# Лабораторна робота №13

## Імпорт логів із Wazuh у Splunk.

**Мета**: Метою роботи є освоєння процесу інтеграції системи моніторингу безпеки Wazuh із платформою аналітики Splunk шляхом налаштування передачі подій, створення індексу для логів Wazuh, перевірки надходження даних.

**Інструменти**: гіпервізор VirtualBox, модель комп'ютерної мережі.

## Теоретичні відомості

У попередніх роботах створено стендове середовище у VirtualBox, що складається з чотирьох хостів:

Serv-G-N-1 (Windows Server 2022) – контролер домену з ролями AD DS, DNS і DHCP. Налаштовано Wazuh Agent для локального моніторингу ресурсів.

Serv-G-N-5 (Ubuntu Server 24.04) – сервер на налаштовано Wazuh Agent для локального моніторингу ресурсів.

Serv-G-N-7 (Amazon Linux 2023) – сервер Wazuh Appliance, що містить компоненти Wazuh Server (Manager, API) та Elasticsearch + Kibana (Dashboard)

Serv-G-N-9 (Ubuntu Server 24.04) – сервер Splunk Free — локальний індексатор та аналітична платформа для збору, обробки та візуалізації журналів безпеки. Сервер призначений для прийому логів з операційних систем та сервісів мережі, а також для інтеграції з Wazuh через Wazuh Splunk App.

Мережеве середовище забезпечує взаємодію між вузлами з доменною інфраструктурою для подальшого моніторингу її елементів.

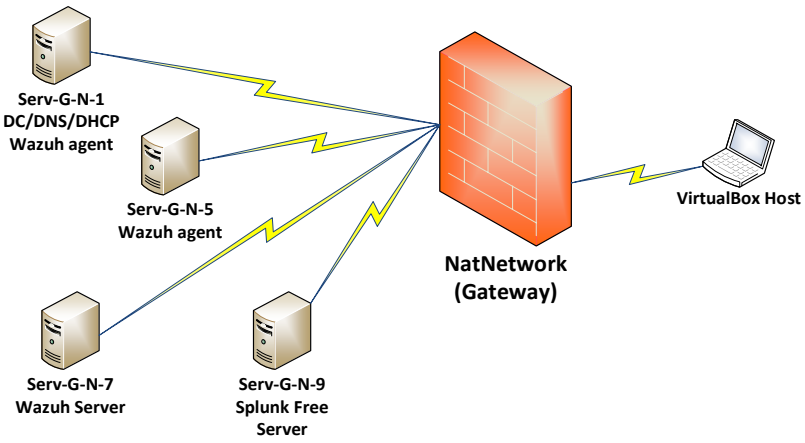



*Рис. 13.1. Топологія мережі*

## Коротка теорія (Wazuh – Splunk: Filebeat, HEC, Wazuh App)

Інтеграція Wazuh зі Splunk дозволяє збирати всі події безпеки, які генерують Wazuh-агенти, і відображати їх у Splunk у вигляді дашбордів, пошукових запитів і звітів. Щоб це працювало, у Wazuh використовується Filebeat, а у Splunk — HTTP Event Collector (HEC) та спеціальний застосунок Wazuh App.

- ***Filebeat у Wazuh Appliance***

У Wazuh Appliance (Serv-G-N-7) використовується компонент Filebeat, який читає лог-файли менеджера Wazuh, упаковує події у JSON та надсилає їх у зовнішні системи SIEM (Splunk, Elasticsearch, Logstash).

З точки зору інтеграції важливо що Filebeat підтримує Splunk HEC як транспорт, а Wazuh вже містить шаблон для Splunk у своєму пакеті.

- ***Splunk HEC (HTTP Event Collector)***

HEC (HTTP Event Collector) — це механізм Splunk, який приймає події через HTTP/HTTPS. Для інтеграції потрібно увімкнути HEC у Splunk, створити токен доступу (authentication token), дозволити прийом JSON-подій, відкрити порт (зазвичай 8088/tcp або інший, якщо змінили).

Після цього Filebeat зможе надсилати події без прямої установки Forwarder-а на сервері.

- ***Wazuh Splunk App (візуалізація)***

Wazuh App for Splunk — це офіційний застосунок, який містить готові дашборди для FIM (File Integrity Monitoring), аналізу активних відповідей (Active Response), аналітики подій агентів, візуалізацій MITRE ATT&CK та модулів Syscheck, Rootcheck, Vulnerability Detector та ін.

Він працює поверх Splunk Search і не вимагає додаткової конфігурації — лише правильного sourcetype та index.

❖ ***Як відбувається передача подій (спрощена схема)***

Wazuh Manager генерує події (FIM, Syscheck, Rootcheck, інтеграції, агенти). Filebeat читає їх із log-файлів та формує JSON. Filebeat надсилає події через HTTP до Splunk HEC endpoint. Splunk приймає події та розміщує їх у вказаному індексі (наприклад, wazuh). Wazuh Splunk App відображає їх у вигляді дашбордів.

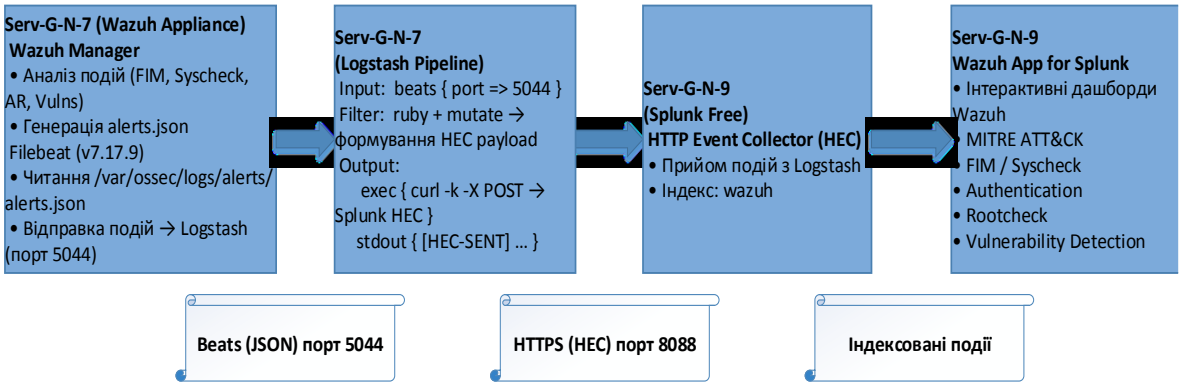



*Рис. 13.2. Архітектура інтеграції Wazuh–Logstash–Splunk*

Основні риси, що визначають взаємодію Wazuh та Splunk у процесі побудови централізованої системи безпекового моніторингу:

- ❖ Universal Forwarder не використовується. Передавання подій здійснюється безпосередньо через Filebeat, що працює на Wazuh Manager.
- ❖ Комунікація відбувається через один порт – 8088/TCP. Це порт прийому подій Splunk HEC (HTTP Event Collector).
- ❖ Для подій Wazuh створюється окремий індекс у Splunk. Це забезпечує структурованість даних та спрощує роботу з дашбордами.
- ❖ Після надходження подій активується модуль “Wazuh App for Splunk”. Додаток автоматично будує візуалізації й дашборди для аналізу безпеки.

## Підготовка Splunk (увімкнення HEC)

Перед увімкненням HEC створимо робочий індекс з іменем wazuh (через Splunk CLI):

***sudo -u splunk /opt/splunk/bin/splunk add index wazuh***

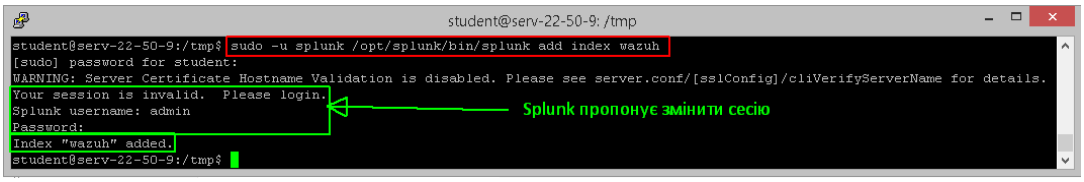



*Рис. 13.3. Створення індексу wazuh у CLI*

Виконуємо послідовність дій для увімкнення HTTP Event Collector (HEC).

