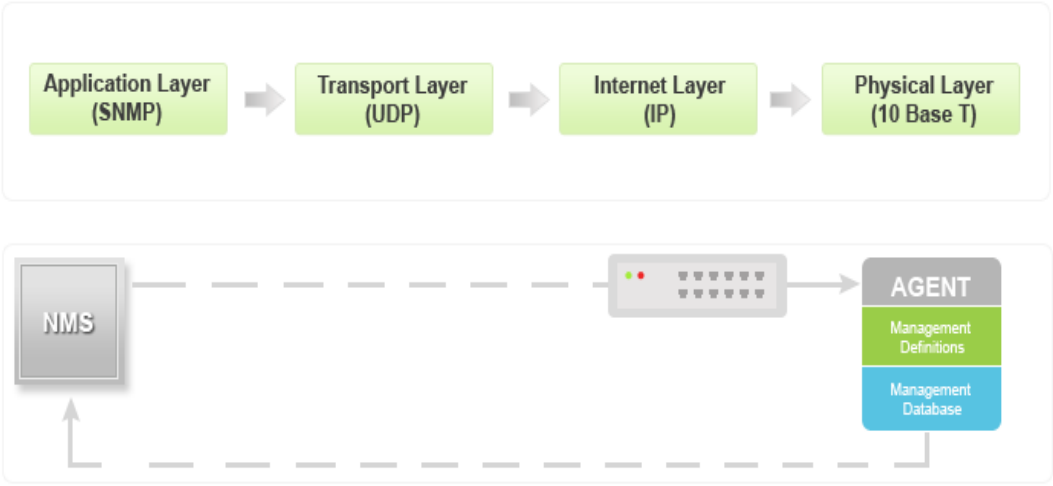


Рис.1.5. Схема чотиришарової моделі

За замовчуванням порт SNMP становить 161, а TRAP/ INFORM використовує порт 16MP SNMP для зв'язку.

З моменту створення SNMP пройшов значні оновлення. Однак протоколи SNMP v1 та v2c є найбільш реалізованими версіями SNMP. Підтримка протоколу SNMP v3 останнім часом почала наздоганяти, оскільки він більш захищений порівняно зі старими версіями, але все ж не досяг значної частки ринку.

Таблиця 1.8

SNMP

Версії SNMP	Версії RFC	Опис
SNMP v1	1155 та 1157	Це перша версія протоколу SNMP
SNMP v2c	1901,1905,1906,2578	Це протокол, який включає в себе вдосконалення SNMPv1 в областях типів пакетів протоколів, транспортних відображень, елементів структури MIB, але використовуючи існуючу структуру адміністрування SNMPv1
SNMP v3	1905,1906,3411,3412, 3414, 3415.	визначає захищену версію SNMP. Протокол SNMPv3 також полегшує віддалену конфігурацію об'єктів SNMP

З моменту створення SNMP пройшов значні оновлення. Однак протоколи SNMP v1 та v2c є найбільш реалізованими версіями SNMP. Підтримка протоколу SNMP v3 останнім часом почала наздоганяти, оскільки він більш захищений порівняно зі старими версіями, але все ж не досяг значної частки ринку.

Таблиця 1.9

Версії SNMP

SNMP v1	Безпека на базі громади
SNMP v2c	Безпека на базі громади
SNMP v2u	Захист на основі користувачів
SNMP v2	Партія безпеки
SNMP v3	Захист на основі користувачів

1.3. Організація журналювання подій в Unix/Linux системах

Журнал роботи є основним джерелом інформації про роботу системи і її помилки. У даному розділі було розглянуто основні аспекти журналювання операційної системи, структуру каталогів, програми для читання і огляду логів.

Всі файли журналів, можна віднести до однієї з наступних категорій:

- додатки;
- події;
- служби;
- системний.

Більшість же лог файлів міститься в директорії /var/log.

/var/log/syslog або /var/log/ messages містить глобальний системний журнал, в якому пишуться повідомлення з моменту запуску системи, від ядра Linux, різних служб, виявлених пристроях, мережевих інтерфейсів і багато іншого.

/var/log/auth.log або /var/log/secure – інформація про авторизацію користувачів, включаючи вдалі і невдалі спроби входу в систему, а також задіяні механізми аутентифікації.

/var/log/dmesg – драйвера пристроїв. Однойменною командою можна переглянути висновок вмісту файлу. Розмір журналу обмежений, коли файл досягне своєї межі, старі повідомлення будуть перезаписані новішими. Задавши ключ --level можна відфільтрувати висновок за критерієм значимості.

Одним з найбільш важливих аспектів в управлінні будь-якою системою – контроль за системними подіями. Linux пропонує незвичайний метод журналювання, а також дозволяє конфігурувати складові частини журналів. У Linux журнали являють собою звичайний текст, так що ми можемо досліджувати і читати їх, не застосовуючи спеціальних засобів. Також можна написати скрипт для перегляду журналів та автоматичного виконання будь-яких функцій на основі їх вмісту. Linux-журнали зберігаються в каталозі /var/log. Тут міститься кілька файлів, які підтримуються системою. Інші сервіси та програми також

можуть розміщувати тут свої log-файли. Більшість журналів доступні для читання тільки суперкористувачеві root, але це можна легко змінити, скорегувавши права доступу до файлів.

/var/log/alternatives.log – висновок програми update-alternatives, в якому знаходяться символічні посилання на команди або бібліотеки за замовчуванням.

/var/log/anaconda.log – записи, зареєстровані під час установки системи.

/ var / log / audit – записи, створені службою аудиту auditd.

/var/log/boot.log – інформація, яка пишеться при завантаженні операційної системи.

/var/log/cron – звіт служби про виконувані команди і повідомлення від самих команд.

/var/log/cups – ввсе, що пов'язано з печаткою і принтерами.

/var/log/faillog – невдалі спроби входу в систему. Дуже корисно при перевірці загроз в системі безпеки, хакерських атаках, спроб злому методом перебору. Прочитати вміст можна за допомогою команди faillog.

var/log/kern.log – журнал містить повідомлення від ядра і попередження, які можуть бути корисні при усуненні помилок користувальницьких модулів вбудованих в ядро.

/var/log/maillog/ або /var/log/mail.log – журнал поштового сервера, використовуваного на ОС.

/var/log/pm-powersave.log – повідомлення служби економії заряду батареї.

/var/log/samba/ - список файлового сервера Samba, який використовується для доступу до загальних папок Windows і надання доступу користувачам Windows до загальних папок Linux.

/var/log/Xorg.0.log – список X сервера. Найчастіше марні, але якщо в них є рядки починаються з EE, то слід звернути на них увагу.

Для кожного дистрибутива буде окремий журнал менеджера пакетів.

/var/log/yum.log – для програм встановлених за допомогою Yumв RedHat Linux.

/var/log/dpkg.log – для програм встановлених за допомогою dpkgв Debian Linux і всім у сімействі родинних дистрибутивів.

І трохи бінарних журналів обліку сесії користувачів.

/var/log/lastlog – остання сесія користувачів. Прочитати можна командою last.

/var/log/tallylog – аудит невдалих спроб входу в систему.

/var/log/btmp – журнал запису невдалих спроб входу в систему.

/var/log/utmp – список входів користувачів в систему на даний момент.

/var/log/wtmp – ще один журнал запису входу користувачів в систему.

Журнал повідомлень – основний системний log-файл. Він містить повідомлення про хід завантаження системи, а також інші повідомлення про статус працюючої системи. У цьому файлі накопичуються повідомлення про помилки введення/виводу (IO), проблеми з мережею та інші повідомлення про системні помилки. Крім того, тут зберігається і інша інформація, наприклад, про те, в який час який-небудь користувач став адміністратором. Якщо запущено будь-який сервіс, наприклад, DHCP-сервер, в цьому файлі ви можете спостерігати за його діяльністю. З вивчення файлу /var/log/messages зазвичай починають виявлення і усунення неполадок.

/var/log/XFree86.0.log – цей журнал містить інформацію про останній запуск сервера Xwindow Xfree86. У разі виникнення проблем з запуском графічної сесії в цьому файлі зазвичай можна знайти причину невдачі. інші журнали У каталозі /var/log можуть бути і інші log-файли в залежності від того, який дистрибутив Linux використовується і які сервіси та додатки запущені. Наприклад, вони можуть бути пов'язані з запуском поштового сервера, розподілом ресурсів, автоматичним виконанням завдань і т.д.

У каталозі /var/log можна побачити кілька файлів, імена яких закінчуються цифрою. Це «rotated» архіви. Log-файли можуть бути досить великими і громіздкими. У Linux є команда для ротації цих файлів, за рахунок чого поточна інформація не змішується з застарілими, вже не потрібними даними. Команда logrotate зазвичай запускається автоматично через певні проміжки часу, але її

можна запустити і вручну. Команда logrotate бере поточну версію log-файлу і додає в кінець імені файлу ".1". Потім всі попередні такі файли нумеруються послідовно: ".2," ".3," і так далі. Чим більше число в кінці імені файлу, тим файл старше. Є можливість настроїти автоматичний режим роботи logrotate, відредагувавши файл /etc/logrotate.conf. Дізнатися більше про команду logrotate можна за допомогою man logrotate.

Для роботи з log-файлами можна використовувати будь-який інструмент роботи з текстом.

Для огляду журналу завантаження з метою перегляду повідомлень про останнє завантаження системи, необхідно скористатися параметром dmesg. Команда, буде здійснювати посторінковий вивід повідомлень про хід завантаження.

Часом виникає необхідність стежити за log-файлом в ході виконання якихось дій. Команда tail, яка дає змогу відтворювати події останніх кількох рядків текстового файлу. Якщо додати опцію -f, tail продовжить показувати нові рядки файлу в міру їх формування.

tail -f /var/log/messages – команда, представлена вище, покаже останні десять рядків файлу /var/log/messages, потім продовжить стежити за файлом і буде видавати повідомлення про нові дії.

Для управління журналюванням існує два сервіси, або демона, klogd і syslogd. klogd має справу тільки з повідомленнями ядра. syslogd має справу з іншими системними повідомленнями, наприклад, що стосуються додатків. Можливо конфігурувати обидва сервіси, відредагувавши файли /etc/syslog.conf і /etc/sysconfig/syslog. Кожне повідомлення, яке видається програмним забезпеченням, надає якусь інформацію для визначення, звідки воно надійшло і в чому його суть. Файл /etc/syslog.conf дозволяє встановити, що потрібно робити з повідомленнями певного типу. Віддалений журнал – чудовий засіб забезпечення безпеки. Розміщуючи журнали на віддаленій системі, можна таким чином закрити пролом в системі безпеки, яка полягає в тому, що зловмисник може видалити сліди своєї діяльності в log-файлах.

1.4. Організація журналювання подій в Windows системах

Журнали подій – це спеціальні файли, в які система і додатки записують важливі для комп'ютера події, такі як події входу користувачів в систему або помилки, що виникають при роботі додатків. Коли виникають подібні типи подій, система Windows створює записи в журналах подій. У детальних описах подій користувачі можуть знайти інформацію, корисну для усунення несправностей, виявлення причин проблем з системою, додатками, обладнанням комп'ютера.

Журнали подій Windows – це не текстові файли (не як UNIX syslog) і їх не можна переглядати і досліджувати простими текстовими редакторами. Журнали подій Windows це бінарні файли спеціального формату, схожі на файли баз даних, що складаються зі спеціальних записів – подій Windows.

Microsoft Windows запускає спеціальну службу (сервіс) – Журнал подій Windows, для обслуговування задач, пов'язаних з журналами подій – управління журналами подій, записом нових подій в журнали, обслуговуванням запитів на читання даних з журналів і т.п. Служба Журнал подій Windows надає спеціальний API, який дозволяє додаткам працювати з журналами подій.

Реєстрація подій як стандартний системний механізм забезпечення безпеки та відмовостійкості з'явилася у версії Microsoft Windows NT 3.1 в 1993 році. Починаючи з цієї версії в будь-який Windows присутні 3 основних журнали – журнал Додатки, журнал Система і журнал Безпека. У сучасних версіях Windows і Windows Server число журналів може перевищувати сотню, так як тепер і додатки можуть створювати свої власні журнали, інтегровані в систему реєстрації подій Windows.

Переглядати і досліджувати події Windows можна стандартним додатком Перегляд подій або спеціальними програмами, що поставляються незалежними розробниками.

Служба (сервіс) «Журнал подій Windows» – це спеціальна служба (сервіс) для управління подіями і журналами подій. Вона підтримує виконання операцій

запису нових подій, читання подій додатками, підписки на подій, резервного копіювання та архівування журналів подій і т.п. За замовчуванням, після установки системи Windows служба «Журнал подій Windows» включена і запускається автоматично. Не варто зупиняти або вимикати цю службу. Зупинка служби «Журнал подій Windows» може погіршити стабільність і безпеку системи.

Журнали подій Windows зберігаються в спеціальних файлах системного каталогу Windows. Служба (сервіс) «Журнал подій Windows» дозволяє користувачам зберігати журнали і робити резервні копії журналів в файли. Windows NT, 2000 і XP/Server 2003 зберігають журнали подій в файлах EVT формату. Windows Vista/Server 2008 і більш нові системи зберігають журнали подій в EVTX форматі. Аналіз файлів журналів подій і їх резервних копія є основою розслідування різних інцидентів.

Робочі версії файлів журналів подій Windows зазвичай заблоковані системою, точніше службою «Журнал подій Windows» і їх неможливо безпосередньо відкрити на працюючій системі. Але якщо комп'ютер буде завантажений іншою системою з іншого диска, то робочі файли журналів системи можна відкрити або скопіювати. Також їх можна скопіювати або відкрити, якщо підключити системний диск до іншого комп'ютера. За замовчуванням, файли журналів подій зберігаються в каталозі «C:\Windows\System32\winevt\Logs\».

Відкрити файл журналу подій в додатку Перегляд подій Windows можна виконавши наступні кроки:

Відкрийте програму Перегляд подій.

Натисніть «Відкрити збережений журнал» в панелі «Дії», розташованої в правій частині вікна.

Знайдіть і виберіть потрібний файл в списку файлів, натисніть кнопку «Відкрити» і його вміст відобразитися в області перегляду подій в додатку.

Журнал подій Додатків містить події, згенеровані додатками, а не системою. Наприклад, сервер бази даних може записувати помилки, що виникають при

його роботі в журнал додатків. Розробники програм самі вирішують які події має сенс протоколювати в журналі подій Додатки. Наприклад, Microsoft SQL Server веде протокол детальну інформцію про важливі аварійні ситуації, що виникають при роботі SQL-сервера, таких як «недостатньо пам'яті», «збій при резервному копіюванні бази даних» і т.д. При цьому події, згенеровані різними додатками, потрапляють в єдиний журнал додатків. Додатки ідентифікуються як різні «джерела» в базовому властивості подій. Тому нескладно виділити події конкретного додатка. Також варто враховувати що ID події (код події) теж визначається додатком, згенерована подія і коди можуть дублюватися для різних джерел. Таким чином події певного типу ідентифікуються і джерелом і кодом, а не тільки кодом, як для інших журналів, наприклад для журналу Безпека.

