Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти

на тему

“Дослідження та програмна реалізація системи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури”

Виконав здобувач вищої освіти
ІІ курсу, групи КІ-24М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Главнов С.І.
«____» _____ 2025 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Смірнов С.А.
«____» _____ 2025 р.
Рецензент _____
_____

м. Кропивницький

АНОТАЦІЯ

Главнов С.І. Дослідження та програмна реалізація системи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Метою розробки є дослідження та програмна реалізація системи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Об'єктом дослідження є процес мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Предметом дослідження є методи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Методи дослідження базуються на методах захисту інформації у мережі, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Visual C++.

Ключові слова: комп'ютерна інженерія, SIEM, аналізу загроз, корпоративна IT-інфраструктура

ABSTRACT

Hlavnov S.I. Research and software implementation of a network SIEM system for analyzing security threats to corporate IT infrastructure. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this final qualification work for the second (master's) level of higher education, software has been developed that is intended for a network SIEM system for analyzing security threats to corporate IT infrastructure.

The purpose of the development is the research and software implementation of a network SIEM system for analyzing security threats to corporate IT infrastructure.

The object of the research is the network SIEM process for analyzing security threats to corporate IT infrastructure.

The subject of the research is network SIEM methods for analyzing security threats to corporate IT infrastructure.

The research methods are based on methods of information protection in the network, methods of mathematical statistics, methods of software development.

The result of the work is a software implementation of a network SIEM system for analyzing security threats to corporate IT infrastructure.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software are provided.

The program can be used on a PC with Windows 10/11.

The program was developed in the Visual C++ environment.

Keywords: computer engineering, SIEM, threat analysis, corporate IT infrastructure

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ЛОМ	–	локальна обчислювальна мережа
MME	–	міжмережні екрани
ATM	–	асинхронний режим передачі
BSD	–	адаптована для Internet реалізація операційної системи UNIX
ICMP	–	міжмережний протокол управляючих повідомлень
IP	–	Internet Protocol – міжмережний протокол
NFS	–	мережна файлова система
PPP	–	протокол передачі від точки до точки
RFC	–	опис набору протоколів Internet
RPC	–	віддалений виклик процедури
SLIP	–	міжмережний протокол для послідовного каналу
SMTP	–	Simple Mail Transfer Protocol – простий протокол передачі пошти
TCP	–	Transmission Control Protocol – протокол управління передачею
UDP	–	User Datagram Protocol – протокол користувальницьких датаграм
UNIX	–	багатозадачна операційна система
UTP	–	незахищена вита пара
URL	–	уніфікований покажчик інформаційного ресурсу

ВСТУП

Актуальність теми. Мережі вашої організації є основною інфраструктурою для ваших інформаційно-технологічних (IT) систем, операційних технологій (OT) та промислових систем управління (ІКС). Тому важливо забезпечити безпеку вашої мережевої інфраструктури, щоб захистити вашу організацію від порушень, вторгнень та інших кіберзагроз. Мережеве ведення журналу та моніторинг подій безпеки допоможуть вам:

- захистіть свою мережеву інфраструктуру;
- визначити індикатори компрометації (IoCs);
- своєчасно вживати коригувальних заходів;
- мінімізувати вплив у разі виникнення інциденту безпеки.

Рішення SIEM об'єднує функції моніторингу та ведення журналу. Термін SIEM вперше був введений Gartner у 2005 році для опису комбінації наступних підходів:

- управління інформацією безпеки (SIM), що стосується діяльності, пов'язаної зі збором даних, таких як файли журналів, з кількох джерел у централізоване сховище;
- управління подіями безпеки (SEM), що стосується діяльності, пов'язаної з моніторингом та аналізом конкретних подій безпеки в режимі реального часу, які можуть бути тривожними сигналами.