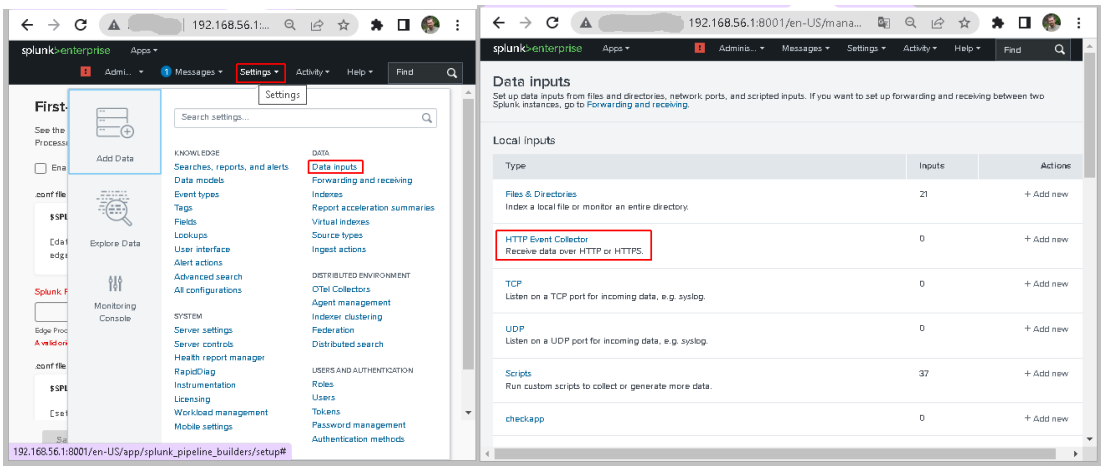



*Рис. 13.4. Вхід до меню HEC*

Входимо до Splunk Web, переходимо у меню Settings – Data Inputs. У рядку “Splunk REST API Origin” вводимо адресу розгорнутого на хості Serv-G-N-9 Splunk у відповідності до формату HTTPS, навіть якщо він розгорнутий локально https://<IP адреса твого сплунк сервера>:8088. Splunk внутрішньо використовує HTTPS для свого REST API на порту 8089 — незалежно від того, чи веб-інтерфейс доступний по HTTPS.

У випадку, наведеному на рис. 13.4 ця адреса відповідає https://192.168.50.11:8088. Натискаємо Save внизу вікна та потрапляємо у меню Splunk Data inputs/Local inputs де можливо увімкнути HEC. У списку обираємо HTTP Event Collector.

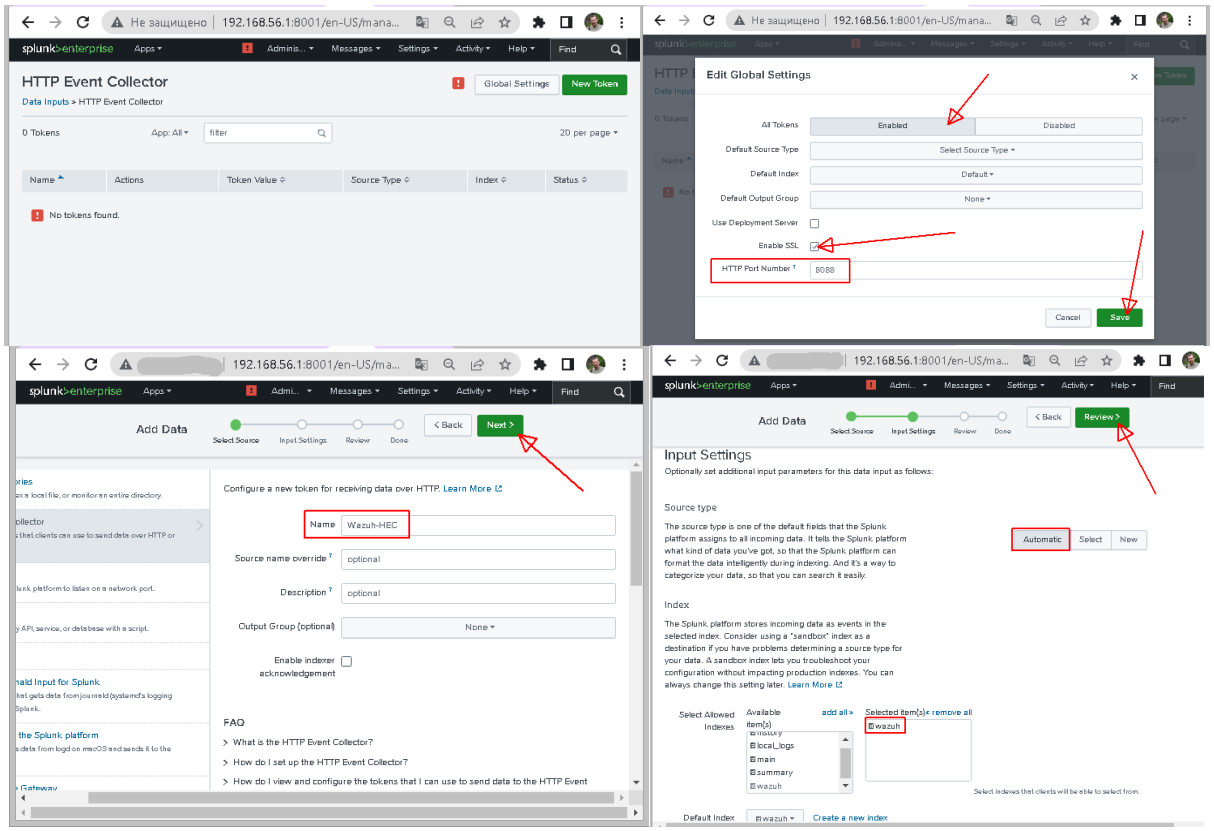



*Рис. 13.5. Генерація токену Wazuh-HEC*

Натискаємо Global Settings та обираємо All Tokens – Enables, Enable SSL = Yes та HTTP Port Number = 8088 (за замовчуванням) і зберігаємо зміни (Save).

У цьому ж меню натискаємо New Token та потрапляємо у вікно «Configure a new token for receiving data over HTTP» Вводимо назву токену Wazuh-HEC. Переходимо далі у полі Source type залишаємо Automatic та обираємо у полі Index створений на початку цього пункту методики у CLI індекс wazuh. Якщо індекс не обрати, Splunk не прийматиме події. Послідовно обираємо пункти Review та Submit.

Переглянути, перевірити стан та скопіювати Token Value (він знадобиться Filebeat) можливо по кнопці Show у пункті Settings – Data Inputs – HTTP Event Collector (рис.13.6).

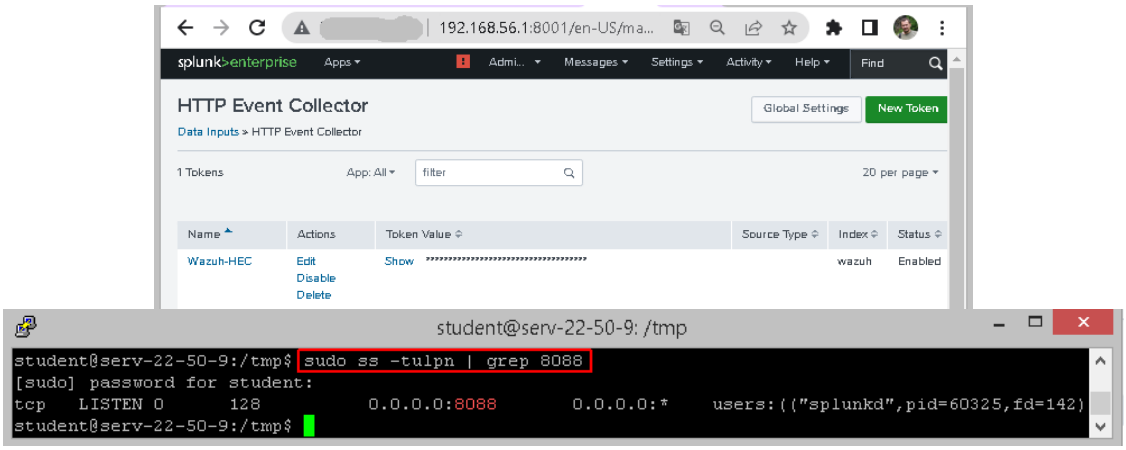



*Рис. 13.6. Перегляд Settings – Data Inputs – HTTP Event Collector та перевірка “чи слухається порт 8088»*

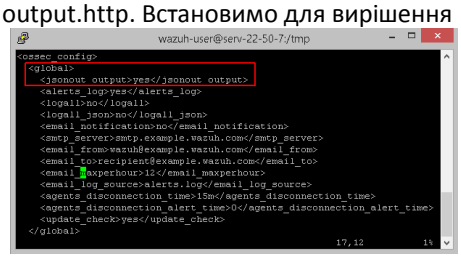
Перевіряємо прослуховування порту в системі на Splunk-сервері (Ubuntu):

***sudo ss -tulpn | grep 8088***

Таким чином, ми отримали очікуваний результат – Splunkd слухає порт 8088 (рис.13.6). Виконані дії у цьому пункті: HEC увімкнули, порт прослуховується, токен є.

## Інтеграція Wazuh – Splunk через Universal Forwarder

Ми використовуємо Wazuh Manager (Appliance), який поставляється з Filebeat, що жорстко прив'язаний до Elasticsearch. Така конфігурація Wazuh не підтримує output.http. Встановимо для вирішення цієї проблеми Splunk Universal Forwarder.