Журнал подій Система містить події, згенеровані системними компонентами. Наприклад, відмови драйверів або інших системних компонентів при запуску системи записуються в системний журнал подій. Типи і коди подій системних компонентів зумовлені розробниками операційної системи Windows. Аналогічно журналу додатків, системний журнал містить події з різних джерел (системних компонентів) і слід враховувати що конкретні події ідентифікуються не тільки кодом, але джерелом. Журнал подій Система – важливе джерело інформації при пошуку причин відмов і допомоги системним адміністраторам та технічними фахівцями.

Журнал подій Безпека містить події, що впливають на безпеку системи. Це спроби (успішні та невдалі) входу в акаунти системи, використання ресурсів (файлів, реєстру, пристроїв), управління обліковими записами, зміни прав і привілеїв акаунтів, запуск і зупинка процесів (програм) і т.д. Адміністратор може налаштувати які категорії подій необхідно реєструвати. Наприклад, за замовчуванням система налаштована реєструвати події управління обліковими записами, події входу в систему, а аудит доступу до об'єктів не включений. Варто бути обережним при налаштуванні аудиту доступу до файлів, так як це може привести до появи великої кількості подій, що в свою чергу може негативно

відобразитися на загальній продуктивності системи і швидкому переповненні журналу безпеки.

Запис в журнал безпеки проводиться лише системними компонентами, коди подій однозначно ідентифікують події. Журнал подій Безпека є важливим джерелом інформації при розслідуванні інцидентів порушення безпеки і його аналіз актуальний для адміністраторів безпеки, фахівців з інформаційної безпеки і фахівців по цифровій криміналістичної експертизи.

Призначена для користувача робоча станція – найвразливіше місце інфраструктури по частині інформаційної безпеки. Користувачам може прийти на робочу пошту лист начебто з безпечного джерела, але з посиланням на заражений сайт. Так можна придумати не один десяток кейсів, як через користувачів шкідливе ПО може потрапити на внутрішньокорпоративні ресурси. Тому робочі станції вимагають підвищеної уваги.

Для виявлення атаки на самій ранній стадії в ОС Windows є три корисних подієвих джерела: журнал подій безпеки, журнал системного моніторингу та журнали Power Shell (рис.1.6).

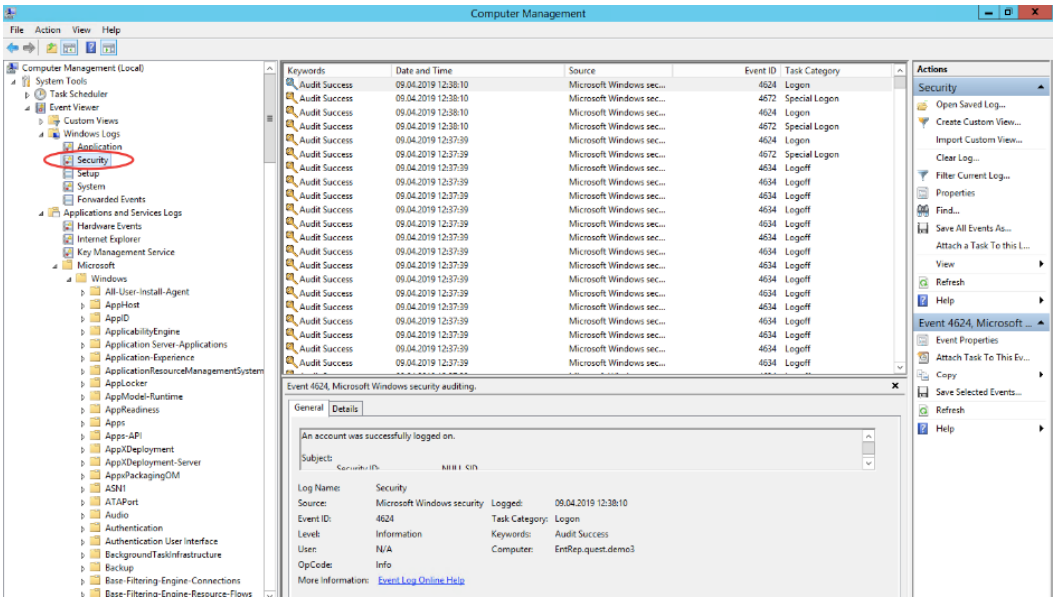


Рис. 1.6. Журналювання Windows

Журнал подій безпеки (Security Log)

Це головне місце зберігання системних логів безпеки. Сюди зберігаються події входу/виходу користувачів, доступу до об'єктів, зміни політик і інших активностей, пов'язаних з безпекою (див. рис.1.7).

Перебір користувачів і груп (події 4798 і 4799). Шкідливе ПО на самому початку атаки часто перебирає локальні облікові записи користувачів і локальні групи на робочої станції, щоб знайти облікові дані для своїх темних справ. Ці події допоможуть виявити шкідливий код раніше, ніж він рушить далі і, використовуючи зібрані дані, поширюється на інші системи.

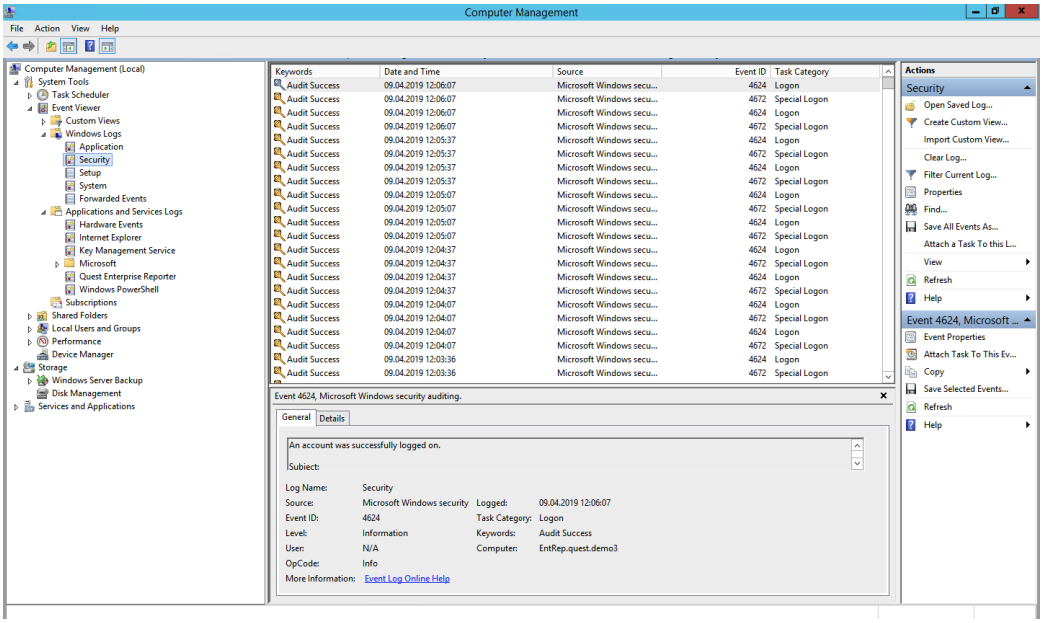


Рис. 1.7. Журнал подій безпеки

Створення локальної облікового запису і зміни в локальних групах (події 4720, 4722-4726, 4738, 4740, 4767, 4780, 4781, 4794, 5376 і 5377). Атака може також починатися, наприклад, з додавання нового користувача в групу локальних адміністраторів.

Спроби входу з локальної обліковим записом (подія 4624). Добропорядні користувачі заходять з доменної обліковим записом і виявлення входу під локальної обліковим записом може означати початок атаки. Подія 4624 включає також входи під доменним обліковим записом, тому при обробці подій потрібно фільтрувати події, в яких домен відрізняється від імені робочої станції.

Спроба входу із заданою обліковим записом (подія 4648). Таке буває, коли процес виконується в режимі «Запуск від імені» (run as). У нормальному режимі

роботи систем такого не повинно бути, тому такі події повинні перебувати під контролем.

Блокування/розблокування робочої станції (події 4800-4803). До категорії підозрілих подій можна віднести будь-які дії, які відбувалися на заблокованій робочої станції.

Зміни конфігурації файєрвола (події 4944-4958). Очевидно, що при установці нового ПО настройки конфігурації файєрвола можуть змінюватися, що викличе помилкові спрацьовування. Контролювати такі зміни в більшості випадків немає необхідності, але знати про них точно зайвим не буде.

Підключення пристроїв Plug'n'play (подія 6416 і тільки для Windows 10). За цим важливо стежити, якщо користувачі зазвичай не підключають нові пристрої до робочої станції, і раптово відбулось підключення.

Windows включає в себе 9 категорій аудиту та 50 субкатегорій для тонкої настройки. Мінімальний набір субкатегорій, який варто включити в настройках:

- Logon / Logoff
- Logon;
- Logoff;
- Account Lockout;
- Other Logon / Logoff Events.
- Account Management
- User Account Management;
- Security Group Management.
- Policy Change
- Audit Policy Change;
- Authentication Policy Change;
- Authorization Policy Change.

Системний моніторинг (Sysmon) – вбудована в Windows утиліта, яка вміє записувати події в системний журнал. Звичайно потрібно його встановлювати окремо.

Ці ж події можна знайти в журналі безпеки (включивши потрібну політику аудиту), але Sysmon дає більше подробиць. Які події можна забирати з Sysmon?

Створення процесу (ID події 1). Системний журнал подій безпеки теж може сказати, коли запустився який-небудь *.exe і навіть покаже його ім'я і шлях запуску. Але на відміну від Sysmon не зможе показати хеш додатки. Зловмисне ПО може називатися навіть нешкідливим notepad.exe, але саме хеш виявить небезпеку та вразливість.

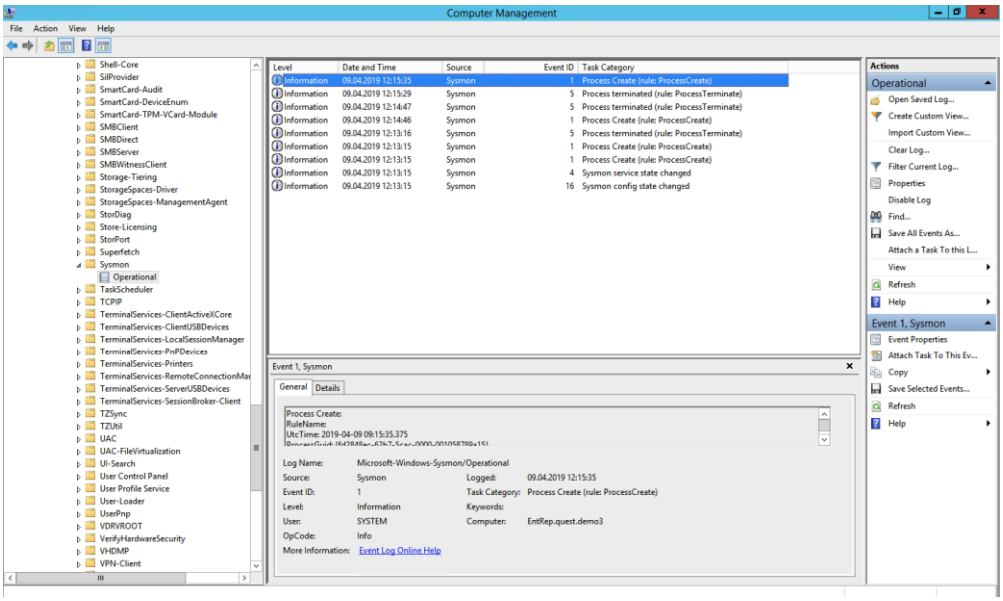


Рис. 1.8. Windows утиліта Sysmon

Сумісність із мережею (ID події 3). Очевидно, що здійснювати підключення до мережі багато, і за всіма не встежити. Але важливо враховувати, що Sysmon на відміну від того ж Security Log вміє прив'язати мережеве підключення до полів ProcessID і ProcessGUID, показує порт і IP-адреси джерела і приймача.

Зміни в системному реєстрі (ID події 12-14). Найпростіший спосіб додати себе в автозапуск – прописатися в реєстрі. Security Log це вміє, але Sysmon показує, хто вніс зміни, коли, звідки, process ID і попереднє значення ключа.

Створення файлу (ID події 11). Sysmon, на відміну від Security Log, покаже не тільки розташування файлу, але і його ім'я. Зрозуміло, що за всім не встежиш, але можна ж проводити аудит певних директорій.

Переваги Sysmon:

1. Зміна часу створення файлу (ID події 2). Деякий шкідливе ПО може підміняти дату створення файлу для його приховування зі звітів з недавно створеними файлами.

2. Завантаження драйверів і динамічних бібліотек (ID подій 6-7). Відстеження завантаження в пам'ять DLL і драйверів пристроїв, перевірка цифрового підпису та її валідності.

3. Створення потоку в виконувати процесі (ID події 8). Один з видів атаки, за яким теж потрібно стежити.

4. Події RawAccessRead (ID події 9). Операції читання з диска за допомогою "\\. \". В абсолютній більшості випадків така активність повинна вважатися ненормальною.

5. Створення іменованого файлового потоку (ID події 15). Подія реєструється, коли створюється іменований файловий потік, який генерує події з хешем вмісту файлу.

6. Створення named pipe і підключення (ID події 17-18). Відстеження шкідливого коду, який комунікує з іншими компонентами через named pipe.

7. Активність по WMI (ID події 19). Реєстрація подій, які генеруються при зверненні до системи по протоколу WMI.

8. Для захисту самого Sysmon потрібно відстежувати події з ID 4 (зупинка і запуск Sysmon) і ID 16 (зміна конфігурації Sysmon).

Журнали Power Shell

Power Shell – потужний інструмент управління Windows-інфраструктурою, тому великі шанси, що атакуючий вибере саме його. Для отримання даних про події Power Shell можна використовувати два джерела: Windows PowerShell log і Microsoft-WindowsPowerShell / Operational log.

Windows PowerShell log

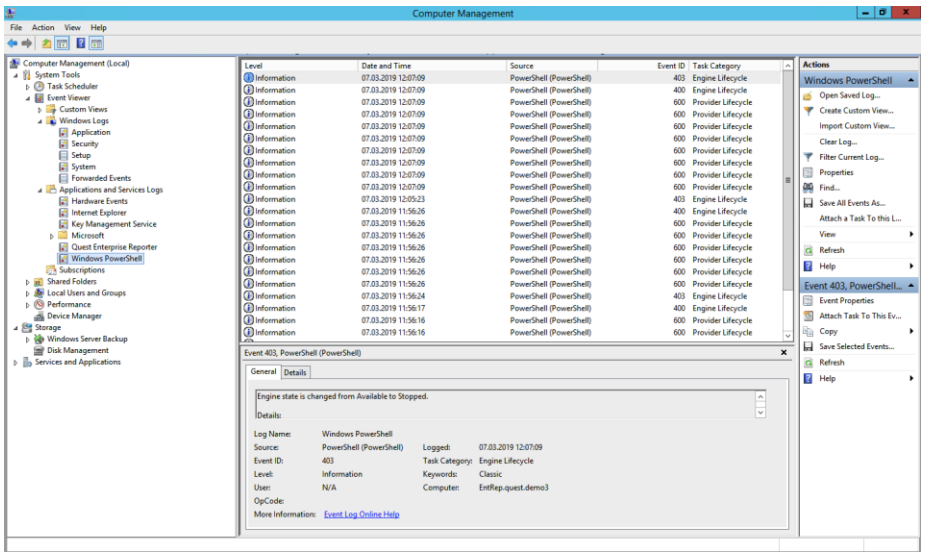


Рис. 1.9. Журнали Power Shell

Завантажений постачальник даних (ID події 600). Постачальники PowerShell – це програми, які служать джерелом даних для PowerShell для перегляду і управління ними. Наприклад, вбудованими постачальниками можуть бути змінні середовища Windows або системний реєстр. За появою нових постачальників потрібно стежити, щоб вчасно виявити зловмисну активність. Наприклад, якщо бачите, що серед постачальників з'явився WSMan, значить було розпочато віддалений сеанс PowerShell (рис.1.10).