Традиційно, SIEM-рішення здебільшого пропонували захист для локальних середовищ з обмеженими джерелами даних та можливостями. SIEM-рішення еволюціонували, і SIEM наступного покоління пропонують більше можливостей для боротьби з передовими кіберзагрозами та обробки величезних обсягів даних. Зараз доступно багато хмарних SIEM-рішень, які можуть захищати активи як локально, так і в хмарі.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

– Дослідження системи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

– Програмна реалізація системи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Об'єктом дослідження є процес мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Предметом дослідження є методи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Методи дослідження базуються на методах захисту інформації у мережі, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

– Розроблено вітчизняний продукт мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями

параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2025 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

SIEM-рішення – це набір інструментів та сервісів, які збирають, агрегують та аналізують обсяги даних з різних джерел у режимі реального часу. Деякі основні можливості SIEM включають:

– агрегування даних з багатьох джерел, таких як користувачі, мережеві пристрої, програми, кінцеві точки та хмарна інфраструктура;

– моніторинг та аналіз подій у реальному часі та історичних подій;

– нормалізація або переформатування даних журналу у стандартний формат для полегшення аналізу;

– співвіднесення подій безпеки, що мають спільні атрибути;

– сприяння кореляції та аналізу аудиторських записів (наприклад, шляхом співвіднесення подій з результатами сканування на вразливості);

– виявлення IoC, зібраних динамічно з каналів загроз;

– надсилання сповіщень та оповіщень у разі виявлення реальних або потенційних загроз;

– управління сортуванням тривог;

– архівування журналів для полегшення кореляції даних з плином часу для розслідування інцидентів та дотримання вимог;

– перевірка криптографічної цілісності та перевірка журналів, щоб визначити, чи були вони підроблені.

1.2 Область застосування

На сьогодні головним споживачем SIEM-продуктів є фінансовий сектор. Причин тому трохи. По-перше, банки працюють із конфіденційною інформацією,

тому у випадку виникнення інцидентів важливо знати, хто й коли допустив витік, чи була вона навмисним або випадкової й т.д. По-друге, банкам необхідно регулярно проводити аудити відповідності. По-третє, зовнішні аудитори іноді розглядають наявність впровадженої SIEM-системи як додатковий плюс.

Ще одна категорія споживачів – великі підприємства, у яких щодня генерується безліч подій різної властивості, відстежити які просто фізично неможливо. Тому керівництво хоче «тримати руку на пульсі», щоб більш оперативно відреагувати на можливі проблеми.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Відповідно до звіту аналітичного агентства Frost & Sullivan за 2025 рік, світовий ринок SIEM-рішень із 2025 по 2026 рік зросте практично вдвічі. Аналітики впевнені, що SIEM-системи усе більше стають значимими IT-інструментами сучасних підприємств. У чималому ступені на їхню популярність впливають такі важливі IT-тренди як консолідація й віртуалізація. Адже SIEM дозволяють централізувати зберігання інформації про події, що відбуваються в IT-інфраструктурі.

Ріст ринку привів до його часткового переформатування, викликаному безліччю голосних злиттів. В 2025 році компанія HP оголосила про придбання Arcsight, інвестуючи значні ресурси в продукти цього вендора. На зміну застарілій лінійці IBM Tivoli прийшов продукт Q1 Radar. McAfee в 2025 році поглинула Nitro Security і підсилила свої SIEM-продуктів функціоналом SEM. Виробник Tibco придбав Loglogic, додав у функціонал SIEM нові аналітичні можливості.

На сьогодні лідерами ринку є HP з лінійкою продуктів ArcSight, IBM Q1 Radar, Symantec із продуктом Security Information Manager, McAfee Nitro і RSA Envision. На інших виробників, включаючи Tibco Loglogic і Splunk, доводиться частка всього в кілька відсотків.