Перед встановленням та першим запуском SplunkUF переконаємось, що JSON-вивід увімкнений у файлі ***/var/ossec/etc/ossec.conf***. Перевіряємо, що присутній блок:

```
<global>
  <jsonout_output>yes</jsonout_output>
</global>
```

Перезапускаємо Wazuh Manager та переконуємось, що файл з логами існує:

***sudo systemctl restart wazuh-manager***

***sudo ls -l /var/ossec/logs/alerts/alerts.json***

Оскільки подальші кроки лабораторної роботи передбачають роботу зі службами та конфігураційними файлами Wazuh, а також читання файлів журналів у каталозі ***/var/ossec/logs/alerts***, необхідно забезпечити користувачу робочої станції право доступу до цих файлів. У Wazuh Appliance власником файлів Wazuh є користувач ***wazuh*** і група ***wazuh***. За замовчуванням каталоги логів мають права:

***drwxrwx--- wazuh:wazuh /var/ossec/logs***

***drwxr-x--- wazuh:wazuh /var/ossec/logs/alerts***

Тому користувачі, які не входять до групи ***wazuh***, не можуть читати файли alerts.json. Щоб надати права, додаємо робочого користувача до групи wazuh:

***sudo usermod -aG wazuh wazuh-user***

Після цього потрібно виконати повторний вхід до системи або команду:

***su - wazuh-user***

Після оновлення прав доступ до логів можна перевірити командою:

***sudo ls -lh /var/ossec/logs/alerts/alerts.json***

Визначаємо архітектуру нашого віртуального хосту та вантажимо з офіційної сторінки продукту https://www.splunk.com/en_us/download/universal-forwarder.html Splunk Universal Forwarder (RPM) у каталог /tmp серверу Serv-G-N-7 (Amazon Linux 2023), скопіювавши офіційне посилання (рис. 13.7).

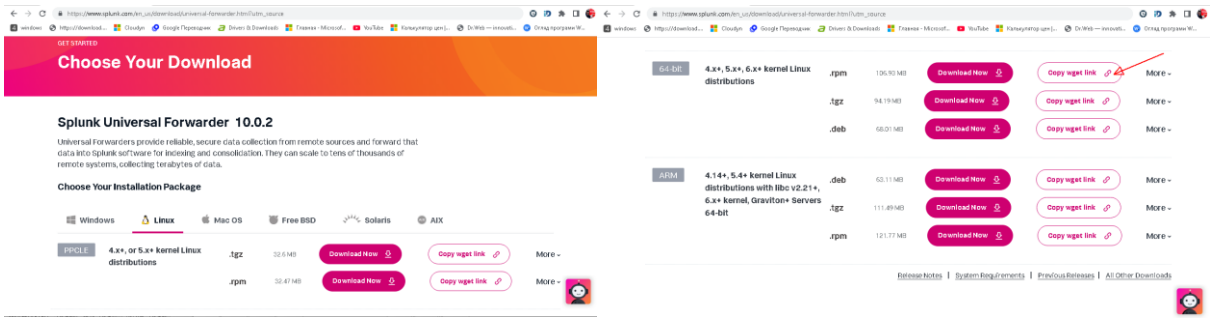



*Рис. 13.7. Отримання лінку для завантаження пакету Splunk Universal Forwarder (RPM) 64-bit*

***cd /tmp***

***wget -O splunkforwarder-10.0.2-e2d18b4767e9.x86_64.rpm***
***https://download.splunk.com/products/universalforwarder/releases/10.0.2/linux/splunkforwarder-10.0.2-e2d18b4767e9.x86_64.rpm***

Якщо wget повідомить про помилку 404, отримуємо на сторінці актуальне посилання. В різних версіях змінюється build-id, тому 404 — нормальна ситуація.

При виникненні незрозумілих помилок у роботі чи запуску Splunk Universal Forwarder на ВМ Wazuh Appliance (Amazon Linux 2023) рекомендовано виконати повне чисте видалення Splunk UF та повторити встановлення. Процедура видалення Splunk Universal Forwarder на ВМ Wazuh Appliance (Amazon Linux 2023) описана у додатку 1 цього документу.


*Рис. 13.8. Завантаження пакету Splunk Universal Forwarder (RPM) 64-bit*

Пересвідчуємось, що RPM-пакет завантажено без збоїв та встановлюємо його, підставивши у команду реальне ім'я завантаженого файлу:

***sudo rpm -ivh splunkforwarder.rpm***

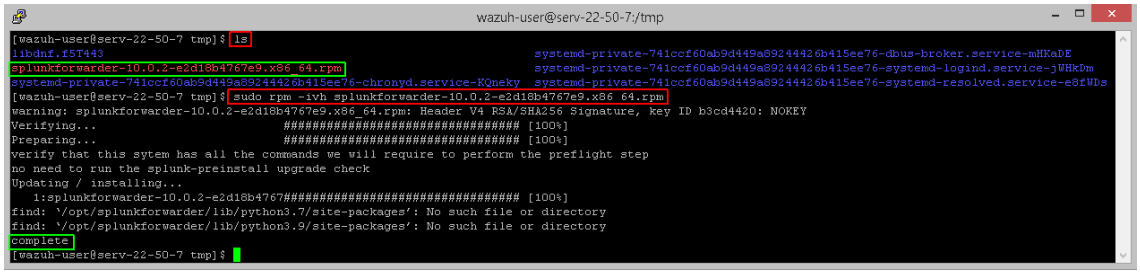



*Рис. 13.9. Встановлення пакету Splunk Universal Forwarder (RPM) 64-bit*

Автоматичне встановлення Splunk UF відбувається у каталог ***/opt/splunkforwarder***. Після цього є базова структура каталогів UF.Splunk


*Рис. 13.10. Базова структура каталогів Splunk UF*

UF автоматично створить користувача splunkfwd, під яким працюватиме служба. Надаємо йому права на Splunk UF каталоги:

***sudo chown -R splunkfwd:splunkfwd /opt/splunkforwarder***

Виконуємо перший запуск UF з прийняттям ліцензії. UF згенерує базові сертифікати, створить каталоги конфігурацій і підготує систему до підключення до Deployment Server.

***sudo -u splunkfwd /opt/splunkforwarder/bin/splunk start --accept-license***

При виконанні цієї команди Splunk попросить створити адміністратора CLI, що ми виконуємо. Пропоноване ім'я адміністратора splunker, у якості паролю використовуємо будь-який сильний пароль

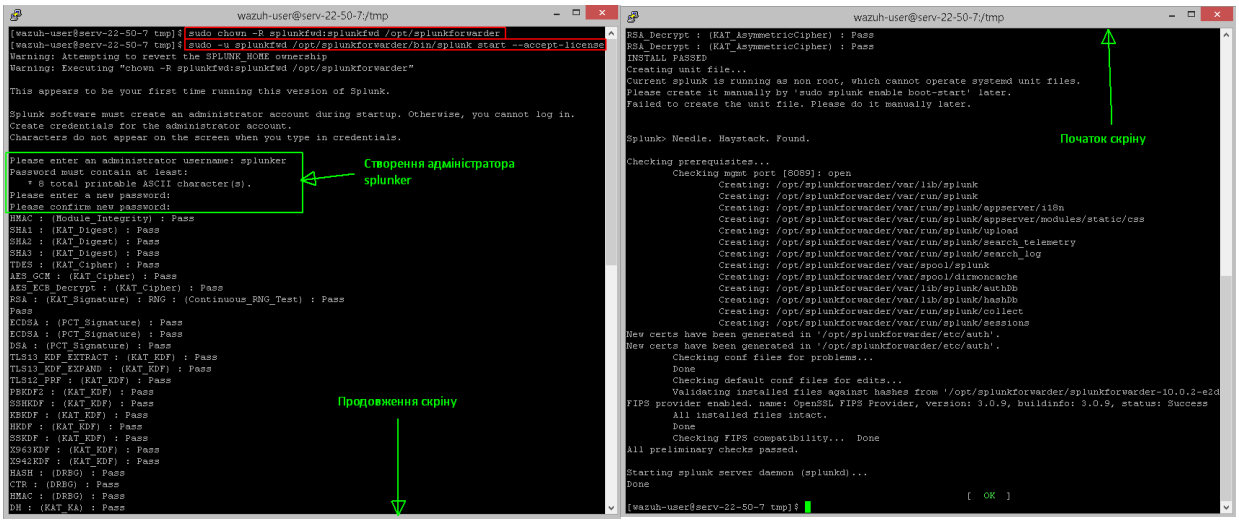
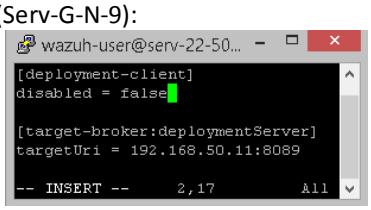



*Рис. 13.11. Перший запуск Splunk Universal Forwarder*

Створюємо файл */opt/splunkforwarder/etc/system/local/deploymentclient.conf* – базову конфігурацію Deployment Client, що задає адресу Deployment Server, до якого UF буде підключатися. У якості <IP_Deployment_Server> «підставляємо» реальну IP-адресу Splunk серверу (Serv-G-N-9):

```
[deployment-client]
disabled = false

[target-broker:deploymentServer]
targetUri = <IP_Deployment_Server>:8089
```




Додаємо UF у systemd, щоб він стартував при завантаженні системи від користувача splunkfwd.