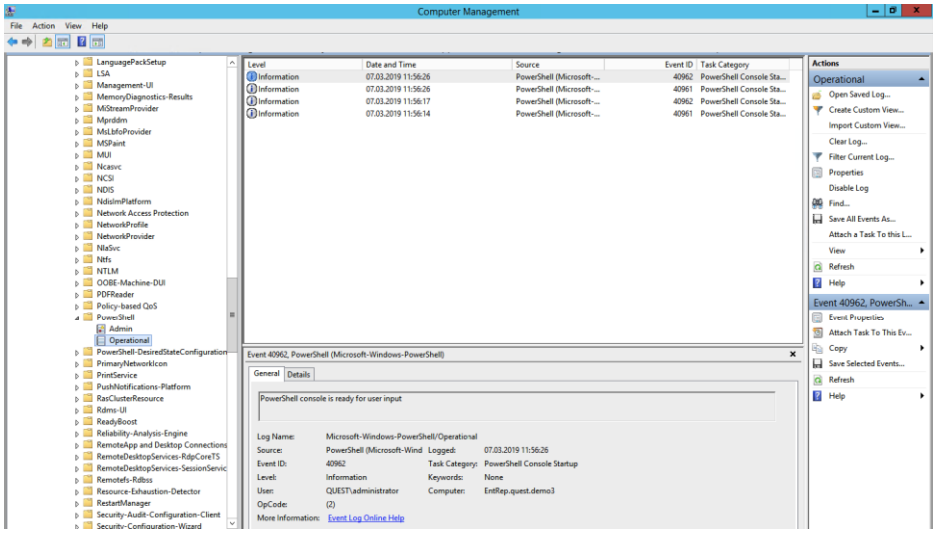


Рис. 1.10. Microsoft-WindowsPowerShell/Operational log

Журнал роботи модулів (ID події 4103). У подіях зберігається інформація про кожну виконану команді і параметрах, з якими вона викликалася.

Журнал блокування скриптів (ID події 4104). Журнал блокування скриптів показує кожен виконаний блок коду PowerShell. Навіть якщо зловмисник

спробує приховати команду, цей тип події покаже фактично виконану команду PowerShell. Ще в цьому типі події можуть фіксуватися деякі виконувані низькорівневі виклики API, ці події зазвичай записується як Verbose, але якщо підозріла команда або сценарій використовуються в блоці коду, він буде зареєстрований як с критичністю Warning.

Також необхідно звертати увагу, що після настройки інструменту збору і аналізу цих подій знадобиться додатковий час на налагодження для зниження кількості помилкових спрацьовувань.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТОПОЛОГІЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ ТА РОЗРОБКА СИСТЕМИ АДРЕСАЦІЇ

2.1. Розробка топології комп'ютерної мережі

При проектуванні мережі була використана трирівнева структура, що включає в себе:

- рівень ядра(базовий рівень);
- рівень розподілу;
- рівень доступу.

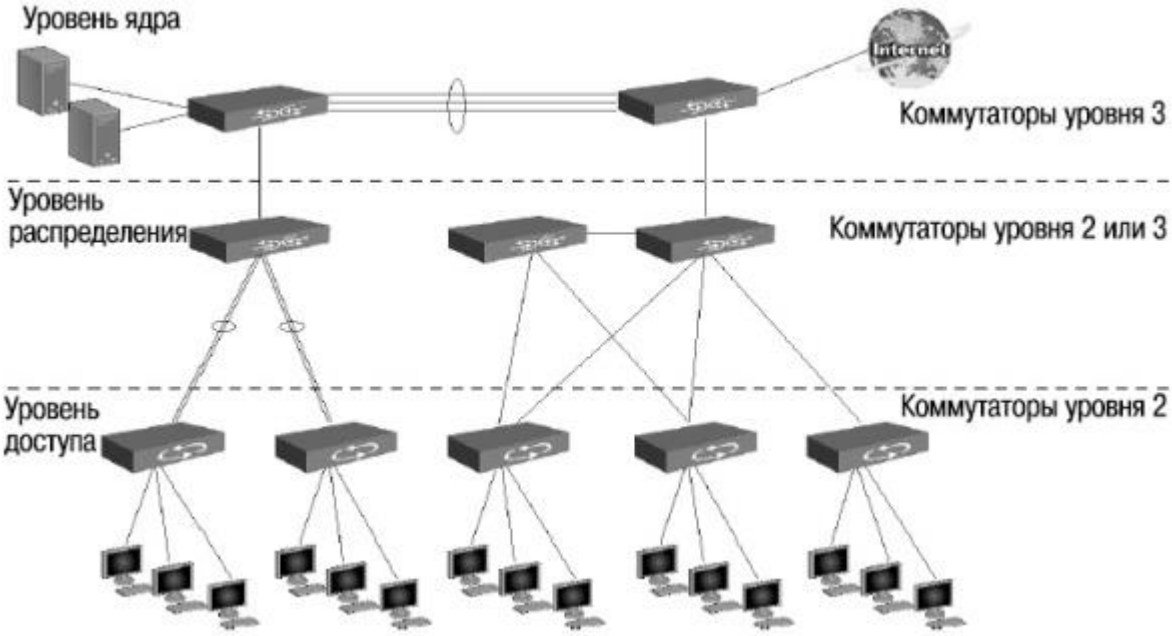


Рис.2.1. Ієрархічний проект мережі

Рівень ядра складається з 1-го маршрутизатора.

Рівень розподілу представляє собою набір з 2 маршрутизаторів, що під'єднані до ядра мережі. З'єднання з ядром мережі виконано за допомогою послідовних двоточкових каналів зв'язку.

Рівень доступу представлений кінцевим устаткуванням користувачів та комутаційним обладнанням, що забезпечує функціонування рівня.

Особливістю з'єднань комутаційного обладнання є те, що для забезпечення високої продуктивності роботи мережі використовуються лінії зв'язку

FastEthernet. Кінцеві вузли під'єднанні до комутаційного обладнання за допомогою ліній Ethernet на витій парі.

В загальному у мережі використано 2 маршрутизатори, 1 сервер та 6 комутаторів на 24 порта. Мережа складається з чотирьох підмереж.

Розглянемо детальніше кожну з під мереж.

Підмережа А має у своєму складі 1 маршрутизатор: R_1_1 і утворює локальну мережу LAN-A (рис. 2.2).

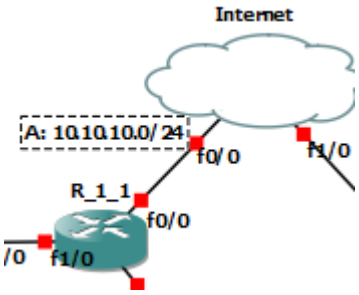


Рис.2.2. Підмережа А

У підмережі А до Internet підключений 1 маршрутизатор. З'єднання у даній підмережі виконано за допомогою витої пари технології FastEthernet.

Підмережа В має у своєму складі 1 маршрутизатор: R_1_2 і утворює локальну мережу LAN-B (рис. 2.3).

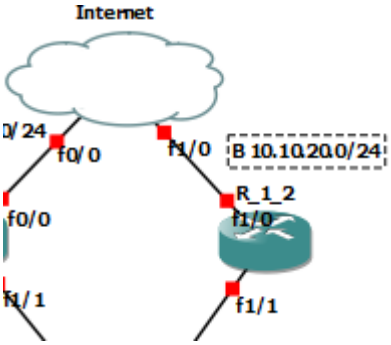


Рис.2.3. Підмережа В

У підмережі В до Internet підключено 1 маршрутизатор. З'єднання у даній підмережі виконано за допомогою витої пари технології FastEthernet.

Підмережа С має у своєму складі 1 маршрутизатор: R_1_1 і утворює локальну мережу LAN-C (рис. 2.4).

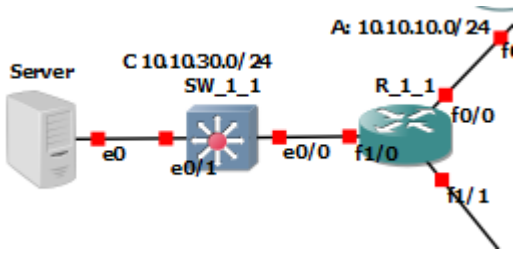


Рис.2.4. Підмережа C

У під мережі C до комутатора підключено один Server. Також у SW_1_1 один порт підключений до R_1_1. Звідси маємо що у комутаторі SW_1_1 задіяні не всі порти, що дає можливість розширення мережі. З'єднання у даній підмережі виконані за допомогою витої пари технології Ethernet.

Підмережа D має у своєму складі 5 комутаторів та 6 робочих станцій, що утворюють локальну мережу LAN-D (рис. 2.5).

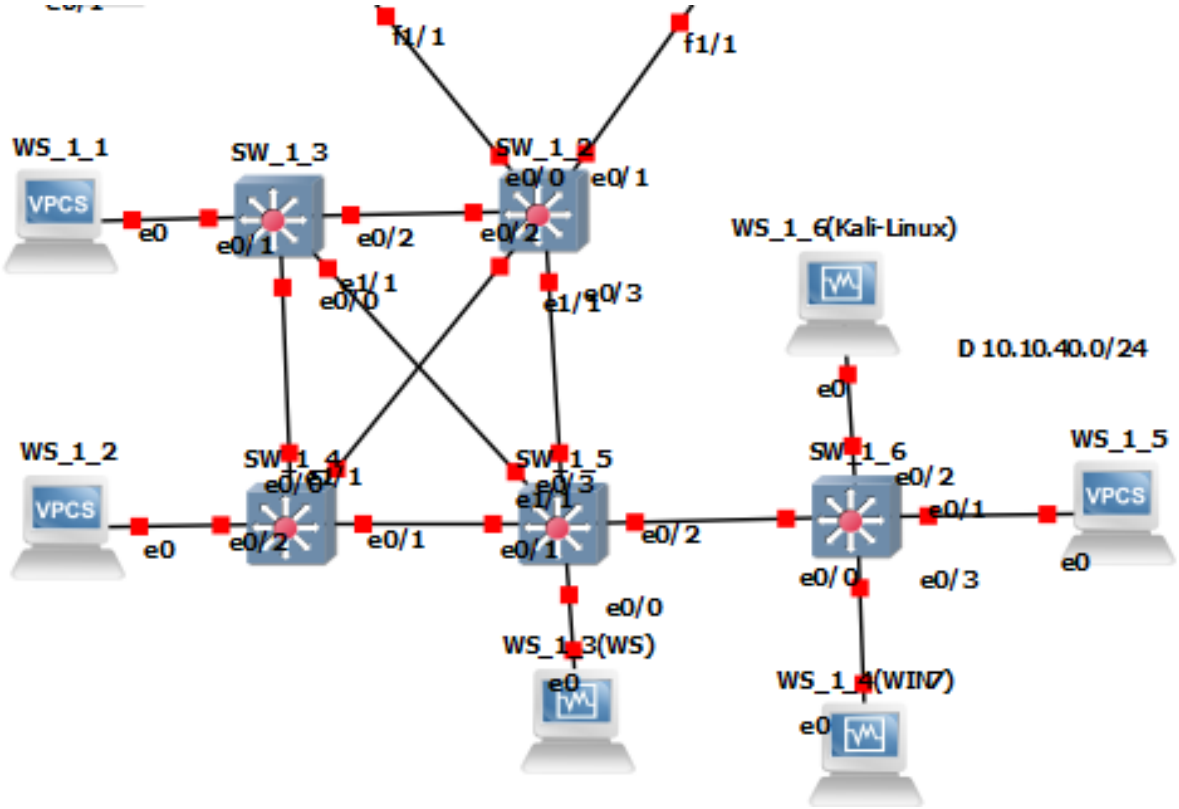


Рис. 2.5. Підмережа D

У підмережі D до комутатора підключено 4 комутатори та 6 робочих станцій. Також у SW_1_2 один порт підключений до R_1_1 та один порт до R_1_2 . Звідси маємо що у комутаторі SW_1_1 задіяні не всі порти, що дає

можливість розширення мережі. З'єднання у даній підмережі виконані за допомогою витої пари технології Ethernet.

2.2. Розробка схеми адресації мережі

Для рівня ядра мережі розподіл адрес по діапазону проводився таким чином, щоб використати мінімальну кількість IP-адрес, забезпечивши при цьому потреби підключених вузлів. Розрахунок приведено нижче:

Розрахунок IP адрес для підмереж наведений у таблицях 2.1-2.4.

Таблиця 2.1

Адресація підмережі A

Призначення	IP-адреса
Адреса мережі	10.10.10.0
Маска мережі	255.255.255.0 /24
Мінімальна адреса	10.10.10.1
Максимальна адреса	10.10.10.254
Широкомовна адреса	10.10.10.255
Шлюз по замовчуванню	10.10.10.1
Діапазон адрес маршрутизаторів	10.10.10.1-10.10.10.3
Діапазон адрес майбутніх ПК	10.10.10.4-10.10.10.7
Діапазон вільних адрес	10.10.10.8-10.10.10.254

Таблиця 2.2

Адресація підмережі B

Призначення	IP-адреса
Адреса мережі	10.10.20.0
Маска мережі	255.255.255.0 /24
Мінімальна адреса	10.10.20.1
Максимальна адреса	10.10.20.254
Широкомовна адреса	10.10.20.255
Шлюз по замовчуванню	10.10.20.1

Продовження таблиці 2.2

Призначення	IP-адреса
Діапазон адрес маршрутизаторів	10.10.20.1 - 10.10.20.3
Діапазон адрес майбутніх ПК	10.10.20.4 - 10.10.20.9

Таблиця 2.3

Адресація підмережі C

Призначення	IP-адреса
Адреса мережі	10.10.30.0
Маска мережі	255.255.255.0 /24
Мінімальна адреса	10.10.30.1
Максимальна адреса	10.10.30.254
Широкомовна адреса	10.10.30.255
Шлюз по замовчуванню	10.10.30.1
Діапазон адрес комутаторів	10.10.30.2 – 10.10.30.3
Діапазон адрес майбутніх ПК	10.10.30.4 – 10.10.30.8
Діапазон вільних адрес	10.10.30.9 -10.10.30.254
Адреса мережі	10.10.30.0

Таблиця 2.4

Адресація підмережі D

Призначення	IP-адреса
Адреса мережі	10.10.40.0
Маска мережі	255.255.255.0 /24
Мінімальна адреса	10.10.40.1
Максимальна адреса	10.10.40.254
Широкомовна адреса	10.10.40.255
Шлюз по замовчуванню	10.10.40.1
Діапазон адрес комутаторів	10.10.40.2 - 10.10.40.3
Діапазон адрес майбутніх ПК	10.10.40.4 - 10.10.40.7
Діапазон вільних адрес	10.10.40.8 -10.10.40.254

Для забезпечення функціонування мережі, перш за все було виконано розрахунок параметрів адресації. Під час розрахунку особлива увага була приділена економії IP-адрес, тому мережі було розбито на підмережі з оптимальними масками, згідно кількості вузлів у кожній з них (Таблиця 2.5).