За даними аналітичного агентства Gartner, протягом декількох років одним з безумовних лідерів на світовому ринку SIEM-рішень є ArcSight компанії Hewlett Packard Enterprise. Цей продукт, споконвічно розроблений для потреб силових відомств США, трохи пізніше було дозволено використовувати й

комерційним підприємствам. Компанія ArcSight заснована в 2000 році, а в 2010-му її придбала HP, що і займається розвитком рішення ArcSight, просуваючи його на ринках США, Європи й інших країн. Можливості HPE ArcSight у плані збору, аналізу й візуалізації подій в області інформаційної безпеки добре відомі російським організаціям, крім того, навколо цього рішення вже давно сформувалася широка партнерська мережа системних інтеграторів, з успіхом проекти, що реалізує, по впровадженню ArcSight.

Компанія Hewlett Packard Enterprise, правонаступниця HP, продовжує розвивати ArcSight, удосконалюючи вже існуючі компоненти й розробляючи нові. Одна з новинок – ArcSight User Behavior Analytics – виявляє аномалії на основі аналізу поводження користувачів і доповнює традиційну кореляцію, що є базовою функцією ArcSight. Традиційний кореляційний механізм працює на основі правил, що фіксують позаштатні дії користувачів. При виявленні інциденту він або сповіщає адміністратора ІБ, або автоматично виконує задану операцію: запускає скрипт, блокує користувача й т.д. На додаток до цьому поведінковий механізм User Behavior Analytics повідомляє про інциденти, схема й ознаки яких ще не відомі адміністраторам.

При розробці User Behavior Analytics був використаний принцип самонавчання на повсякденних діях користувачів. Надалі активність, що не укладається в профіль нормального поводження, фіксується як підозріла відповідно до розрахованого рівня ризику. Як приклад такого поводження можна привести зміна звичних дій користувача, що, посилаючи у звичайні дні не більше десятка електронних листів, раптом відправив 100 або 1000 повідомлень. Для кожного користувача в User Behavior Analytics автоматично формується індивідуальний поведінковий профіль, і при виході за його рамки система відправляє відповідний сигнал. Такий підхід спрощує роботу адміністраторів ІБ, дозволяючи їм реагувати тільки на важливі інциденти й події.

Ще одне нове рішення на платформі HPE ArcSight – DNS Malware Analytics. Воно аналізує DNS-трафік і забезпечує повну видимість IT-

інфраструктури, що допомагає виявити мережні уразливості ще до того, як ними скористаються зловмисники. Ідея аналізу DNS-трафіку с метою виявлення зловмисної активності зародилася в дослідницькому підрозділі HP Labs чотири роки тому, і от уже півтора року створене його фахівцями рішення тестується в компанії HP, здійснюючи пошук заражених машин, які оказалися під керуванням зловмисників. Складна гетерогенна мережа й величезна кількість співробітників – у таких непростих умовах проходили польові випробування HPE ArcSight DNS Malware Analytics.

Сьогодні це рішення доступно й російським замовникам. Принцип його роботи наступний: заражена машина намагається щось завантажити або передати за межі корпоративної мережі; ці дії викликають спрацьовування профілів негативного поводження, і адміністратори інформаційної безпеки сповіщаються про що відбувається, у тому числі про тип злобливої троянської програми, який заражений конкретний комп'ютер. Оскільки система аналізує винятково DNS-трафік, вона легко інтегрується з будь-якими мережами, і здобувати дороге мережне встаткування не потрібно. Як правило, традиційні засоби захисту охороняють периметр мережі (DMZ), але не весь заражений трафік виходить за її межі, адже співробітник, що працює на своєму ноутбуці, може перебувати в будь-якому місці (будинку, в аеропорті, у кафі й т.д.). Аналіз DNS-трафіку дозволяє виявити й убезпечити корпоративну мережу від подібного роду заражених пристроїв. Нарешті, DNS-трафік простіше агрегувати: досить зконфігурувати інфраструктуру таким чином, щоб копія трафіку концентрувалася в певнім місці. Компанія HPE забезпечує коректну роботу й актуалізацію сигнатур, що позначають зараження DNS-трафіку.