*sudo /opt/splunkforwarder/bin/splunk enable boot-start -user splunkfwd*

Перший виклик Enable boot-start для systemd встановлює unit systemd, але якщо splunkd вже запущений, потрібно його зупинити (pkill -f splunkd) перед повторним викликом та перезавантажити:

*sudo pkill -f splunkd*
*sudo /opt/splunkforwarder/bin/splunk enable boot-start -user splunkfwd*
*sudo systemctl daemon-reload*

Виконуємо контроль запуску через systemd: старт, зупинка, перезапуск та перевірка статусу UF

*sudo systemctl start SplunkForwarder*
*sudo systemctl stop SplunkForwarder*
*sudo systemctl restart SplunkForwarder*
*sudo systemctl status SplunkForwarder*

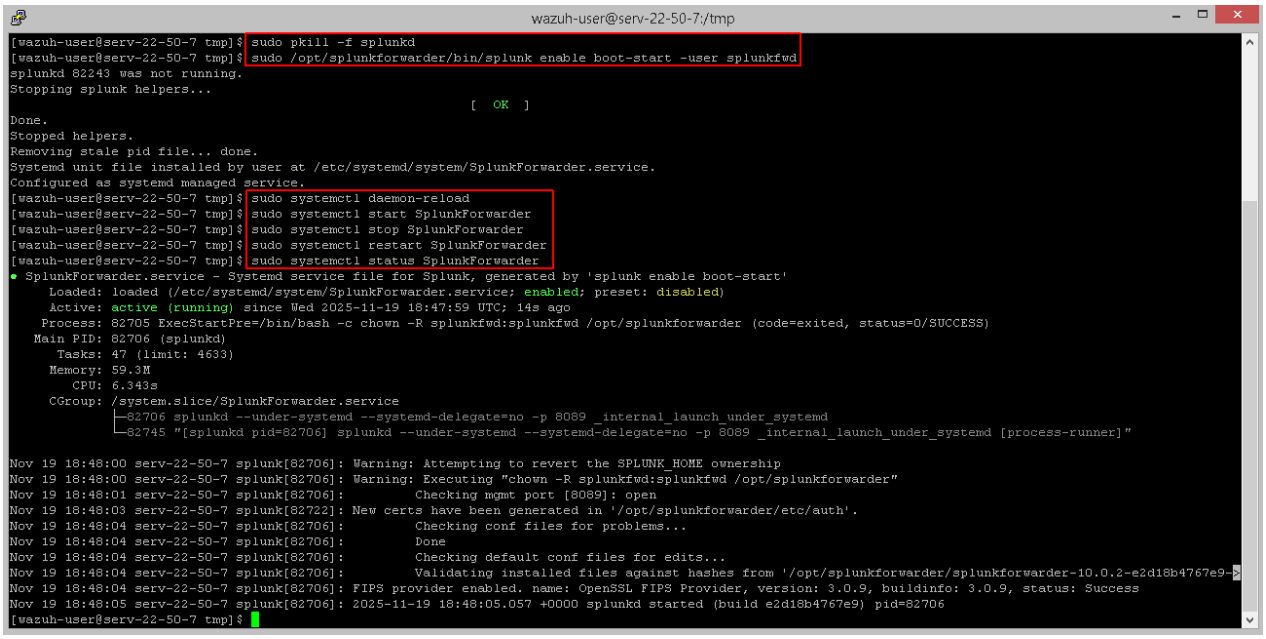



*Рис. 13.12. Налаштування та запуск служби Splunk Universal Forwarder*

На рис. 13.12 показане налаштування запуску Splunk Forwarder під systemd, служба активна (active, running), основний процес splunkd запущений, права користувача splunkfwd коректні. Тобто, з системної точки зору UF повністю готовий до роботи.

Виконуємо виправлення прав на каталоги (рекомендовано після будь-якого редагування конфігу)

*sudo chown -R splunkfwd:splunkfwd /opt/splunkforwarder*

Переходимо до наступного пункту методички.

## Автоматичне архівування логів Wazuh для стабільної роботи Filebeat

Filebeat не вміє працювати з вкладеною структурою директорій у модулі wazuh. Щоб уникнути падіння Filebeat через старі логи, можна автоматично переміщати та архівувати старі каталоги Wazuh. Створюємо файл скрипта */usr/local/bin/wazuh_archive.sh*, куди вставляємо наступний лістинг:

```
#!/bin/bash
ALERTS_DIR="/var/ossec/logs/alerts"
ARCHIVE_DIR="/var/ossec/logs/archives"
DATE=$(date +%Y-%m-%d-%H%M)
mkdir -p "$ARCHIVE_DIR"
for d in "$ALERTS_DIR"/*/ ; do
```

```
  [ -d "$d" ] || continue
  BASENAME=$(basename "$d")
  tar -czf "$ARCHIVE_DIR/alerts-$BASENAME-$DATE.tgz" -C "$ALERTS_DIR" "$BASENAME"
  rm -rf "$d"
done
```

Зберігаємо файл та робимо скрипт виконуваним:

***sudo chmod +x /usr/local/bin/wazuh_archive.sh***

ВМ, що ми використовуємо (Amazon Linux 2023) не має встановленого пакету cronie або crontabs. Встановлюємо його, активуємо та запускаємо сервіс:

***sudo dnf install cronie -y***
***sudo systemctl enable crond***
***sudo systemctl start crond***
***sudo systemctl status crond***

Для налаштування автоматичного запуску скрипта (через cron), відкриваємо cron для root (щоб скрипт мав права на всі файли Wazuh):

***sudo crontab -e***

та додаємо рядок, щоб скрипт ***/usr/local/bin/wazuh_archive.sh*** запускався щодня о 23:50:

```
50 23 * * * /usr/local/bin/wazuh_archive.sh
```

Зберігаємо файл cron та тестуємо роботу скрипта вручну:

***sudo /usr/local/bin/wazuh_archive.sh***

Переконуємось, що старі каталоги з ***/var/ossec/logs/alerts/*** перемістились у ***/var/ossec/logs/archives/*** у вигляді .tgz.

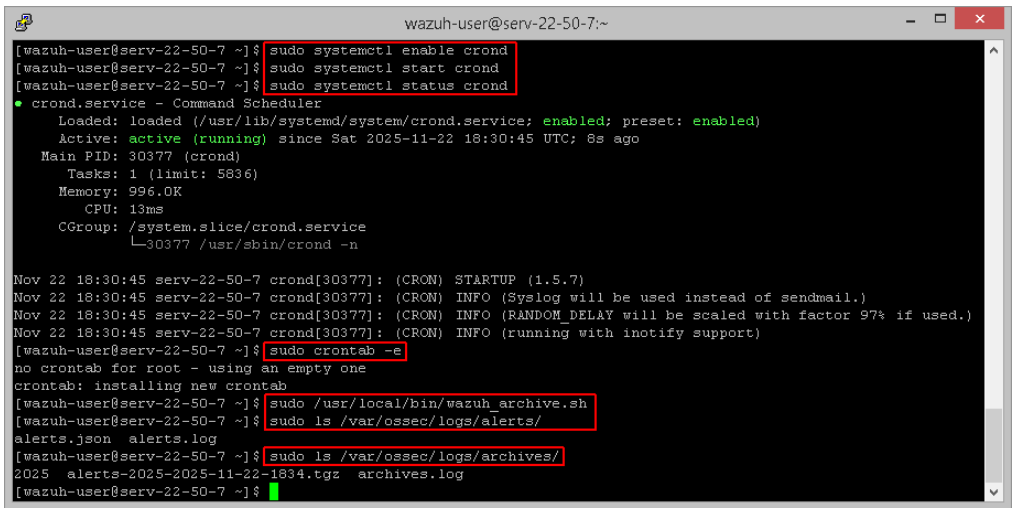



*Рис. 13.13. Налаштування автоматичного архівування логів Wazuh*

## Встановлення та Logstash на Wazuh Appliance (Amazon Linux 2023).

Logstash постачається через репозиторій Elastic. Виконуємо додавання репозиторію. Завантажуємо і встановлюємо ключ GPG та створюємо репозиторій для Elastic:

***sudo rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch***

Команда створення репозиторію для Elastic:

```
sudo tee /etc/yum.repos.d/elastic.repo <<EOF
[elastic-8.x]
name=Elastic repository for 8.x packages
baseurl=https://artifacts.elastic.co/packages/8.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
EOF
```

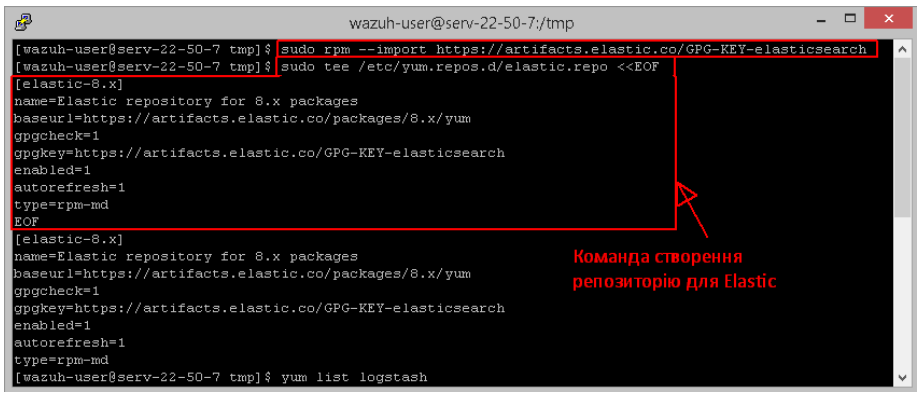



*Рис. 13.14. Встановлення ключа та створення репозиторію для Elastic*

Перевіряємо доступність пакету Logstash (рис.13.15):