Таблиця 2.5

Параметри адресації мережі

Підмережа/ Пристрій	Інтерфейс/Мережний адаптер/Шлюз	IP-адреса	Маска підмережі	Префікс
Підмережа A	—	10.10.10.0	255.255.255.0	/24
Підмережа B	—	10.10.20.0	255.255.255.0	/24
Підмережа C	—	10.10.30.0	255.255.255.0	/24
Підмережа D	—	10.10.40.0	255.255.255.0	/24
Wan	—	8.8.8.8	255.255.255.255	/32
Маршрутизатор R_1_1	Fa0/0	10.10.10.2	255.255.255.0	/24
	Fa1/1	10.10.40.3	255.255.255.0	/24
	Fa1/0	10.10.30.1	255.255.255.0	/24
Маршрутизатор R_1_2	Fa1/0	10.10.20.2	255.255.255.0	/24
	Fa1/1	10.10.40.2	255.255.255.0	/24
Комутатор SW_1_2	Інтерфейс Vlan1	10.10.40.1	255.255.255.0	/24
Server	Eth0	10.10.30.2	255.255.255.0	/24
Робоча станція WS_1_1	Eth0	10.10.40.6	255.255.255.0	/24
Робоча станція WS_1_2	Eth0	10.10.40.7	255.255.255.0	/24
Робоча станція WS_1_3(WS)	Eth0	10.10.40.5	255.255.255.0	/24

Продовження таблиці 2.5

Підмережа/ Пристрій	Інтерфейс/Мережний адаптер/Шлюз	IP-адреса	Маска підмережі	Префікс
Робоча станція WS_1_4(Win7)	Eth0	10.10.40.10	255.255.255.0	/24
Робоча станція WS_1_5	Eth0	10.10.40.9	255.255.255.0	/24
Робоча станція WS_1_6(Kali-Linux)	Eth0	10.10.40.8	255.255.255.0	/24

2.3. Проект підсистеми журналювання подій

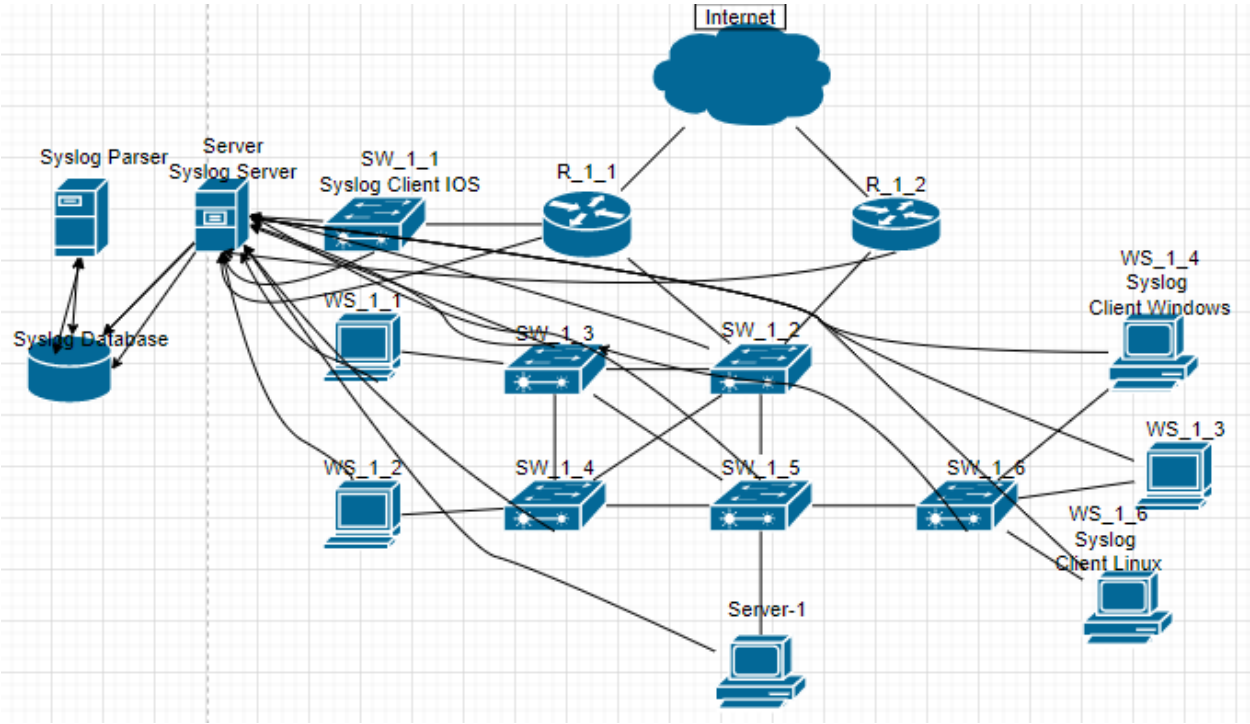


Рис. 2.6. Схема мережі журналювання подій

На рис. 2.6 зображена робота мережі журналюання подій в якій постійно йде процес запису інформації про події що відбуваються з пристроями. Реєстрація здійснюється на сервері в системному журналі. Особливістю є те що він записує все в хронологічному порядку. І зберігає дані в базу даних.

РОЗДІЛ 3. АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНОЇ МЕРЕЖІ НАЛАГОДЖЕННЯ ОБЛАДНЕННЯ

3.1. Вибір обладнання для організації роботи мережі

При виборі моделі маршрутизатора потрібно враховувати як він «маршрутизує» пакети в складній мережі з різними видами топологій. Маршрутизатор повинен містити достатню кількість портів, забезпечувати легке розширення мережі, забезпечувати необхідну швидкість передачі даних та швидкість їх обробки.

На рівні розподілу нам потрібний маршрутизатор який би підтримував технологію FastEthernet та Serial . Для цього чудово підійде маршрутизатор Cisco 7200 (рис. 3.1). Це найбільш оптимальний варіант для великої мережі підприємства чи офісу.



Рис. 3.1. Маршрутизатор Cisco 7200

Серія Cisco 7200 – обладнання провайдерського класу для розрулювання інтернет-трафіку зі швидкістю до 2 мільйонів пакетів в секунду.

Основні переваги: величезна швидкість маршрутизації (NPE-G2 Network Processing Engine), апаратна підтримка VPN-шифрування (VPN Services Adapter), великий набір різних варіантів конфігурації, підтримка інтерфейсів від DS0 до OC-3, а також Fast Ethernet, Gigabit Ethernet, Packet over SONET, надмірність і відмовостійкість.

Основним критерієм вибору моделей комутаторів є кількість портів. Також враховується швидкість передачі цих портів. У нашій мережі використовуються всі комутатори на 24 порти. Найбільш оптимальним є комутатор фірми Cisco, а саме Cisco Systems WS-C2960-24TC-S (рис. 3.2).



Рис. 3.2. Cisco Systems WS-C2960-24TC-S

Cisco Systems WS-C2960-24TC-S – керований комутатор з 24 портами: Fast Ethernet портів і 2 комбо Gigabit SFP. Перемикачі Cisco Catalyst 2960, 2960-C і 2960-S підтримують передачу голосу, відео і даних з високими стандартами безпеки доступу. Крім того, вони забезпечують масштабоване управління відповідно до потреб бізнесу.

Особливості комутатору Cisco Catalyst 2960:

- Універсальність: підтримка передачі даних, бездротового і голосового зв'язку.

- Інтелектуальне управління: призначення пріоритетів голосовим трафікам або передачі даних з метою домовленості про доставку інформації від організацій.

- Підвищена безпека: захист важливої інформації, попередження про неавторизованийдоступ користувачів, і забезпечення безпечної роботи.

- Надійність: використання переваг стандартних методів і модулі FlexStack для підвищення надійності та швидкого знищення помилок.

- Зручність налаштування конфігурації: використання набору функцій Cisco Catalyst Kit Smart Operations і Cisco Network Assistant для спрощення налаштування, оновлення та виправлення всіх помилок.

Запорука спокою: всі комутатори серії Catalyst 2960, 2960-C і 2960-S захищені гарантією, діючою протягом усього терміну служби устаткування, і відсутність обмежень на оновлення.

Інтернет – надає можливості, настільки потужні та загальні, що його можна використовувати майже для будь-яких цілей, що залежать від інформації, і вона є доступною для кожної людини, яка підключається до однієї з її складових мереж. Він підтримує спілкування між людьми через соціальні медіа, електронну

пошту, «чати», групи новин, передачу аудіо та відео, дозволяє людям спільно працювати в багатьох різних місцях. Також підтримує доступ до цифрової інформації багатьма програмами, включаючи всесвітню павутину.

У комп'ютерних мережах DMZ (демілітаризована зона), також іноді відома як периметрова мережа або екранована підмережа – це фізична або логічна підмережа, яка відокремлює внутрішню локальну мережу (LAN) від інших ненадійних мереж – зазвичай це загальнодоступний Інтернет. Зовнішні сервери, ресурси та сервіси розміщені в DMZ. Тому вони доступні з Інтернету, але решта внутрішньої локальної мережі залишається недосяжною. Це забезпечує додатковий рівень безпеки для локальної мережі, оскільки обмежує можливість хакера безпосередньо отримувати доступ до внутрішніх серверів та даних через Інтернет.

Будь-яка послуга, що надається користувачам у загальнодоступному Інтернеті, повинна розміщуватися в мережі DMZ. Деякі з найпоширеніших з цих послуг включають веб-сервери та проксі-сервери, а також сервери для електронної пошти, системи доменних імен (DNS), протоколу передачі файлів (FTP) та передачі голосу через IP (VoIP).

Архітектура мережевих ДМЗ

Існують різні способи проектування мережі з DMZ. Два основні методи – це використання одного або двох брандмауерів, хоча більшість сучасних DMZ розроблені з двома брандмауерами. Цей базовий підхід може бути розширений для створення складних архітектур, залежно від вимог мережі.

Один брандмауер з щонайменше трьома мережевими інтерфейсами можна використовувати для створення мережевої архітектури, що містить DMZ. Зовнішня мережа формується шляхом підключення загальнодоступного Інтернету – через з'єднання провайдера Інтернет (ISP) – до брандмауера першого інтерфейсу мережі. Внутрішня мережа формується з другого мережевого інтерфейсу, а сама мережа DMZ підключена до третього мережевого інтерфейсу.

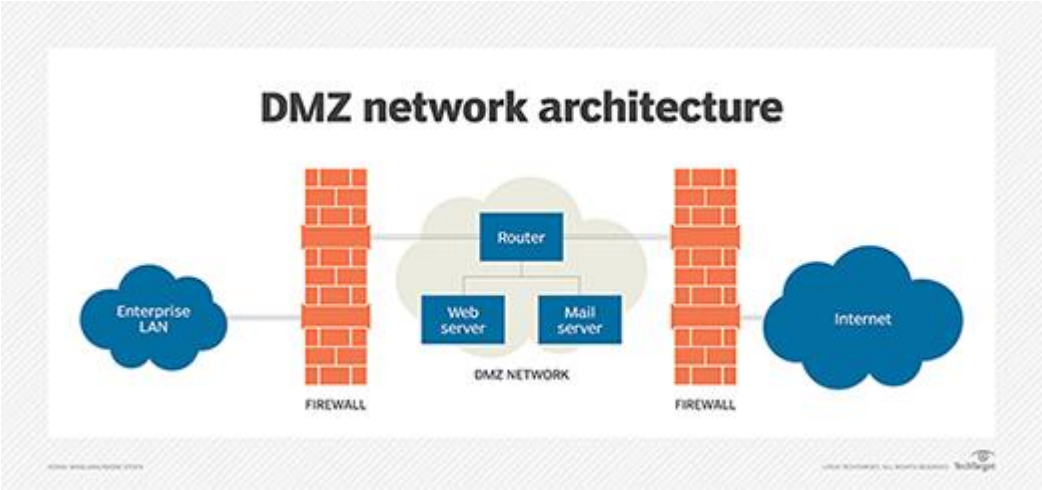


Рис. 3.3 Архітектура DMZ мережі

Різні набори правил брандмауера для моніторингу трафіку між Інтернетом та DMZ, LAN та DMZ, а також локальною мережею та Інтернетом жорстко контролюють, які порти та типи трафіку дозволені до DMZ з Інтернету, обмежують підключення до конкретних хостів у внутрішню мережу та запобігають стороннім підключенням доступ до Інтернету чи внутрішньої локальної мережі через DMZ.

Більш безпечним підходом до створення мережі DMZ є конфігурація подвійного брандмауера, в якій два брандмауери розгорнуті з розташованою між ними мережею DMZ. Перший брандмауер – також його називають периметром брандмауера – налаштований так, щоб дозволяти зовнішній трафік, призначений лише для DMZ. Другий, або внутрішній, брандмауер дозволяє лише здійснювати трафік від DMZ до внутрішньої мережі. Це вважається більш безпечним, оскільки для того, щоб зловмисник мав доступ до внутрішньої локальної мережі, потрібно буде поставити під загрозу два пристрої.

Основна перевага DMZ полягає в тому, що він пропонує користувачам із загальнодоступного Інтернету доступ до певних захищених послуг, зберігаючи при цьому буфер між цими користувачами та приватною внутрішньою мережею. Переваги безпеки цього буфера проявляються кількома способами, включаючи:

Контроль доступу для організацій. Необхідність організацій у наданні користувачам доступу до послуг, розташованих поза їх периметром мережі через загальнодоступний Інтернет, майже суттєва в сучасній організації. Мережа DMZ

забезпечує доступ до цих необхідних сервісів, одночасно вводячи рівень сегментації мережі, що збільшує кількість перешкод, які повинен перейти несанкціонований користувач, перш ніж вони зможуть отримати доступ до приватної мережі організації.

Не допускати зловмисників до проведення розвідувальної мережі. Доступний буфер, який надає DMZ, не дозволяє зловмиснику мати можливість розмахувати потенційні цілі в мережі. Навіть якщо система в DMZ порушена, приватна мережа все ще захищена внутрішнім брандмауером, що відокремлює її від DMZ. Це також ускладнює зовнішню розвідку з тієї ж причини. Незважаючи на те, що сервери в DMZ відкриті для всіх, вони мають бути та підтримуються іншим рівнем захисту. Публічне обличчя DMZ не дозволяє зловмисникам бачити вміст внутрішньої приватної мережі. Якщо зловмисникам вдасться компрометувати сервери всередині DMZ, вони все ще залишаються ізольованими від приватної мережі внутрішнім бар'єром DMZ.