Розроблювачі HPE ArcSight відслідковують новітні тенденції ринку й зміни в перевагах замовників. Щорічно в США проводяться всесвітні конференції користувачів ArcSight, де обговорюються побажання по реалізації нових можливостей, а також анонсуються нові модулі й функції. З тією же регулярністю такі користувальницькі конференції проводяться й у Києві.

Особливий інтерес при відборі доповідей викликають проекти, при реалізації яких за допомогою ArcSight удалося вирішити складні або нетривіальні завдання в області інформаційної безпеки.

Крім розповіді про нові компоненти HPE ArcSight, хотілося б нагадати й про ключові «стовпи» рішення. Насамперед, це Security Data Platform – набір функцій, відповідальних за збір і класифікацію подій, а також за їхнє зберігання й архівування. До складу продукту поряд з коннекторами, що виконують вилучений збір подій, входять логгер, що забезпечує їхнє зберігання, регулярний пошук і аналіз, а також безкоштовний Management Center, що здійснює відновлення коннекторів, резервне копіювання всієї інфраструктури по зборі подій і моніторинг. Подібні можливості підходять для тих замовників, яким потрібно одержувати інформацію про інциденти ІБ не в реальному часі, а у вигляді звітів за тиждень, місяць і т.д. Ліцензується Security Data Platform по обсязі оброблених даних, тобто отриманих коннекторами із пристроїв замовника. Сьогодні коннектори, що входять до складу Security Data Platform, підтримують як джерела подій понад 350 різні інформаційні системи від різних виробників, однак за допомогою безкоштовного SDK замовники або партнери можуть самостійно написати такий коннектор для будь-якої системи. Модуль SDP є повноцінним продуктом, тому підприємства, що не бідують у додаткових можливостях, здобувають тільки його.

Інший базовий компонент ArcSight – Enterprise Security Manager – стежить за подіями ІБ у реальному часі. Цей компонент необхідний тим, кому потрібна моментальна реакція на інциденти. Ліцензується Enterprise Security Manager виходячи з кількості подій, оброблених за секунду часу. Мінімальний поріг ліцензії – 250 подій у секунду. Для порівняння досить відзначити, що в компанії Hewlett Packard Enterprise обробляється 40-50 тисяч подій у секунду. Усередині ESM перебуває модуль Threat Detector, що виявляє погрози не по заздалегідь запрограмованих сигнатурах, а на основі аналізу незвичайного поводження й повторюваної активності користувачів або додатків. Для представників

середнього або малого бізнесу пропонується спеціальна редакція ArcSight ESM Express? теж повністю самостійний продукт. Від «великого» ESM він відрізняється, мабуть, лише відсутністю декількох функцій, таких, наприклад, як підтримка відказостійкого кластера.

Нарешті, ArcSight Threat Central, інтерактивна база знань про погрози, дозволяє обмінюватися відомостями про способи їхнього виявлення й ліквідації, а на порталі MarketPlace утримуються правила й ознаки виявлених погроз (пакети безпеки) і додаткові додатки. Розроблювачі з HPE сподіваються, що до формування таких пакетів безпеки й створенню додаткових додатків підключаться й партнери компанії.

Звичайно, всі проблеми в області інформаційної безпеки, як зовнішні, так і внутрішні, викликані насамперед людським фактором. Причому це може бути не якась зловмисна дія, а проста неуважність і зневага правилами й регламентами. Найчастіше співробітники ІБ-відділів не обертають особливої уваги на невеликі інциденти, поки не трапиться НП.

У той же час SIEM-рішення, такі як HPE ArcSight, з одного боку, допомагають тримати ситуацію під контролем, а з іншого боку – вчасно сповіщають про потенційні проблеми, які, якщо не звернути на них уваги, можуть привести до катастрофічних наслідків.

HP ArcSight

Можливості:

– Лог-менеджмент.

– Інцидент-менеджмент.

– Кореляція.

– Оповіщення про інциденти.

– Можливість установки на сервер віртуалізації.

– Передвстановлений контент.

– Можливість установки модуля виявлення поведінкових моделей і закономірностей Threat Detector.