***yum list logstash***

Для встановлюємо Logstash виконуємо:

***sudo yum install -y logstash***

Після встановлення перевіряємо, чи з'явилися каталоги:

***ls -l /etc/logstash***

***ls -l /etc/logstash/conf.d***

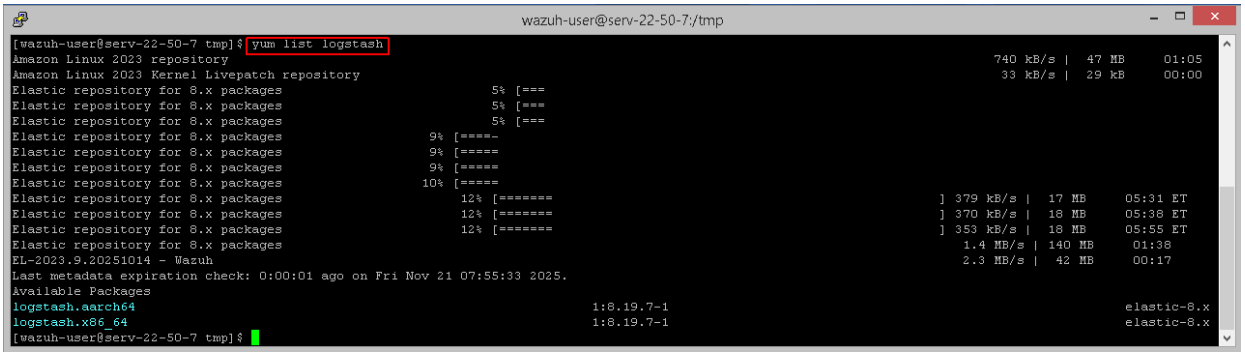



*Рис. 13.15. Доступність пакету Logstash*

Wazuh – Logstash – Splunk інтеграція потребує правильного формування подій у специфічному форматі, який сприймає Splunk HTTP Event Collector (HEC). Для цього в Logstash необхідно встановити спеціалізований плагін logstash-output-exec, що спрощує процес форматування та передачу даних. Встановлення плагіну та перевірка виконується командами:

***sudo /usr/share/logstash/bin/logstash-plugin install logstash-output-exec***

***sudo /usr/share/logstash/bin/logstash-plugin list | grep output-exec***


*Рис. 13.16. Встановлення плагіну logstash-output-exec*

На попередньому кроці до каталогу ***/etc/logstash*** встановлено пакет, а каталог ***/etc/logstash/conf.d*** призначений для створення pipeline, який ми описуємо у конфігураційному файлі ***wazuh-to-splunk.conf***

Вміст файлу конфігурації pipeline ***/etc/logstash/conf.d/wazuh-to-splunk.conf*** з коментарями приведено у додатку 3 цієї методики. Редагуємо файл за приведеним зразком, замінюючи значення ***token_Wazuh-HEC*** на власне, яке копіюємо з WEB-UI Splunk по кнопці Show у пункті Settings – Data Inputs – HTTP Event Collector (рис.13.6). Не забуваємо змінити IP-адресу Splunk-серверу (***192.168.50.11***) та ім'я хосту (***serv-22-50-7***), де розгорнуто Wazuh-Logstash.

У додатку 3 параметри, які необхідно змінити, виділені *таким кольором* ☺, а коментарі виділено *таким кольором* ☺

Конфігураційний файл Logstash описує pipeline, що складається з трьох основних секцій — input, filter та output.

Секція ***input*** відповідає за приймання подій від Filebeat. У конфігурації використовується вхідний плагін beats, який відкриває порт 5044 та приймає структуровані JSON-події, що надходять від модуля Wazuh-Agent.

Секція ***filter*** виконує попередню обробку отриманих даних перед їхньою передачею до Splunk. Для цього застосовуються:

- фільтр ruby — для формування Unix-мітки часу, необхідної HEC;
- фільтр mutate — для видалення службових полів, які не повинні потрапляти у Splunk;
- додатковий блок ruby — для побудови структурованого JSON-об'єкта у форматі, що відповідає вимогам Splunk HEC (time, host, index, sourcetype, event). Результат формування зберігається у полі hec_payload.

Секція ***output*** забезпечує відправлення події до Splunk. У ній використовується плагін exec, який викликає команду curl для виконання HTTPS-запиту до Splunk HTTP Event Collector. Дані передаються у вигляді JSON-рядка з поля hec_payload, сформованого у секції filter. Для контролю та налагодження передбачений додатковий вихід stdout з виведенням сформованого HEC-повідомлення у консоль.

Вмикаємо Logstash у автозапуск, запускаємо та переглядаємо статус (рис. 13.17):

***sudo systemctl enable logstash***
***sudo systemctl start logstash***
***sudo systemctl status logstash***

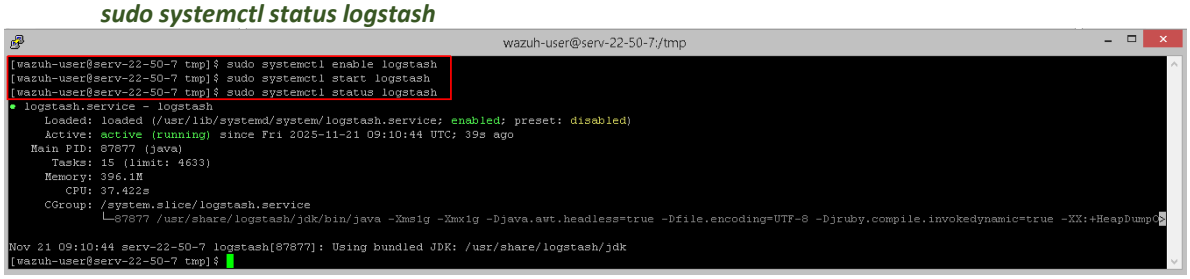



*Рис. 13.17. Вмикання автозапуску, запуск та перегляд статусу Logstash*

## Заміна стандартної версії Filebeat, запуск та тестування конвейера.

Стандартна версія Filebeat, що постачається у складі ВМ Wazuh appliance, може бути несумісною з актуальними модулями Logstash або використовуваними плагінами. Тому перед подальшим налаштуванням необхідно замінити вбудовану версію Filebeat на стабільну та сумісну редакцію. Імпорт GPG-ключа та додавання репозиторію Elastic 7.x було виконано на одному з попередніх кроків (рис. 13.14). Видаляємо вбудовану версію Filebeat:

***sudo dnf remove filebeat -y***
***sudo dnf install filebeat-7.17.9 -y***

Налаштовуємо Filebeat для відправки логів у Logstash. Для цього виконуємо бекап файлу конфігурації Filebeat та створюємо новий файл ***/etc/filebeat/filebeat.yml*** мінімальної конфігурації, приведеної у додатку 4, таблиця 13.1.

***sudo mv /etc/filebeat/filebeat.yml ~ filebeat.yml.bkp***

Filebeat міг зберегти старі дані (state), які не збігаються з новою версією. Щоб запобігти такій помилці конфігурації виконуємо видалення наступних файлів:

***sudo rm -rf /var/lib/filebeat/{registry,registry-filebeat,meta.json}***

Перевіряємо конфігурацію Filebeat:

***sudo filebeat test config***

Має бути ***Config OK***. Переконуємось, що Filebeat встановлено як системний сервіс:

***which filebeat***

Очікується ***/usr/bin/filebeat***. Якщо помилок не виникло, виконуємо повну перевірку конфігу:

***sudo /usr/share/logstash/bin/logstash --config.test_and_exit -f /etc/logstash/conf.d/wazuh-to-splunk.conf***

Має бути: ***Configuration OK*** (рис.13.18).

Перезапускаємо Logstash, дивимось статус (має бути активний) та перевіряємо чи «слухає» порт 5044. Перевірку порту виконуємо через 40-50 сек після перезапуску logstash. Існує висока ймовірність затримки.