Захист від підробки IP. У деяких випадках зловмисники намагаються обійти обмеження контролю доступу, підробляючи авторизовану IP-адресу, щоб представити себе іншим пристроєм у мережі. DMZ може затримувати потенційні IP-підрозділи, тоді як інша послуга в мережі перевіряє законність IP-адреси, перевіряючи, чи вона доступна.

У кожному випадку DMZ забезпечує рівень сегментації мережі, що створює простір, де можна організувати трафік, і доступ до публічних служб можна отримати на безпечній відстані від приватної мережі.

Все-таки DMZ мають своє використання. Деякі хмарні сервіси, такі як Microsoft Azure, реалізують гібридний підхід до безпеки, в якому DMZ реалізується між локальною мережею організації та віртуальною мережею. Цей гібридний підхід зазвичай застосовується в ситуаціях, коли програми організації працюють частково локально і частково у віртуальній мережі. Він також використовується в ситуаціях, коли вихідний трафік потрібно перевіряти, або коли необхідний детальний контроль трафіку між віртуальною мережею та локальним центром обробки даних.

DMZ також може бути корисним у домашній мережі, в якій комп'ютери та інші пристрої підключені до Інтернету за допомогою широкосмугового маршрутизатора та налаштовані в локальну мережу. Деякі домашні маршрутизатори містять функцію хоста DMZ, яка може протиставлятись підмережі DMZ, частіше реалізованій в організаціях, що мають набагато більше пристроїв, ніж у будинку. Функція хоста DMZ позначає один пристрій у домашній мережі, який функціонує поза брандмауером, де він виступає як DMZ, тоді як решта домашньої мережі лежить всередині брандмауера. У деяких випадках для ігрової консолі DMZ вибирається ігрова консоль, щоб брандмауер не заважав іграм. Також консоль є хорошим кандидатом для хоста DMZ, оскільки вона, ймовірно, містить менш чутливу інформацію, ніж ПК.

Окрім вибіркового використання в домашніх умовах та у хмарі, DMZ забезпечує потенційне рішення для ризиків для безпеки, спричинених зростаючою конвергенцією IT та OT (операційна технологія). Промислове обладнання, наприклад, турбінні двигуни або системи промислового управління, об'єднуються з IT-технологіями, що робить виробниче середовище більш розумним та ефективним, а також створює більшу загрозу для кібератак. Значна частина обладнання OT, що підключається до Інтернету, не розроблена для обробки атак кібербезпеки таким же чином, як і IT-пристрої.

3.2. Налаштування обладнання мережі.

Розглянемо лістинг налаштування на прикладі R_1_1 та R_1_2. У даному фрагменті лістингу було налаштовано адресацію інтерфейсів маршрутизатора та було використано протокол динамічної маршрутизації OSPF (рис. 3.3-3.4).

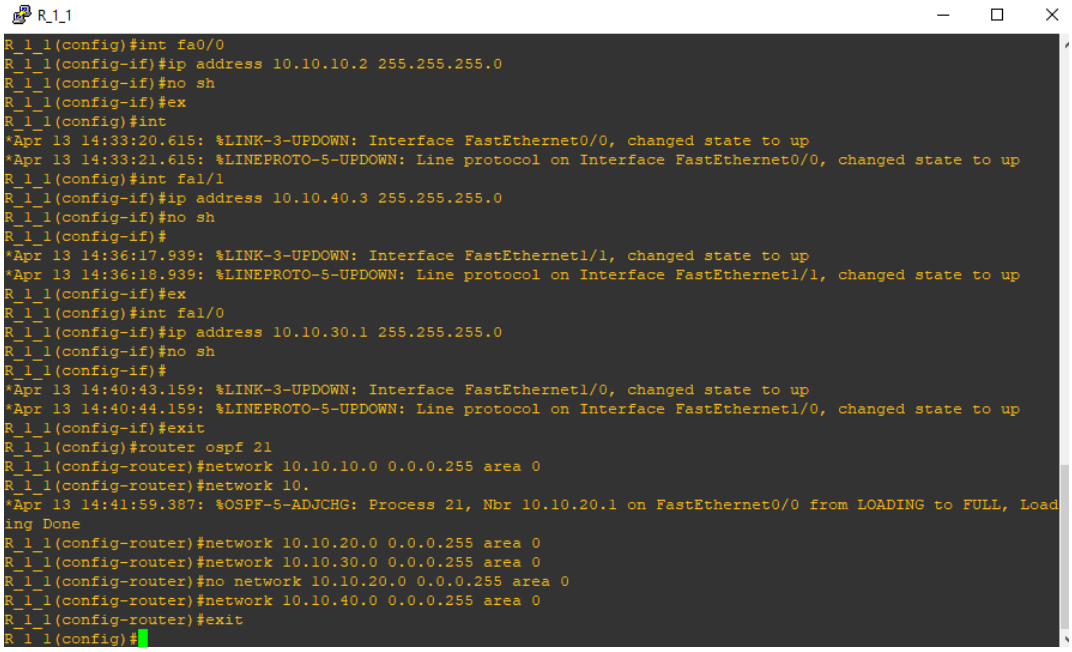


Рис. 3.4. Налаштування IP-адресації та OSPF на прикладі R_1_1

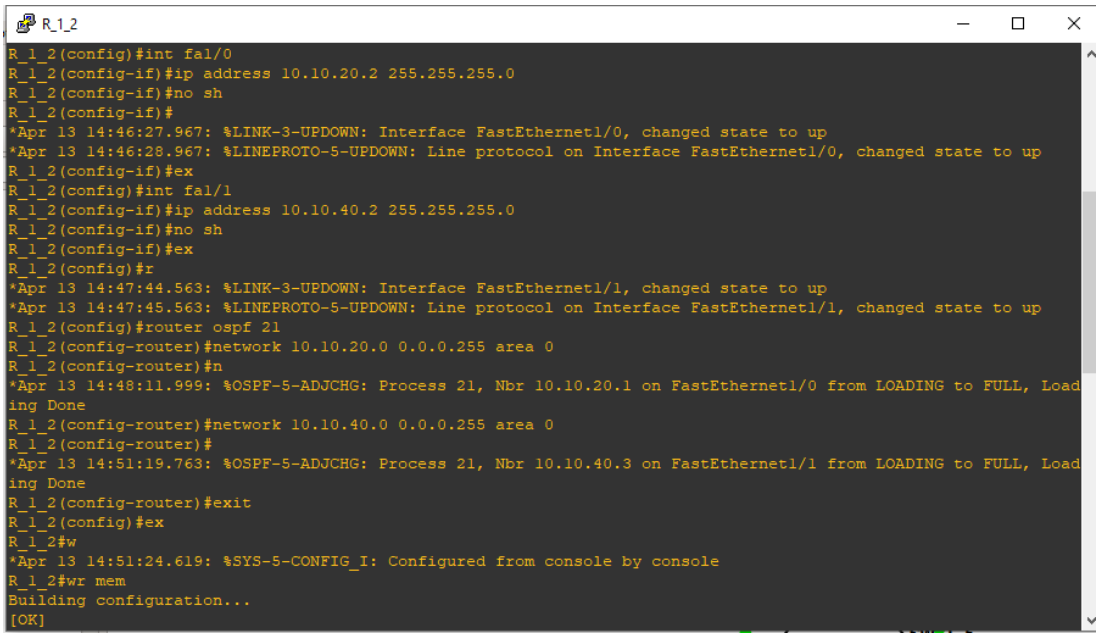


Рис. 3.5. Налаштування IP-адресації та OSPF на прикладі R_1_2

Для забезпечення захисту від втручання у мережу не довірених осіб було налагоджено механізм автентифікації (рис.3.5-3.6).

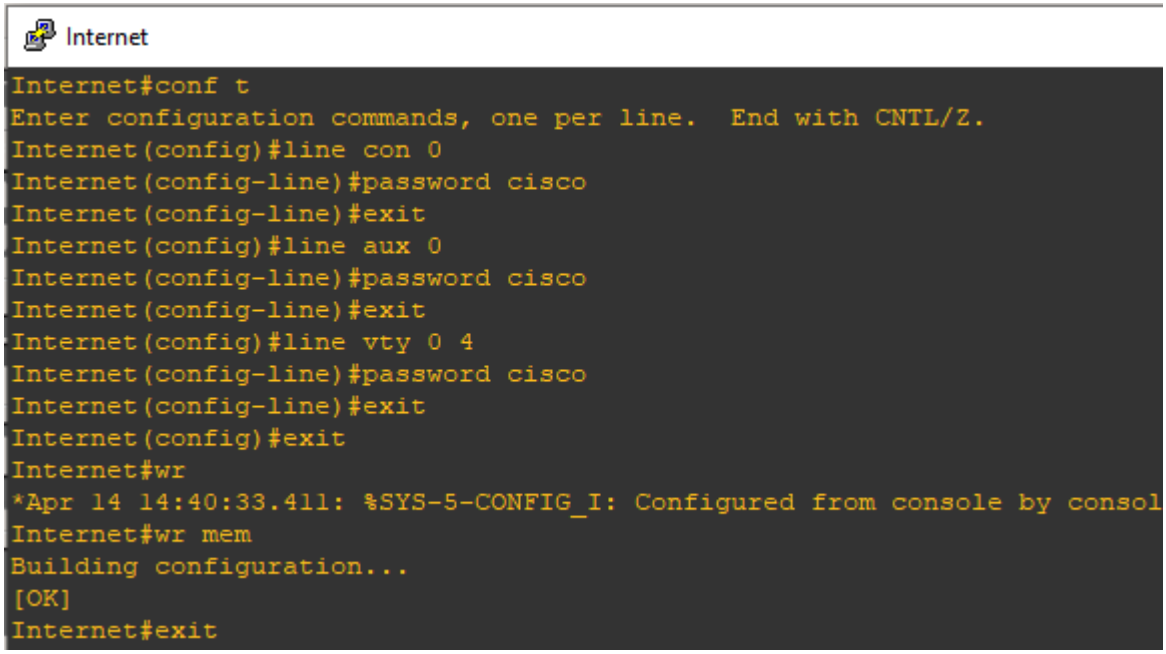


Рис. 3.6.Налаштування засобів автентифікації

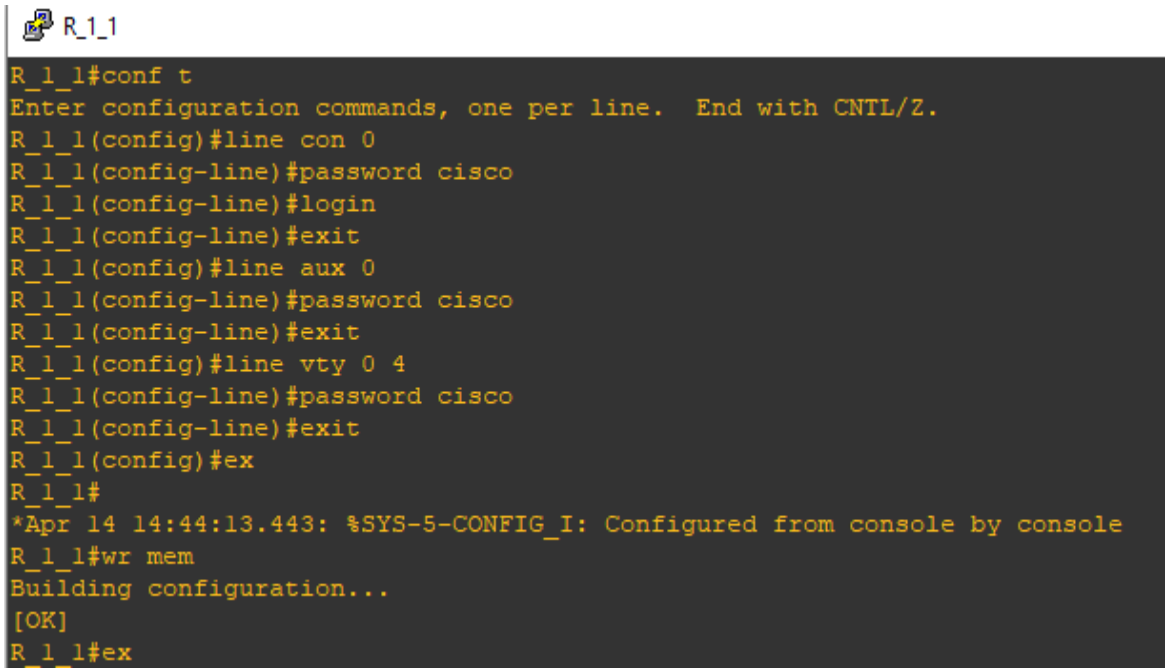


Рис. 3.7. Налаштування засобів автентифікації

Для зручності обслуговування та адміністрування на усіх маршрутизаторах мережі налаштовано засоби віддаленого доступу. Для віддаленого доступу, було обрано протокол SSHv2, через його надійність та захищеність (рис. 3.8-3.10).

R_1_1

```
Enter configuration commands, one per line.  End with CNTL/Z.
R_1_1(config)#username admin password cisco
R_1_1(config)#ip domain-name ARP.NET
R_1_1(config)#ip ssh version 2
Please create RSA keys (of atleast 768 bits size) to enable SSH v2.
R_1_1(config)#crypto key generate rsa
The name for the keys will be: R_1_1.ARP.NET
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys, keys will be non-exportable...[OK]

R_1_1(config)#
*Apr 14 15:23:54.443: %SSH-5-ENABLED: SSH 2.0 has been enabled
R_1_1(config)#line vty 0 15
R_1_1(config-line)#transport inpt ssh
                   ^
% Invalid input detected at '^' marker.

R_1_1(config-line)#transport input ssh
R_1_1(config-line)#login local
R_1_1(config-line)#ex
% Ambiguous command:  "ex"
R_1_1(config-line)#exit
R_1_1(config)#ex
R_1_1#
*Apr 14 15:31:21.231: %SYS-5-CONFIG_I: Configured from console by console
R_1_1#
```

Рис. 3.8. Налаштування SSHv2

Internet

```
Internet(config)#int loopback 1
Internet(config-if)#
*Apr 14 15:54:13.015: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to u
Internet(config-if)#ip address 8.8.8.8 255.255.255.255
Internet(config-if)#no sh
Internet(config-if)#ex
Internet(config)#router ospf 21
Internet(config-router)#network 8.8.8.8 0.0.0.0 area 0
Internet(config-router)#exit
Internet(config)#
```

Рис. 3.9. Налаштування Loopback

R_1_1

```
R_1_1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R_1_1(config)#int Fa1/1
R_1_1(config-if)#standby 1 ip 10.10.40.250
R_1_1(config-if)#standby 1 preempt
R_1_1(config-if)#
*Apr 14 16:30:36.167: %HSRP-5-STATECHANGE: FastEthernet1/1 Grp 1 state Standby -> Active
R_1_1(config-if)#standby 1 priority 110
R_1_1(config-if)#end
R_1_1#
*Apr 14 16:31:53.943: %SYS-5-CONFIG_I: Configured from console by console
R_1_1#wr mem
Building configuration...
[OK]
R_1_1#
```