– Додаткові пакети по контролю виконання вимог міжнародних стандартів.

– Модуль контролю дій користувачів IdentityView.

– Виявлення шахрайських операцій (Fraud Detection).

Tibco Loglogic

LogLogic SIEM є модульною системою, що складається з наступних частин:

– LogLogic MX: готове рішення для малого й середнього бізнесу.

– LogLogic ST: довгострокове зберігання подій.

– LogLogic SEM: кореляція й оповіщення про події ІБ.

– LogLogic LX: моментальний пошук подій.

– Регуляторна відповідність: пакети для забезпечення регуляторної відповідності PCI DSS, ISO 27001/ ISO 27002, ITIL, COBIT, SOX і ін.

– Database Security Manager. Активний моніторинг і виявлення уразливостей баз даних.

Можливості:

– Збір подій з більш ніж 340 джерел.

– Кореляція подій і оповіщення в режимі реального часу.

– Моментальний пошук за даними за останні 90 днів.

– Зберігання й пошук за даними за 10 років.

– Передналаштовані звіти й правила.

– Інтеграція із зовнішніми середовищами.

McAfee NitroSecurity

В основі SIEM-системи від McAfee лежить рішення Enterprise Security Manager, що здійснює збір, кореляцію, оцінку й розподіл пріоритетів подій безпеки.

Будучи частиною архітектури Security Connected, рішення McAfee Enterprise Security Manager тісно інтегроване із програмним забезпеченням McAfee ePolicy Orchestrator (McAfee ePO), рішенням McAfee Risk Advisor, і

технологією Global Threat Intelligence, забезпечуючи контекст, необхідний для автономного й гнучкого керування погрозами безпеки.

Склад і можливості рішення:

– McAfee Enterprise Security Manager.

– Технологія McAfee Global Threat Intelligence for Enterprise Security Manager (ESM), призначена для роботи з «великими даними в сфері безпеки», дозволяє використовувати результати роботи McAfee Labs безпосередньо для моніторингу безпеки.

– McAfee Enterprise Log Manager автоматизує керування всіма типами журналів і їхній аналіз, включаючи журнали подій Windows, журнали баз даних, журнали додатків і системні журнали (Syslogs).

– McAfee Advanced Correlation Engine виконує моніторинг даних у режимі реального часу, дозволяючи одночасно використовувати системи кореляції подій як засновані на правилах, так і не використовують правил з метою виявлення ризиків і погроз до їхнього виникнення.

– McAfee Application Data Monitor виконує дешифрування повного сеансу додатка до Рівня 7, забезпечуючи комплексний аналіз всієї інформації – від використовуваних протоколів і цілісності сеансу до безпосереднього вмісту додатка, такого як текст електронного листа або вкладень до нього.

– McAfee Database Event Monitor for SIEM забезпечує детальну реєстрацію в журналі безпеки транзакцій у базах даних.

– McAfee Event Receiver збирає дані подій і журналів сторонніх постачальників.

Symantec Security Information Manager (SSIM)

Система автоматизації виявлення й реагування на інциденти інформаційної безпеки контролю переміщення конфіденційної інформації, побудована на базі рішення Symantec SIM складається з наступних компонентів:

– Сервер Symantec SIM.

– Об'єкти спостереження.

– Колектори.

– Агенти.

– Symantec Global Intelligence Network.

Symantec Global Intelligence Network – глобальна мережа, що використовує пастки для виявлення зловмисної активності.

Велика увага приділяється аналізу нетрадиційної активності по різних портах/протоколам.

Досліджуються додатки, що використовують ці порти/протоколи, перевіряється, чи не з'являлися нові уразливості в цих додатках, аналізується ймовірність використання додатків у зловмисних цілях.

Також створюється статистика найбільш атакуючих і систем, що атакуються.

Вся ця інформація переробляється в правила й використовується в Symantec SIM при аналізі й кореляції подій.