***sudo systemctl restart logstash***
***sudo systemctl status logstash***
***sudo ss -tulpn | grep 5044***

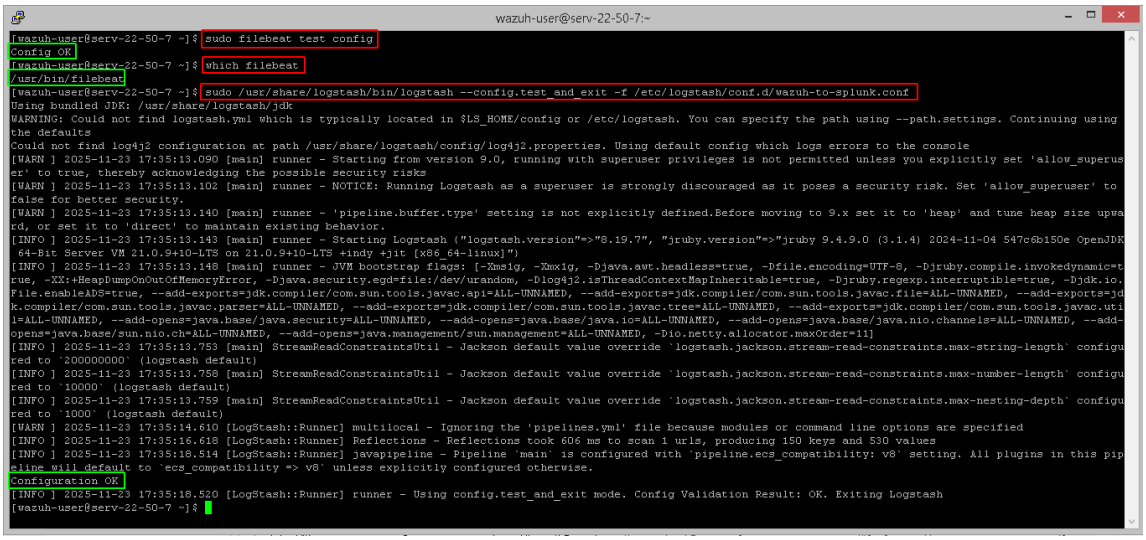



*Рис. 13.18. Перевірка коректності конфігурації Filebeat та Logstash*

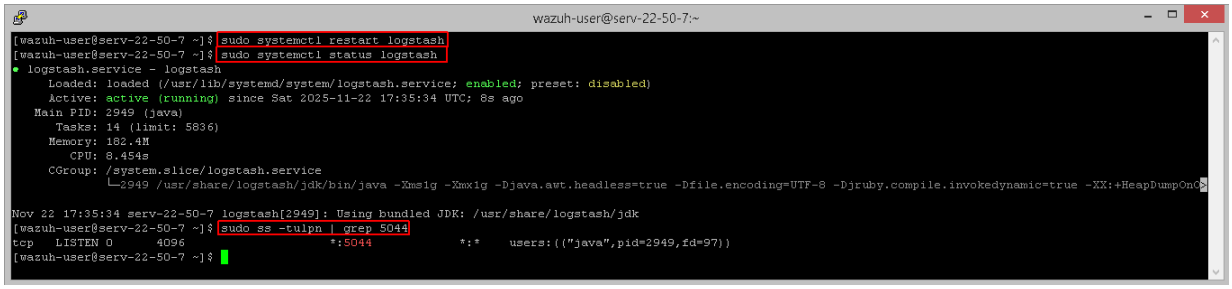



*Рис. 13.19. Перевірка коректності запуску Logstash*

Перезапускаємо Filebeat та перевіряємо його повний стан і логи:

***sudo systemctl restart filebeat***
***sudo systemctl status filebeat -l***

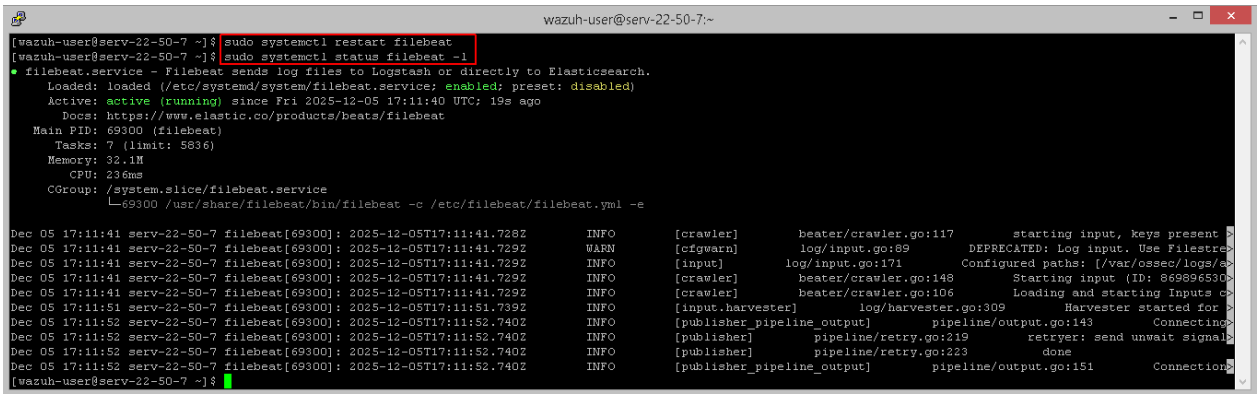



*Рис. 13.20. Перезапуск filebeat після редагування.*

Переглядаємо журнал у пошуках маркерів успішної роботи logstash (рис.13.21)

***sudo journalctl -u logstash -f***

Існує 3 групи маркерів, кожен з яких сигналізує про окремий етап успішної роботи:

- ✓ У журналі присутні рядки з [HEC-SENT] Це означає, що Logstash успішно сформував коректний HEC JSON.
- ✓ Присутні повідомлення {"text":"Success","code":0} Це означає, що Splunk HTTP Event Collector прийняв подію.
- ✓ Повторювані цикли HEC-SENT + Success при різних подіях. Це означає стабільну, безперебійну роботу інтеграції.

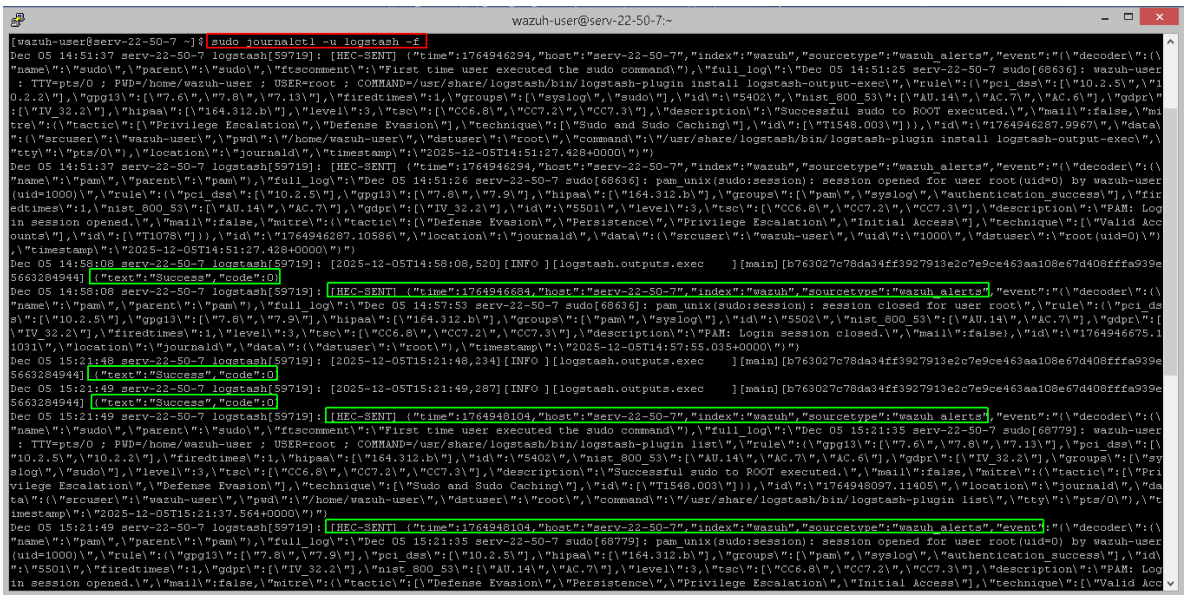



*Рис. 13.21. sudo journalctl -u logstash –f та «маркери» успішної роботи.*

## Перевірка прийому даних до індексу Wazuh у Splunk

Перевіряємо, чи Splunk бачить індекс wazuh. В меню Settings – Indexes шукаємо індекс wazuh, який має створитись автоматично під час першого успішного отримання подій Logstash – Splunk HEC.

Перевіряємо, що події надходять у Splunk через HEC. Відкриваємо меню Search & Reporting та виконуємо пошук ***index=wazuh | head 20*** (рис. 13.21). Результатом цієї операції мають бути JSON-події — інтеграція працює. Якщо результат порожній, повертаємося до попередніх кроків — дивимось лог pipeline.