Рис. 3.10. Налаштування HSRP

```
R_1_1 (config)# Int fa0/0
R_1_1 (config-if)# description outside
R_1_1 (config-if)# ip nat outside
R_1_1 (config-if)# exit
R_1_1 (config)# int fa1/0
R_1_1 (config-if)# description dmz
R_1_1 (config-if)# exit
R_1_1 (config)# int fa1/1
R_1_1 (config-if)# description inside
R_1_1 (config-if)# ip nat inside
R_1_1 (config-if)# exit
R_1_1 (config)# ip access-list  standart FOR-NAT
R_1_1 (config-std-nacl)# permit 10.10.40.0 0.0.0.255
R_1_1 (config-std-nacl)# exit
R_1_1 (config)# ip nat inside source list FOR-NAT interface fa0/0 overload
R_1_1 (config)# ip access-list extended FROM-OUTSIDE
R_1_1 (config-std-nacl)# permit icmp any host 10.10.30.2
R_1_1 (config-std-nacl)# deny ip any any
R_1_1 (config-std-nacl)# exit
R_1_1 (config)# int fa0/0
R_1_1 (config-if)# ip access-group FROM-OUTSIDE in end
R_1_1 (config-if)# ip inspect name INSIDE-OUTSIDE tcp
R_1_1 (config)# int fa1/1
R_1_1 (config-if)# ip inspect INSIDE-OUTSIDE in
R_1_1 (config-if)# exit
R_1_1 (config)# ip access-list extended FROM-DMZ
R_1_1 (config-std-nacl)# deny ip host 10.10.30.2 10.10.40.0 0.0.0.255
R_1_1 (config-std-nacl)# permit ip any any
R_1_1 (config-std-nacl)# exit
R_1_1 (config)# int fa1/0
R_1_1 (config-if)# ip access-group FROM-DMZ in
R_1_1 (config-if)# end
```

Рис. 3.11. Було проведено налаштування DMZ

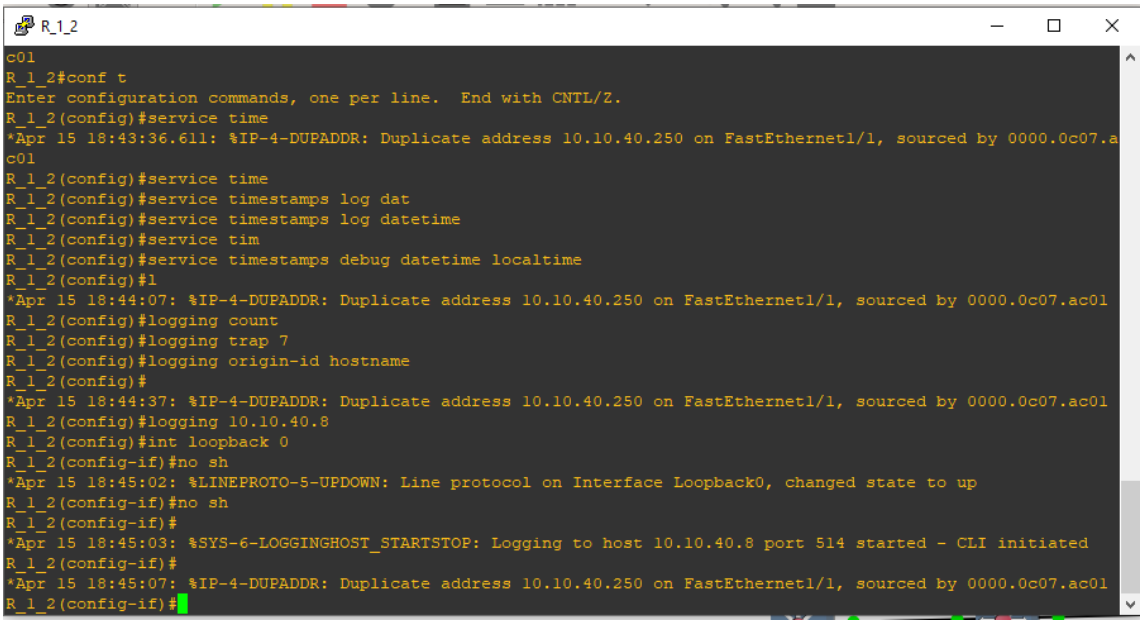


Рис. 3.12. Налаштування SysLog

Перегляд подій має в собі записи пов’язані з програмами на наших комп’ютерах, в разі того якщо додаток запущений з помилкою то в засобі буде відображена помилка. Також цей журнал нам потрібен для розуміння того, що хтось зайшов або вийшов з системи, або якось взаємодіяв з нею. Аудити пишуть

всі вдалі та відхилені спроби та всі додатки які встановлювались або оновлювались (рис. 3.18).

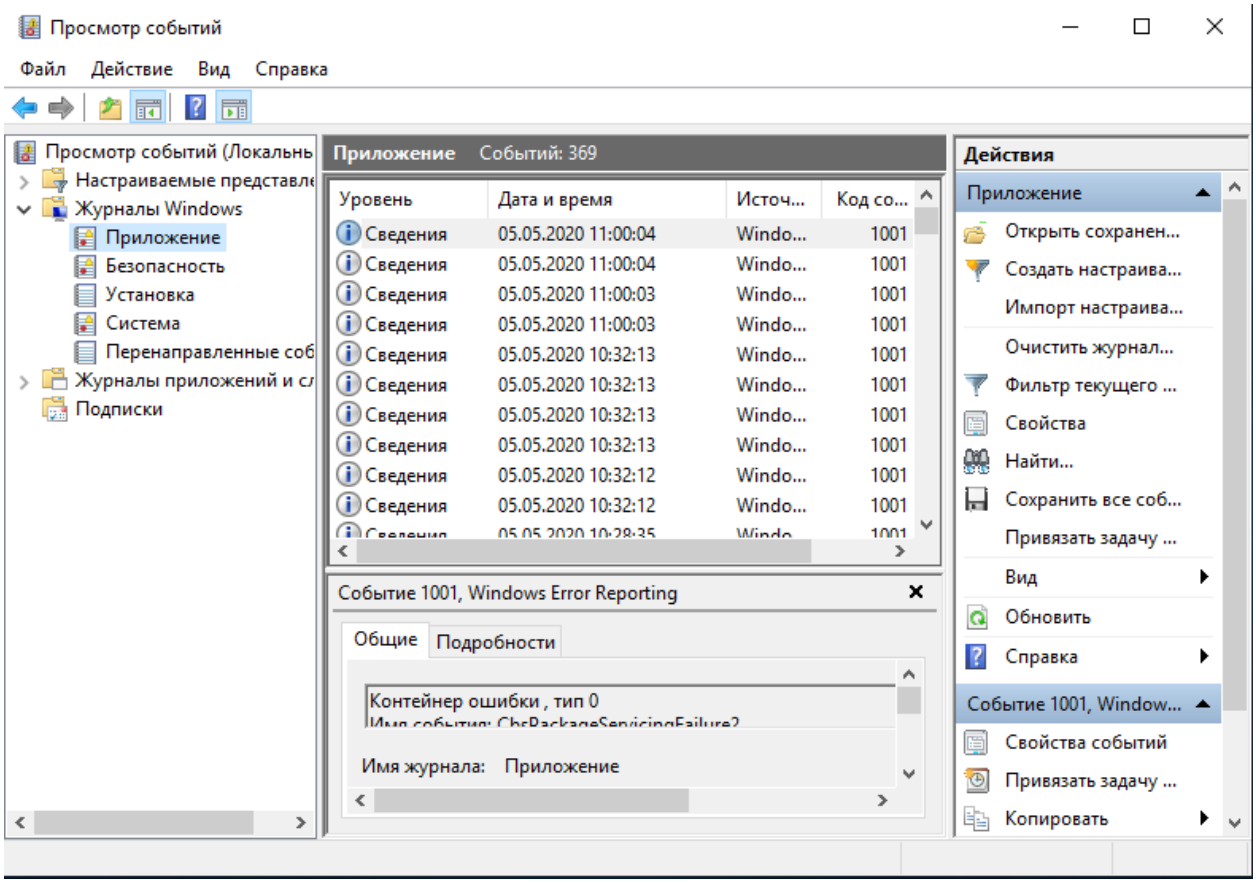


Рис.3.13.Вікно Журналу подій

В журналі подій присутній фільтр, в якому можливо відсіяти залишивши тільки те, що потрібно (рис. 3.14).

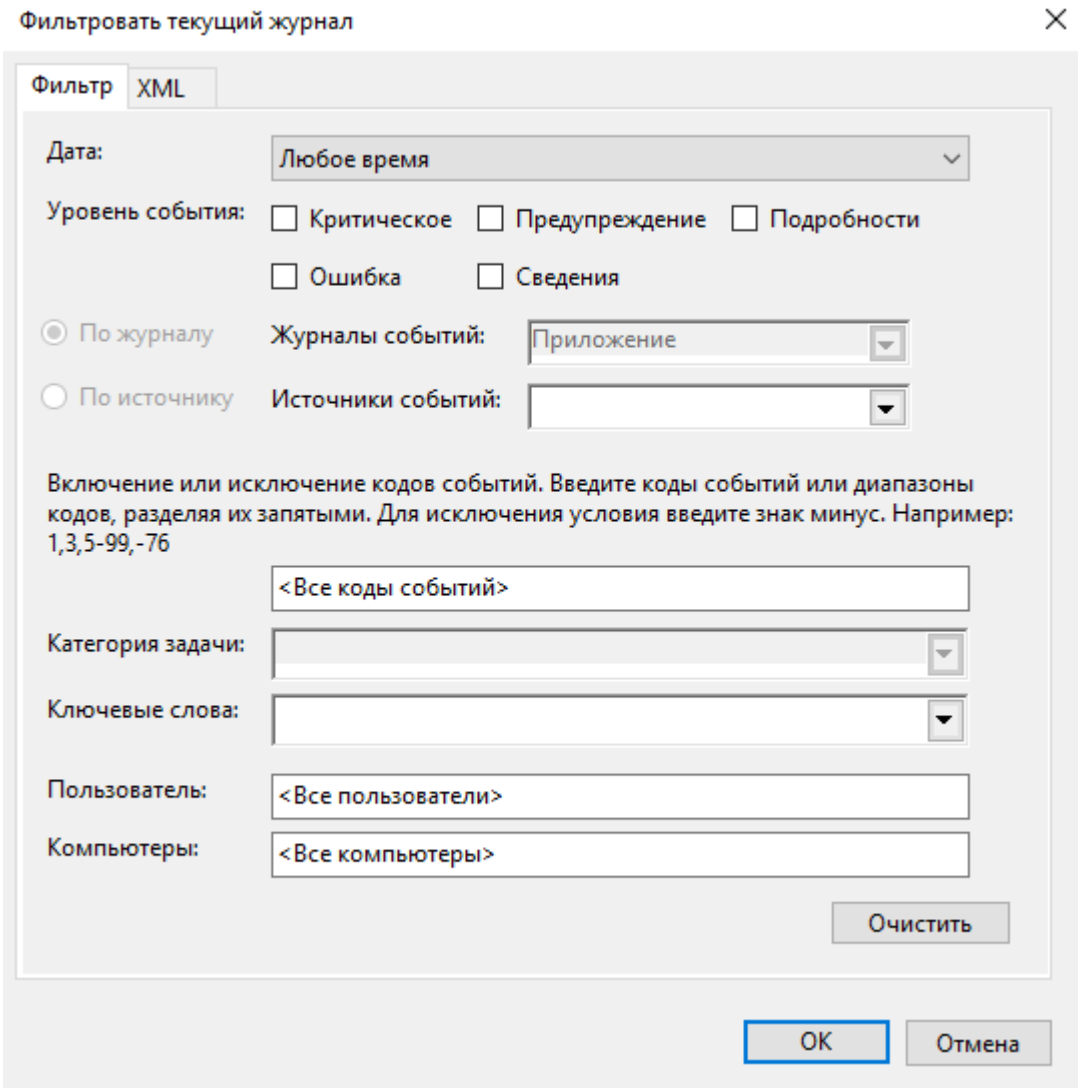


Рис.3.14.Фільтрація журналу перегляду подій

Переглянути логи можливо через Windows PowerShell наступною командою:

Get-EventLog –Logname ‘System’ (рис. 3.15) – для перегляду логів журналу Системи.

Get-EventLog –Logname ‘Application’ (рис. 3.16) – для перегляду логів журналу Додатків.