– Особливості системи:

– Централізований збір, зберігання, аналіз журналів безпеки.

– Виявлення інцидентів у режимі реального часу.

– Визначення пріоритетів інцидентів.

– Автоматизація контролю над процесом виправлення інцидентів.

– Створення звітів про дотримання нормативних вимог і аудита

RSA Security Analytics

RSA Security Analytics являє собою платформу безпеки нової формації, що забезпечує аналіз усього мережного трафіку й журналів подій організації.

Особливості системи:

– Лінійна масштабованість як по обсягах збираємих даних, так і по швидкості їхньої обробки.

– Високопродуктивна система кореляції подій, що дозволяє аналізувати величезні потоки подій.

– Реконструкція мережних сесій і аналіз їхнього вмісту.

– Можливість аналізу й реконструкції мережних сесій для довільних TCP/IP протоколів.

– Уніфіковане подання для мережних сесій і даних журналів подій.

– Нові підходи до розслідувань і аналізу більших обсягів різнорідних даних, що дозволяють відсівати малозначиму інформацію, швидко виявляти підозрілу активність і відновлювати вміст підозрілої активності.

– Можливість аналізу на шкідливість всіх файлів, що виконуються, вступників у мережу організації несигнатурними методами.

– Використання при аналізі даних бізнес-контексту, що враховує цінність інформаційних активів організації.

– зіставлення в реальному часі даних, що збираються в інфраструктурі з найбільш свіжими даними про погрози – як сторонніми й так власними.

Splunk

Splunk Enterprise – це провідна на ринку платформа для операційної аналітики.

Здатна здійснювати моніторинг і аналіз всіх дій, від відвідувань веб-сайтів і транзакцій до мережних операцій і зареєстрованих викликів.

Особливості системи:

– Збір даних з вилучених джерел за допомогою модуля Splunk Forwarder

– Кореляція складних подій, що охоплюють безліч різнорідних джерел даних у середовищі.

– Масштабування для збору й індексації сотень терабайтів даних у день Можливість комбінування даних із традиційних реляційних БД і Hadoop для наступного аналізу.

– Рольова модель доступу до даних.

OSSIM

Open Source Security Information Management (OSSIM) є безкоштовною SIEM-системою.

Проект розвивається з 1996 року.

Є комерційний варіант за назвою AlienVault Unified Security Management.

Особливості системи:

– Безкоштовна.
– Убудовані правила кореляції (більше 1600).
– Доступно більше 150 звітів.
– Відповідність PCI, HIPAA, SOXGPG13.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для реалізації програми мною була використана мова програмування Visual C++. У зв'язку з тим, що сьогодні рівень складності програмного забезпечення дуже високий, розробка додатків Windows з використанням тільки якої-небудь мови програмування значно утрудняється. Програміст повинен затратити масу часу на рішення стандартних завдань по створенню багатовіконного інтерфейсу. Реалізація технології зв'язування й вбудовування об'єктів – OLE – зажадає від програміста ще більш складної роботи. Щоб полегшити роботу програміста практично всі сучасні компілятори з мови C++ містять спеціальні бібліотеки класів. Такі бібліотеки містять у собі практично весь програмний інтерфейс Windows і дозволяють користуватися при програмуванні засобами більш високого рівня, чим звичайні виклики функцій. За рахунок цього значно спрощується розробка додатків, що мають складний інтерфейс користувача, полегшується підтримка технології OLE і взаємодія з базами даних. Сучасні інтегровані засоби розробки додатків Windows дозволяють автоматизувати процес створення додатка. Для цього використовуються генератори додатків. Програміст відповідає на питання генератора додатків і визначає властивості додатка – чи підтримує воно багатовіконний режим, технологію OLE, тривимірні органи керування, довідкову систему. Генератор додатків, створить додаток, що відповідає вимогам, і надасть вихідні тексти. Користуючись їм як шаблоном, програміст зможе швидко розробляти свої