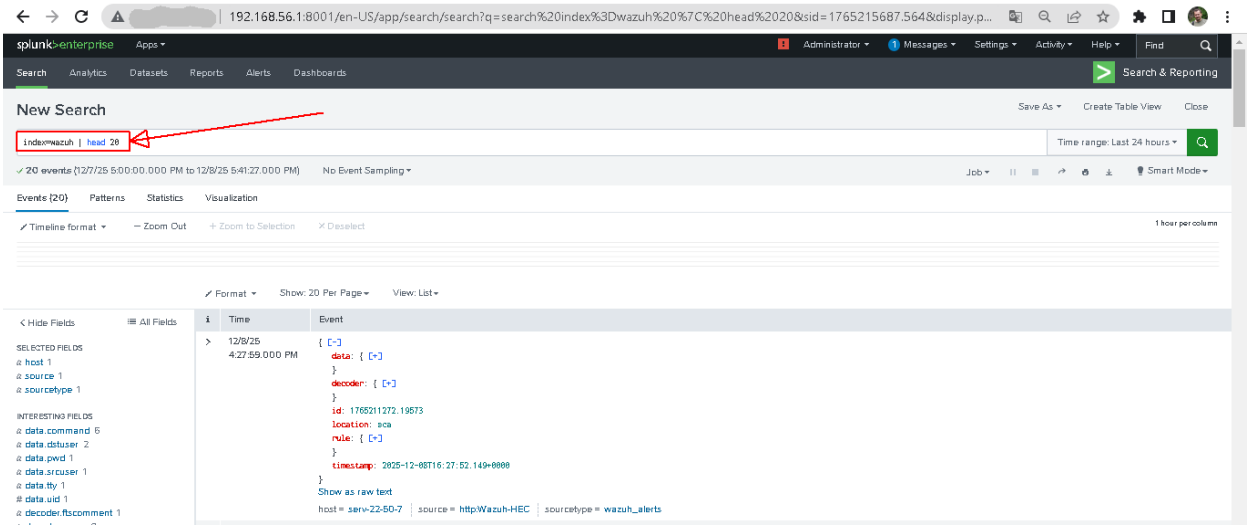



*Рис. 13.21. Меню Search & Reporting. Пошук index=wazuh | head 20*

## Завдання до лабораторної роботи

1. Увімкніть та налаштуйте Splunk HTTP Event Collector (HEC) на сервері Splunk (Serv-G-N-9), створіть окремий токен для прийому подій Wazuh та перевірте його доступність.
2. Налаштуйте інтеграцію Wazuh – Splunk через Universal Forwarder та HEC, забезпечивши передачу JSON-подій у Splunk.
3. Встановіть та налаштуйте Logstash на сервері Wazuh (Serv-G-N-7). Додайте власний pipeline для обробки логів Wazuh та перенаправлення їх у Splunk HEC.
4. Замініть стандартний Filebeat Wazuh. Запустіть конвейер: видаліть некоректні конфігурації Filebeat, застосуйте оптимізований filebeat.yml (додаток 4), та перевірте стабільність передачі подій у Logstash та Splunk.
5. Перевірте створення та роботу індексу wazuh у Splunk. У разі відсутності індексу — створіть його та виконайте пошукові запити index=wazuh | head 20.

Додаток 1.

## Повне чисте видалення Splunk Universal Forwarder Wazuh Appliance (Amazon Linux 2023)

***Процедура видалення Splunk Universal Forwarder приведена на випадок необхідності «відкату» та реконфігурування побудованої системи. У описаній методиці ця процедура не використовується.***

Зупиняємо UF і видаляємо systemd unit

```
sudo systemctl stop SplunkForwarder
sudo systemctl disable SplunkForwarder
sudo rm -f /etc/systemd/system/SplunkForwarder.service
sudo rm -f /usr/lib/systemd/system/SplunkForwarder.service
sudo systemctl daemon-reload
```

Видаляємо rpm-пакет UF. Якщо rpm скаже “not installed” — значить вже видалено, і це ок.

```
sudo rpm -e splunkforwarder
```

Видаляємо каталоги Splunk UF

```
sudo rm -rf /opt/splunkforwarder
sudo rm -rf /opt/splunk
```

Видаляємо створених користувачів. UF створює splunkfwd та splunker.

```
sudo userdel -r splunkfwd
sudo userdel -r splunker
```

Якщо скаже “user does not exist” — це нормально.

Додаток 2.

## Видалення сертифікату на Wazuh Appliance (Amazon Linux 2023)

***Процедура видалення сертифікату приведена на випадок необхідності його заміни. У описаній методиці ця процедура не використовується.***

Перед видаленням не потрібного alias, перевіряємо, що він існує

```
sudo /usr/share/logstash/jdk/bin/keytool -list \
 -keystore /usr/share/logstash/jdk/lib/security/cacerts \
 -storepass changeit | grep splunk-ca
```

Якщо рядок знайдено — alias точно є і видаляємо не потрібний сертифікат

```
sudo /usr/share/logstash/jdk/bin/keytool \
 -delete \
 -alias splunk-ca \
 -keystore /usr/share/logstash/jdk/lib/security/cacerts \
 -storepass changeit
```

Додаток 3.

## Приклад налаштування файлу конфігурації pipeline /etc/logstash/conf.d/wazuh-to-splunk.conf

```
input {
  beats {
    port => 5044
  }
}

filter {
  # 1 — Конвертація timestamp у Unix time (цілі секунди)
  ruby {
    code => '
      event.set("[@metadata][hec_time]", event.get("@timestamp").to_i)
    '
  }
  # 2 — Очищення зайвих кореневих полів
  mutate {
    remove_tag => ["_jsonparsefailure"]
    remove_field => ["@version", "tags", "ecs", "[agent]", "[input]", "[log]", "[predecoder]", "[manager]", "[host]", "type",
"path", "@timestamp"]
  }
  # 3 — Формування Splunk HEC JSON
  ruby {
    code => '
      require "json"
      clean_event_json = event.to_json
      hec_payload = {
        "time" => event.get("[@metadata][hec_time]"),
        "host" => "serv-22-50-7",
        "index" => "wazuh",
        "sourcetype" => "wazuh_alerts",
        "event" => clean_event_json
      }
      event.set("hec_payload", hec_payload.to_json)
    '
  }
}

output {
  if [hec_payload] {
    # 4 — Відправка через exec → curl
    exec {
      command => "curl -k -s -X POST -H 'Authorization: Splunk token_Wazuh-HEC' -H 'Content-Type: application/json'
'https://192.168.50.11:8088/services/collector/event' -d '%{[hec_payload]}'"
    }
    # 5 — Логування у stdout для аудиту
    stdout {
      codec => line {
        format => '[HEC-SENT] %{[hec_payload]}'
      }
    }
  }
}
```

Додаток 4.

## Приклад налаштування Filebeat для відправки логів у Logstash. Файл конфігурації Filebeat /etc/filebeat/filebeat.yml

Таблиця 13.1.

Конфігураційний файл після редагування

```
# Wazuh - Filebeat configuration file

output.logstash:
  hosts: ["127.0.0.1:5044"]

filebeat.inputs:
  - type: log
    enabled: true
    paths:
      - /var/ossec/logs/alerts/alerts.json
    json.keys_under_root: true
    json.add_error_key: true

setup.template.json.enabled: true
setup.template.json.path: '/etc/filebeat/wazuh-template.json'
setup.template.json.name: 'wazuh'
setup.ilm.overwrite: true
setup.ilm.enabled: false

logging.level: info
logging.to_files: true
logging.files:
  path: /var/log/filebeat
  name: filebeat
  keepfiles: 7
  permissions: 0644

logging.metrics.enabled: false
```

Додаток 5.

## Імпорт CA Splunk у Logstash

У додатку описана процедура додавання кореневого сертифікату CA Splunk (CN=SplunkCommonCA) у Java truststore Logstash. Підключаємось по SSH до серверу Splunk Serv-G-N-9 [192.168.50.11]

Зберігаємо сертифікати, які віддає порт HEC (8088) у файлі ***/tmp/splunk-full-chain.pem***:

```
openssl s_client -connect 192.168.50.11:8088 -showcerts </dev/null \
 | sed -n '/BEGIN CERTIFICATE/,/END CERTIFICATE/p' > /tmp/splunk-full-chain.pem
```

Розбиваємо ланцюжок збережених сертифікатів на окремі файли (рис.13.14):

```
csplit -f cert- -b "%02d.pem" /tmp/splunk-full-chain.pem "/-----BEGIN CERTIFICATE-----/" "{*}"
```

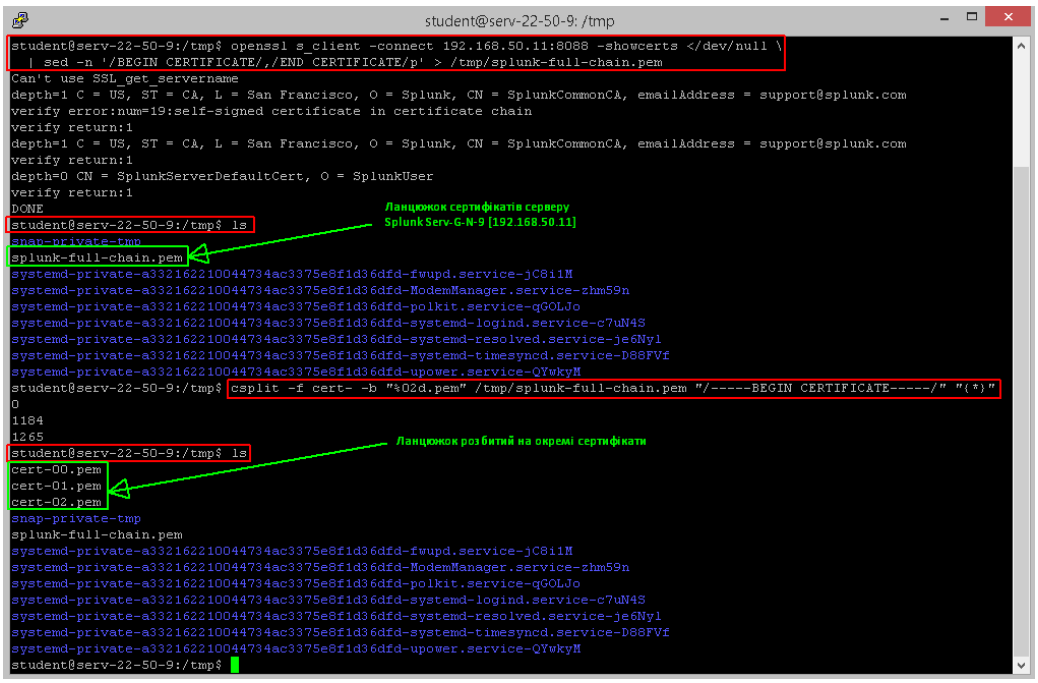



*Рис. 13.14. Отримання файлів сертифікатів серверу Splunk Serv-G-N-9 [192.168.50.11]*