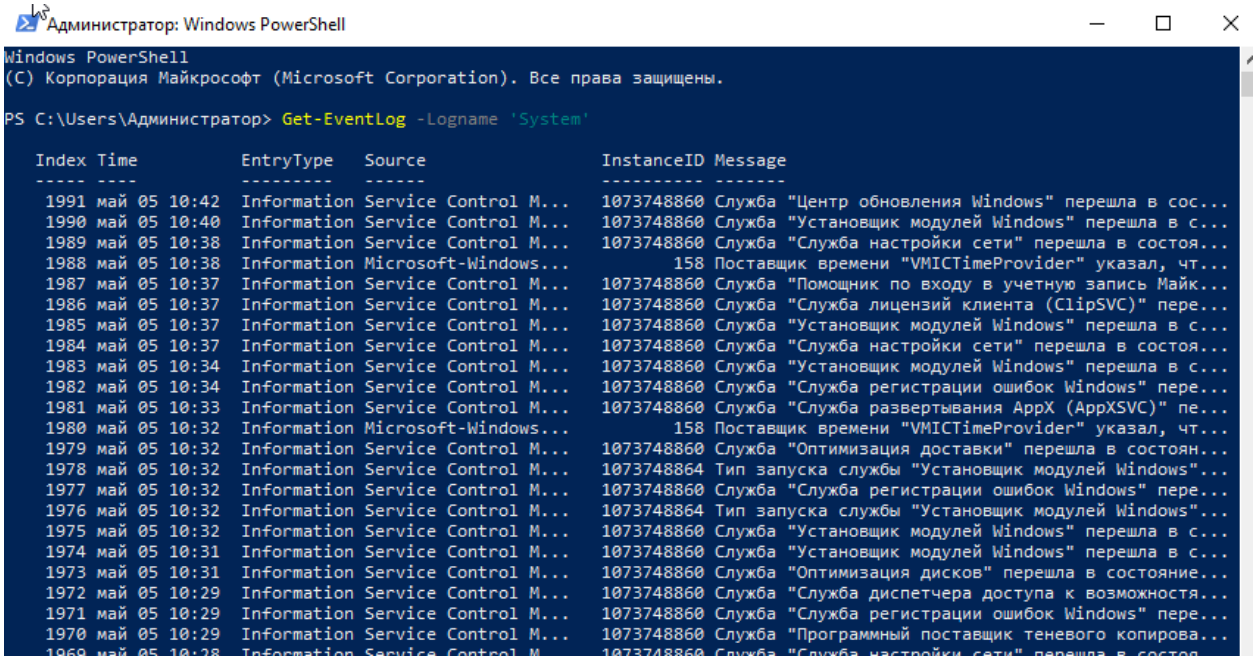


Рис. 3.15. Відображення списка логів журналу Системи

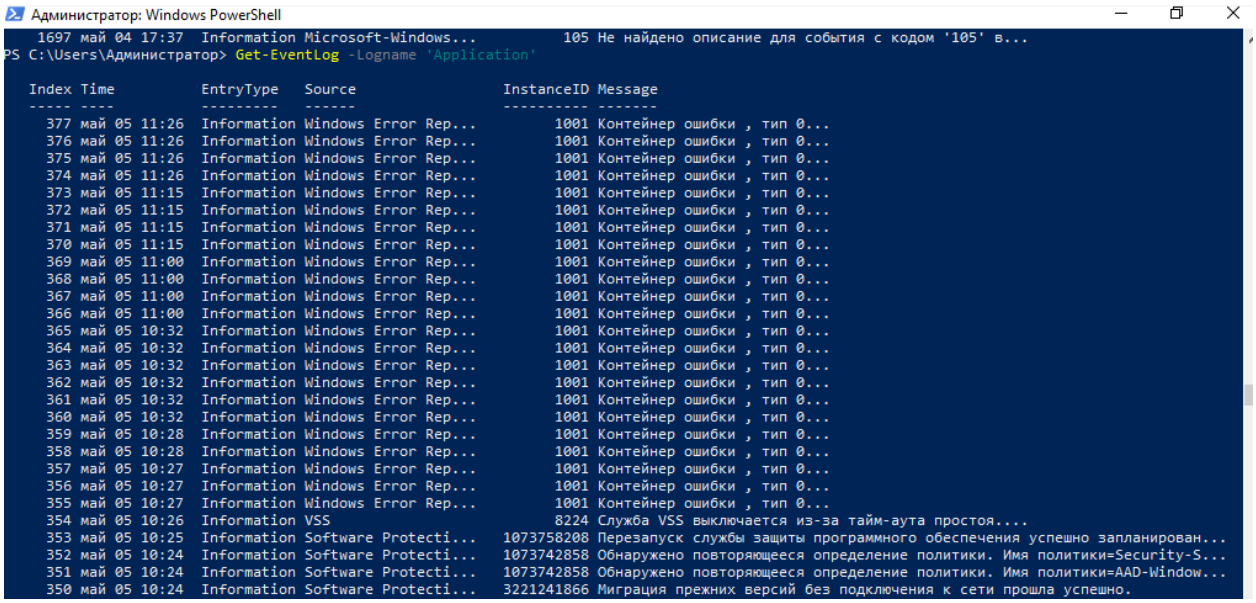


Рис. 3.16. Відображення списка журналів Додатків

Get-EventLog –Logname ‘System’| Format – Table, EntryType, TimeWritten, Source, EventID, Category, Message (рис. 3.22) – для перегляду логів журналу Системи більш детально.

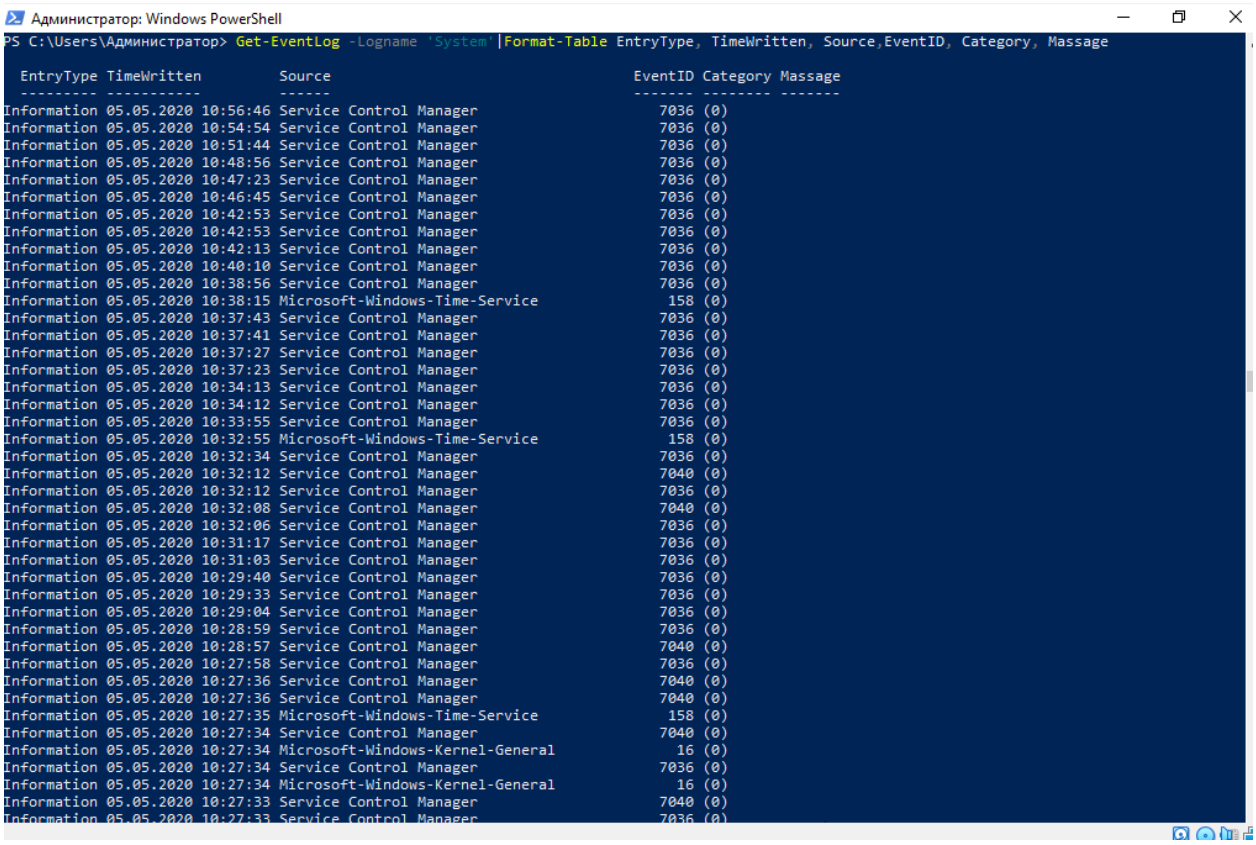


Рис. 3.17. Відображення більш детального списка журналів Системи

3.3. Перевірка працездатності мережі

Для перевірки працездатності мережі між кінцевими пристроями, а також шляху пакету при передачі було використано такі команди, як ping та tracert.

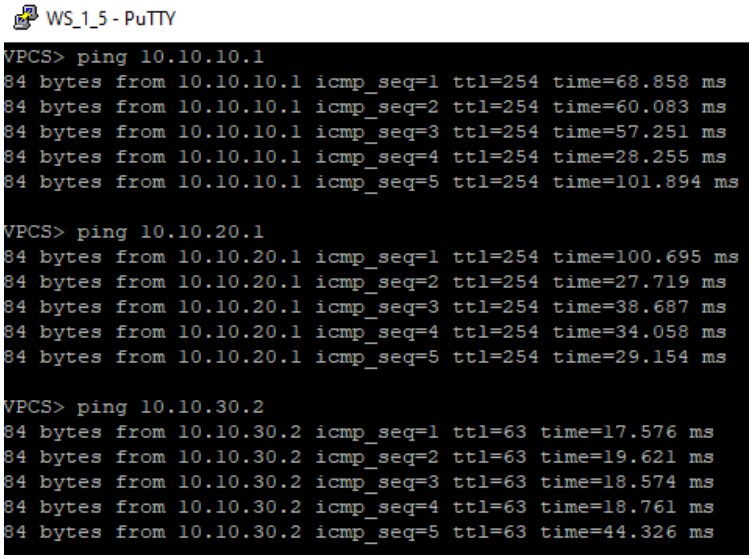


Рис.3.18. Перевірка працездатності мережі за допомогою команди ping

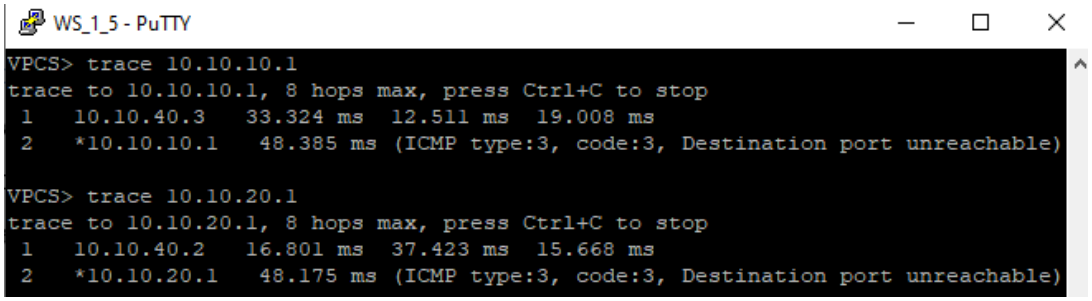


Рис. 3.19. Перевірка наявності зв'язку між кінцевими пристроями за допомогою команди tracert

Для перевірки механізму віддаленого доступу, було виконано вхід на маршрутизатор Internet з маршрутизатора R_1_1 за допомогою протоколу SSH (рис. 3.20).



Рис. 3.20.Перевірка можливості віддаленого підключення по SSH

Досліджено особливості отримання службової та діагностичної інформації протоколу за допомогою відповідних команд (рис. 3.26-3.27).



Рис. 3.21. Результат роботи команди show ip route

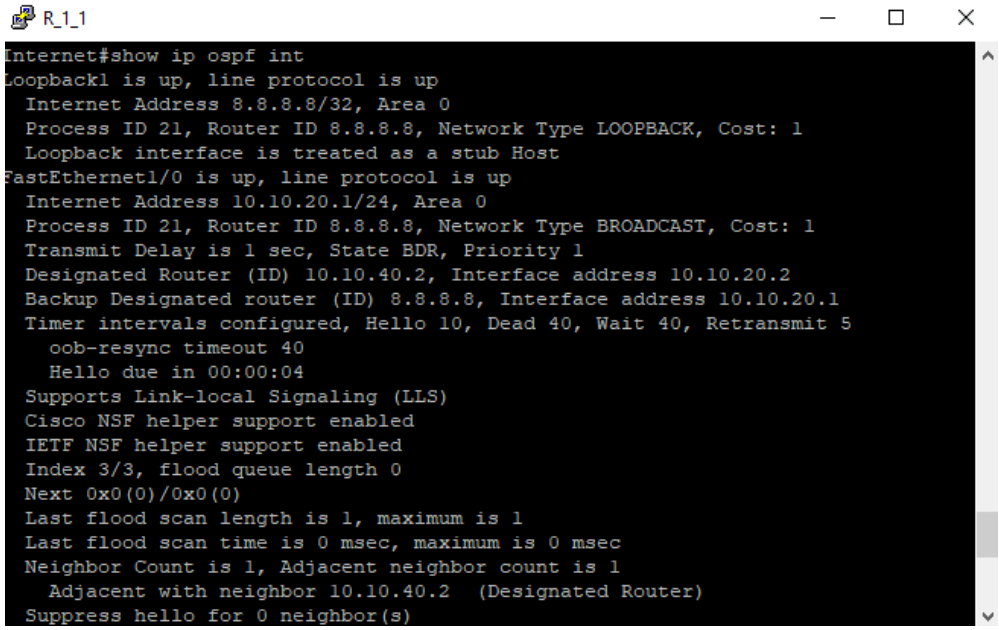


Рис. 3.22. Результат роботи команди show ip ospf int

Для перевірки механізму демілітаризованої зони, була використана команда ping з робочої станції WS_1_3 на маршрутизатор Internet (рис. 3.23-3.24).

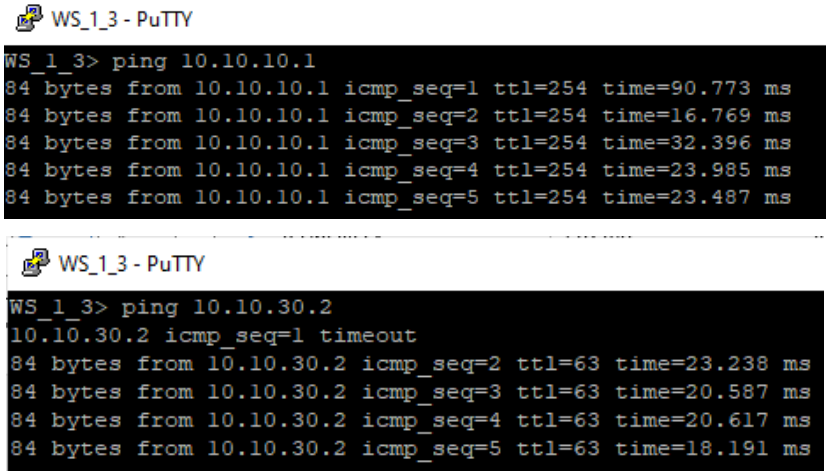


Рис. 3.23-3.24. Перевірка механізму демілітаризованої зони з WS_1_3

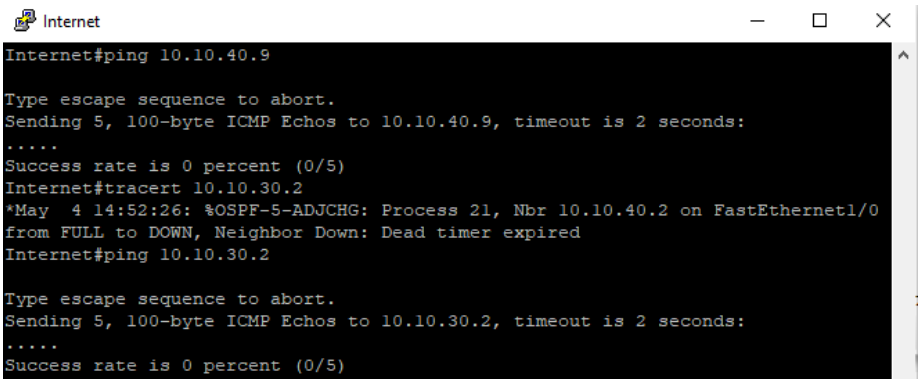


Рис. 3.25. Перевірка механізму демілітаризованої зони з Internet

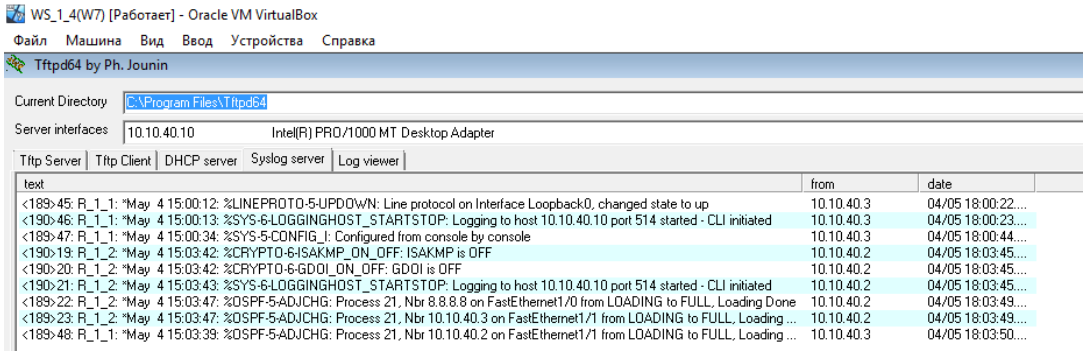


Рис. 3.26. Результат роботи відправки і реєстрації повідомлень про події в системі подій

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було спроектовано та реалізовано проекту підсистеми журналювання подій гетерогенної комп’ютерної мережі.

Було використано 4 підмереж, 3 маршрутизаторів та 6 комутаторів.

У першому розділі було обґрунтовано загальні принципи та аналіз програмного забезпечення пов’язані з Cisco system, IOS та підходи роботи журналювання подій.

У другому розділі було продумано та створено таблицю адресації пристроїв в глобальній мережі організації. Керуючись принципом економії IP-адрес, дві мережі було розбито на відповідну кількість підмереж, розміри яких відповідають потребам різних відділів організації.

У третьому розділі було обрано відповідні маршрутизатори та комутатори для використанні у організації роботи мережі та було налаштовано їх у програмному симуляторі GNS3. Також мережа була перевірена на працездатність та досліджено особливості отримання службової та діагностичної інформації протоколу за допомогою відповідних команд.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Налагодження віддаленого доступу та адміністрування. Основні завдання, які вирішує «SSH» [Електронний ресурс] – Режим доступу до ресурсу:https://learn.ztu.edu.ua/pluginfile.php/44687/mod_resource/content/1/KM_KIKB-LR-09-2018-2019_KI-02_KB-01.pdf.

2. Налагодження та дослідження роботи протоколу маршрутизації OSPF [Електронний ресурс] – Режим доступу до ресурсу: https://learn.ztu.edu.ua/pluginfile.php/50959/mod_resource/content/1/KM_KIKB_LR-21-2018-2019_KI-2KB-1.pdf

3. Робота технології NAT. [Електронний ресурс] – Режим доступу до ресурсу: https://learn.ztu.edu.ua/pluginfile.php/52835/mod_resource/content/1/KM_KIKB_LR-27-2018-2019_KI-2KB-1.pdf.

4. Протокол динамічного резервування шлюзу HSRP [Електронний ресурс] – Режим доступу до ресурсу: https://learn.ztu.edu.ua/pluginfile.php/52717/mod_resource/content/1/KM_KIKB_LR-28-2018-2019_KI-2KB-1.pdf.

5. SYSLOG [Електронний ресурс] – Режим доступу до ресурсу: https://learn.ztu.edu.ua/pluginfile.php/52838/mod_resource/content/1/KM_KIKB_LR-30-2018-2019_KI-2KB-1.pdf.

6. Демілітаризована зона DMZ [Електронний ресурс] – Режим доступу до ресурсу:https://www.cisco.com/c/en/us/support/docs/security/asa-5500-x-series-next-generation-firewalls/115904-asa-config-dmz-00.html.

7. Loopback [Електронний ресурс] – Режим доступу до ресурсу: http://microsin.net/adminstuff/cisco/loopback-null-tunnel-interfaces.html.

8. Зручний моніторинг Syslog повідомлень [Електронний ресурс] – Режим доступу до ресурсу: https://habr.com/ru/company/zabbix/blog/252915/.

9. Syslog в Linux [Електронний ресурс] – Режим доступу до ресурсу: https://unlix.ru/syslog-%D0%B2-linux-%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D

0%B8%D0%B5-%D0%BB%D0%BE%D0%B3%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%D0%BC/.

10. Системний журнал повідомлень [Електронний ресурс] – Режим доступу до ресурсу: https://www.cisco.com/c/en/us/td/docs/routers/access/wireless/software/guide/SysMsgLogging.html.

11. Аудит видалення і доступу до файлів і запис подій [Електронний ресурс] – Режим доступу до ресурсу: https://habr.com/ru/post/150149/.

12. Проблеми в системі журналювання подій [Електронний ресурс] – Режим доступу до ресурсу: https://habr.com/ru/post/476464/.

13. Журнал роботи Windows EventLog і системи оповіщення [Електронний ресурс] – Режим доступу до ресурсу: https://habr.com/ru/post/65652/.

14. Курячий Г.В. Операційні системи Linux/ Г.В. Курячий. - М: ДМК Пресс, 2010. - 348 с.

15. Тейлор Дейв, Сценарии командної оболочки/ Дейв Тейлор-М: СПБ: Питер ,2017. - 448 с.

16. Дехтярук М.Т. Програмне забезпечення комп'ютерних мереж/ М: Університет «Україна», 2012. - 353 с.

17. Микитишин А.Г. Комп'ютерні мережі /А.Г. Микитишин-М: Магноліт, 2013-256 с.

18. Войтов Н.В. Администрирование Red Hat Enterprice Linux/Н.В. Войтов-. М: ДМК Пресс, 2011. – 192 с.

19. Оливер Ибе, Компьютерные сети и службы удаленного доступа/Ибе Оливер -М: ДМК Пресс, 2019. - 336 с.

20. Рамський Ю. Адміністрування комп'ютерних мереж і систем/Ю. Рамский Тернопіль, 2010. - 196 с.

ДОДАТКИ

ДОДАТОК А

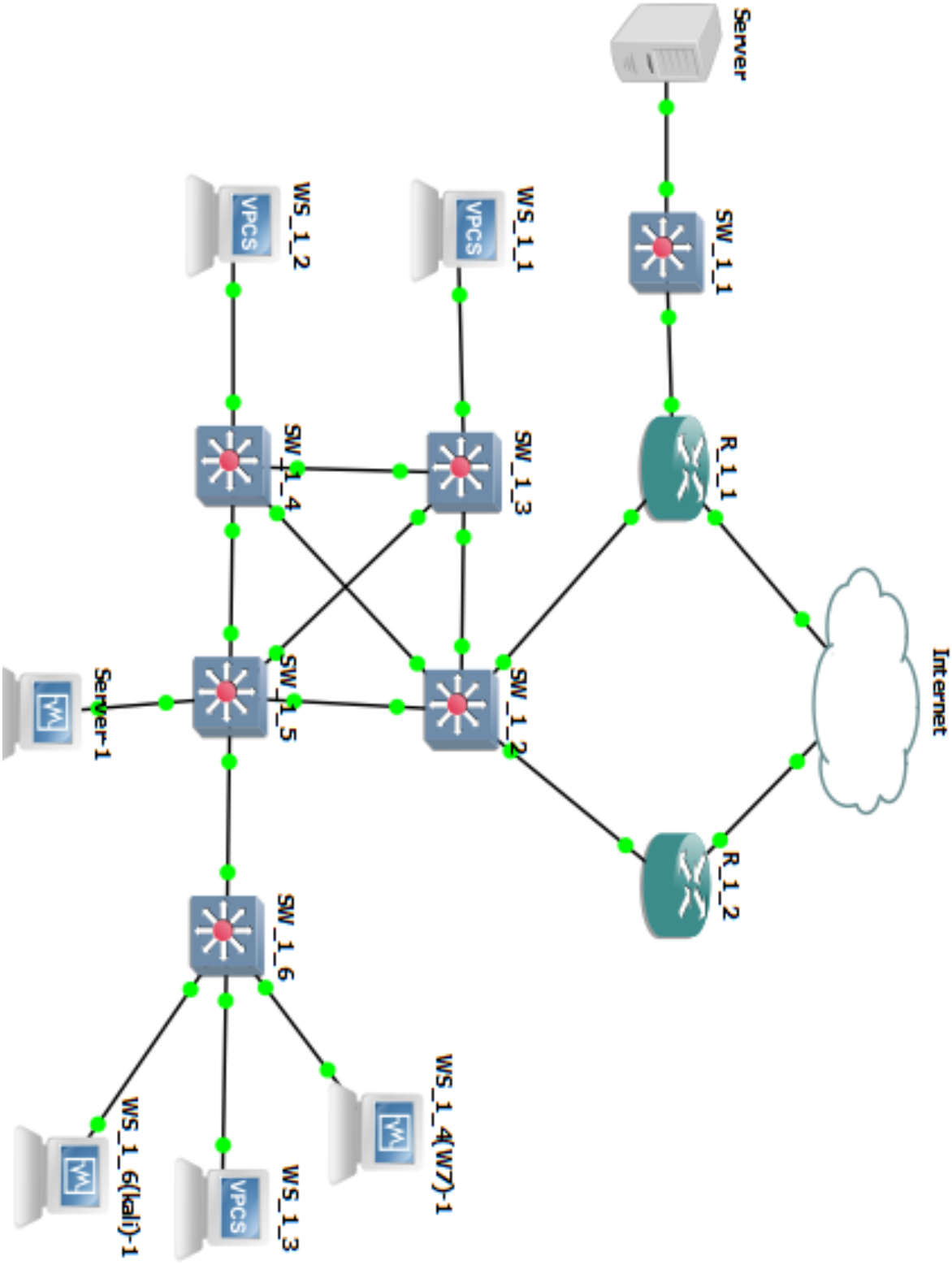


Рис.А.1.Схема мережі

ДОДАТОК Б

Нижче наведені лістинги конфігураційних файлів маршрутизаторів

```
R_1_1#show run
Building configuration...

Current configuration :
2297 bytes
upgrade fpd auto
version 12.4
service timestamps debug
datetime localtime
service timestamps log
datetime
no service password-
encryption
hostname R_1_1
boot-start-marker
boot-end-marker
logging count
logging message-counter
syslog
no aaa new-model
ip source-route
no ip icmp rate-limit
unreachable
ip cef
no ip domain lookup
ip domain name ARP.NET
ip inspect name INSIDE-
OUTSIDE tcp
ip inspect name INSIDE-
OUTSIDE icmp
ip inspect name INSIDE-
OUTSIDE http
no ipv6 cef
multilink bundle-name
authenticated
username admin password 0
cisco
archive
 log config
  hidekeys
ip tcp synwait-time 5
ip ssh version 2
interface Loopback0
 no ip address
interface FastEthernet0/0
 description OUTSIDE
 ip address 10.10.10.2
255.255.255.0
 ip access-group FROM-
OUTSIDE in
 ip nat outside
 ip virtual-reassembly
 duplex half
interface FastEthernet1/0
 description DMZ
 ip address 10.10.30.1
255.255.255.0
 ip access-group FROM-DMZ
in
 duplex auto
 speed auto
interface FastEthernet1/1
 description INSIDE
 ip address 10.10.40.3
255.255.255.0
 ip nat inside
 --More--
*May  4 15:10:12: %IP-4-
DUPADDR: Duplicate address
10.10.40.250 on
FastEthernet1/1, sourced by
0000.0c07.ac02
 ip inspect INSIDE-OUTSIDE
in
 ip virtual-reassembly
 duplex auto
 speed auto
 standby 1 ip 10.10.40.250
 standby 1 priority 110
 standby 1 preempt
router ospf 21
 log-adjacency-changes
 network 10.10.10.0
0.0.0.255 area 0
 network 10.10.30.0
0.0.0.255 area 0
 network 10.10.40.0
0.0.0.255 area 0
ip forward-protocol nd
no ip http server
no ip http secure-server
ip nat inside source list
FOR-NAT interface
FastEthernet0/0 overload
ip access-list standard
FOR-NAT
```

```
   permit 10.10.40.0
0.0.0.255
  ip access-list extended
FROM-DMZ
   deny   ip host 10.10.30.2

  10.10.40.0 0.0.0.255
   permit ip any any
  ip access-list extended
FROM-OUTSIDE
   permit icmp any host
10.10.30.2
   deny   ip any any
  logging trap debugging
  logging origin-id hostname
  logging 10.10.40.10
  no cdp log mismatch duplex
  control-plane
  gatekeeper
   shutdown
  line con 0
   exec-timeout 0 0
   privilege level 15
   password cisco
   logging synchronous
   login
   stopbits 1
  line aux 0
   exec-timeout 0 0
   privilege level 15
   password cisco
   logging synchronous
   stopbits 1
  line vty 0 4
   password cisco
   login local
   transport input ssh
  line vty 5 15
   login local
   transport input ssh
  end
```

```
R_1_2#show run
Building configuration...
Current configuration : 1609
bytes
upgrade fpd auto
version 12.4
service timestamps debug
datetime localtime
service timestamps log
datetime
no service password-encryption
hostname R_1_2
boot-start-marker
boot-end-marker
logging count
logging message-counter syslo
no aaa new-model
ip source-route
no ip icmp rate-limit
unreachable
ip cef
no ip domain lookup
ip domain name ARP.NET
no ipv6 cef
multilink bundle-name
authenticated
username admin password 0
cisco
archive
 log config
  hidekeys
ip tcp synwait-time 5
ip ssh version 2
interface Loopback0

 no ip address
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
interface FastEthernet1/0
 ip address 10.10.20.2
255.255.255.0
 duplex auto
 speed auto
interface FastEthernet1/1
 ip address 10.10.40.2
255.255.255.0
 duplex auto
 speed auto
 standby 0 preempt
 standby 2 ip 10.10.40.250
 standby 2 priority 90
```

```
 standby 2 preempt
router ospf 21
 log-adjacency-changes
 network 10.10.20.0 0.0.0.255
area 0
 network 10.10.40.0 0.0.0.255
area 0
ip forward-protocol nd
no ip http server
no ip http secure-server
logging trap debugging
logging origin-id hostname
logging 10.10.40.10
no cdp log mismatch duplex
--More--
*May  4 15:13:54: %IP-4-
DUPADDR: Duplicate address
10.10.40.250 on
FastEthernet1/1, sourced by
0000.0c07.ac01
control-plane
gatekeeper
 shutdown
line con 0
 exec-timeout 0 0
 privilege level 15
 password cisco
 logging synchronous
 login
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 password cisco
 logging synchronous
 stopbits 1
line vty 0 4
 password cisco
 login local
 transport input ssh
line vty 5 15
 login local
 transport input ssh
End
Internet#show run
Building configuration...
Current configuration : 1614
bytes
upgrade fpd auto
version 12.4
service timestamps debug
datetime localtime
service timestamps log
datetime
no service password-encryption
hostname Internet
boot-start-marker
boot-end-marker
logging count
logging message-counter syslog
no aaa new-model
ip source-route
no ip icmp rate-limit
unreachable
ip cef
no ip domain lookup
ip domain name ARP.NET
no ipv6 cef
multilink bundle-name
authenticated
username admin password 0
cisco
archive
 log config
  hidekeys
ip tcp synwait-time 5
ip ssh version 2
interface Loopback0
 no ip address
interface Loopback1
 ip address 8.8.8.8
255.255.255.255
interface FastEthernet0/0
 ip address 10.10.10.1
255.255.255.0
 duplex half
interface FastEthernet1/0
 ip address 10.10.20.1
255.255.255.0
 duplex auto
 speed auto
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
router ospf 21
 log-adjacency-changes
 network 8.8.8.8 0.0.0.0 area
0
 network 10.10.10.0 0.0.0.255
area 0
 network 10.10.20.0 0.0.0.255
area 0
ip forward-protocol nd
```

```
no ip http server
no ip http secure-server
logging trap debugging
logging origin-id hostname
logging 10.10.40.10
no cdp log mismatch duplex
control-plane
gatekeeper
 shutdown
line con 0
 exec-timeout 0 0
 privilege level 15
 password cisco
 logging synchronous
 login
 stopbits 1
line aux 0

 exec-timeout 0 0
 privilege level 15
 password cisco
 logging synchronous
 stopbits 1
line vty 0 4
 password cisco
 login local
 transport input ssh
line vty 5 15
 login local
 transport input ssh
end
```