В результаті описаній дій (рис.13.14) у каталозі /tmp серверу збережено три файли сертифікатів. Визначаємо, кореневий сертифікат по вмісту поля subject у виводі команд:

```
openssl x509 -in cert-01.pem -noout -text
openssl x509 -in cert-02.pem -noout -text
```

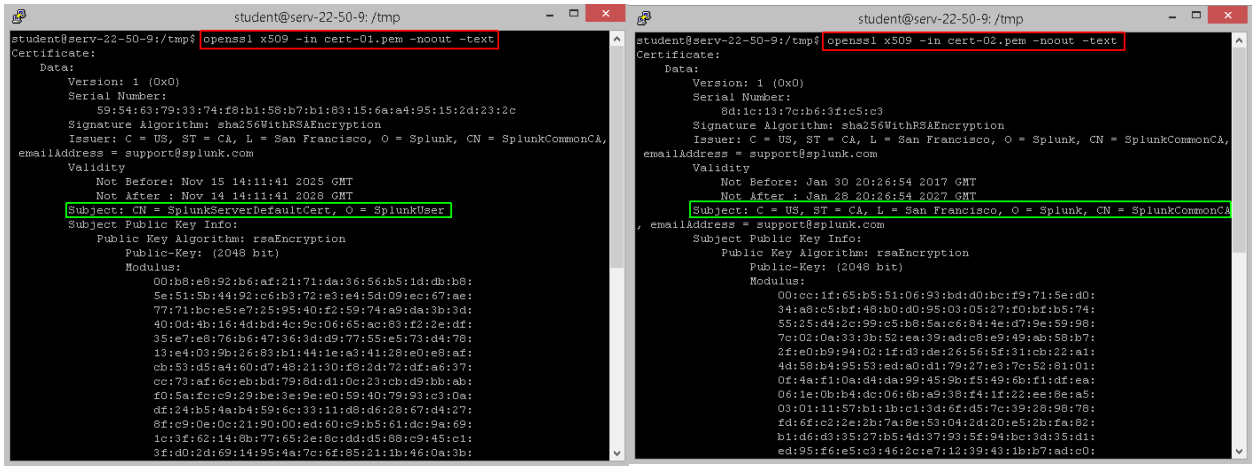



*Рис. 13.15. Перегляд інформації про сертифікати серверу Splunk Serv-G-N-9 [192.168.50.11]*

Перегляд інформації про сертифікати дає наступне:

- ***/tmp/cert-01.pem*** — серверний сертифікат Splunk
- ***/tmp/cert-02.pem*** — кореневий CA Splunk (SplunkCommonCA). Саме його треба імпортувати.

Перейменовуємо кореневий CA:

```
mv /tmp/cert-02.pem /tmp/splunk-ca.pem
```

Копіюємо Splunk CA на Logstash-сервер. Замініть адресу 192.168.50.9 на адресу свого Serv-G-N-7 у рядку та виконайте на Splunk-сервері:

***scp -o StrictHostKeyChecking=no /tmp/splunk-ca.pem wazuh-user@192.168.50.9:/tmp/splunk-ca.pem***

Далі користувач вводить пароль wazuh. Це – стандартний пароль для Wazuh Appliance (рис.13.16).

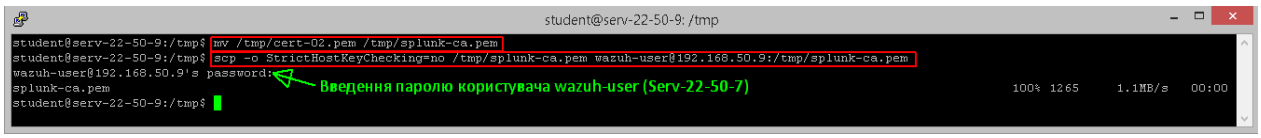



*Рис. 13.16. Копіювання кореневого сертифікату CA Splunk (CN=SplunkCommonCA) на сервер Serv-22-50-7*

Залишаємо Splunk-сервер та переходимо у SSH-сесію на Logstash-сервері, він же Wazuh Appliance, він же Serv-G-N-7 (192.168.50.9) ☺. Створюємо каталог та переміщуємо до нього завантажений сертифікат:

***sudo mkdir -p /etc/logstash/certs***

***sudo mv /tmp/splunk-ca.pem /etc/logstash/certs/splunk-ca.pem***

***sudo chmod 644 /etc/logstash/certs/splunk-ca.pem***

Перевіряємо місце розташування truststore та імпортуємо Splunk CA у Java truststore Logstash

***sudo find /usr/share/logstash -name cacerts***

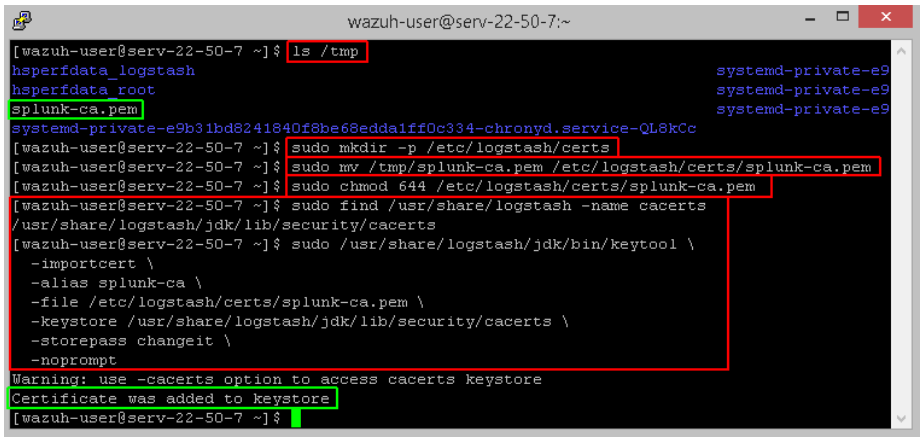



*Рис. 13.17. Імпорт кореневого сертифікату CA Splunk на сервері Wazuh/Logstash*

Імпортуємо CA:

***sudo /usr/share/logstash/jdk/bin/keytool ***
***-importcert ***
***-alias splunk-ca ***
***-file /etc/logstash/certs/splunk-ca.pem ***
***-keystore /usr/share/logstash/jdk/lib/security/cacerts ***
***-storepass changeit ***
***-noprompt***

Logstash довіряє Splunk.

Додаток 6.

## Запуск / перезапуск служб Wazuh (Manager + Dashboard + Indexer)

Перевірка стану всіх сдужб Wazuh

***sudo systemctl status wazuh-manager --no-pager -l***
***sudo systemctl status wazuh-dashboard --no-pager -l***
***sudo systemctl status wazuh-indexer --no-pager -l***

Коректний перезапуск (рекомендований порядок) Indexer (OpenSearch) – Manager – Dashboard:

***sudo systemctl restart wazuh-indexer***
***sudo systemctl restart wazuh-manager***
***sudo systemctl restart wazuh-dashboard***

Якщо Wazuh Manager не стартує (timeout / зависання), зупиняємо Manager, перевіряємо що залишилось "живим" та запускаємо Manager заново

***sudo systemctl stop wazuh-manager***
***ps aux | grep wazuh | head -n 50***
***sudo systemctl start wazuh-manager***
***sudo systemctl status wazuh-manager --no-pager -l***

Діагностуємо помилку запуску Manager. Якщо старт з помилкою:

***sudo journalctl -xeu wazuh-manager.service --no-pager | tail -n 80***

А також основний лог Wazuh:

***sudo tail -n 80 /var/ossec/logs/ossec.log***

Якщо нічого не допомагає, виконуємо повний "reset-перезапуск" всього Wazuh (жорсткий сценарій)

***sudo systemctl stop wazuh-dashboard***
***sudo systemctl stop wazuh-manager***
***sudo systemctl stop wazuh-indexer***
***sudo systemctl start wazuh-indexer***
***sudo systemctl start wazuh-manager***
***sudo systemctl start wazuh-dashboard***

## Корисні посилання


- Wazuh. Splunk integration.
https://documentation.wazuh.com/current/integrations-guide/splunk/index.html

- Integrating Wazuh and Splunk
https://www.linkedin.com/pulse/integrating-wazuh-splunk-vaibhav-karayat-p2gnc

- Integrations guide: Elastic, OpenSearch, Splunk, Amazon Security Lake
https://documentation.wazuh.com/current/integrations-guide/index.html

- wazuh / wazuh-splunk
https://github.com/wazuh/wazuh-splunk

- IBM Storage Defender. Splunk® Configuration Guide
https://www.ibm.com/docs/en/SSDR5G6_prod/pdf/splunk_config_guide.pdf