додатки. Подібні засоби автоматизованого створення додатків включені в компілятор Microsoft Visual C++ і називаються MFC AppWizard. Заповнивши кілька діалогових панелей, можна вказати характеристики додатка й одержати його тексти, постачені великими коментарями. MFC AppWizard дозволяє створювати одновіконні й багатовіконні додатки, а також додатки, що не мають головного вікна, – замість нього використовується діалогова панель. Можна також включити підтримку технології OLE, баз даних, довідкової системи. Звичайно, MFC AppWizard не всесильний. Прикладну частину додатка програмістові прийдеться розробляти самостійно. Вихідний текст додатка, створений MFC AppWizard, стане тільки основою, до якої потрібно підключити інше. Але працюючий шаблон додатка – це вже половина всієї роботи. Вихідні тексти додатків, автоматично отриманих від MFC AppWizard, можуть становити сотні рядків тексту. Набір його вручну був би дуже стомлюючий. Потрібно відзначити, що MFC AppWizard створює тексти додатків тільки з використанням бібліотеки класів MFC (Microsoft Foundation Class library). Тому тільки вивчивши мову C++ і бібліотеку MFC, можна користуватися засобами автоматизованої розробки й створювати свої додатки в найкоротший термін. Як уже згадувався, MFC – це базовий набір (бібліотека) класів, написаних мовою C++ і призначених для спрощення й прискорення процесу програмування для Windows. Бібліотека містить багаторівневу ієрархію класів, що нараховує близько 200 членів. Вони дають можливість створювати Windows-додатки на базі об’єктно-орієнтованого підходу. З погляду програміста, MFC являє собою каркас, на основі якого можна писати програми для Windows. Бібліотека MFC розроблялася для спрощення завдань, що стоять перед програмістом. Як відомо, традиційний метод програмування під Windows вимагає написання досить довгих і складних програм, що мають ряд специфічних особливостей. Зокрема, для створення тільки каркаса програми таким методом знадобиться близько 75 рядків коду. У міру ж збільшення складності програми її код може досягати воістину неймовірних розмірів. Однак та ж сама програма, написана з використанням

MFC, буде приблизно в три рази менше, оскільки більшість приватних деталей приховано від програміста.

Одною з основних переваг роботи з MFC є можливість багаторазового використання того самого коду. В зв'язку з тим, що бібліотека містить багато елементів, загальних для всіх Windows-додатків, немає необхідності щораз писати їх заново. Замість цього їх можна просто успадковувати (говорячи мовою об'єктно-орієнтованного програмування). Крім того, інтерфейс, забезпечуваний бібліотекою, практично незалежний від конкретних деталей, його що реалізують. Тому програми, написані на основі MFC, можуть бути легко адаптовані до нових версій Windows (на відміну від більшості програм, написаних звичайними методами). Ще однією істотною перевагою MFC є спрощення взаємодії із прикладним програмним інтерфейсом (API) Windows. Будь-який додаток взаємодіє з Windows через API, що містить кілька сотень функцій. Значний розмір API утрудняє спроби зрозуміти й вивчити його цілком. Найчастіше навіть складно простежити, як окремі частини API зв'язані один з одним! Але оскільки бібліотека MFC поєднує (шляхом інкапсуляції) функції API у логічно організовану безліч класів, інтерфейсом стає значно легше управляти.

Оскільки MFC являє собою набір класів, написаних мовою C++, тому програми, написані з використанням MFC, повинна бути в той же час програмами на C++. Для цього необхідно володіти відповідними знаннями. Для початку необхідно вміти створювати власні класи, розуміти принципи спадкування й уміти перевизначати віртуальні функції. Хоча програми, що використовують бібліотеку MFC, звичайно не містять занадто специфічних елементів з арсеналу C++, для їхнього написання проте потрібні солідні знання в даній області.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи мережевої SIEM для аналізу загроз

безпеці корпоративної IT-інфраструктури.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформувати висновки про виконаний обсяг робіт та одержані результати.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Рішення SIEM наступного покоління включають такі технології для виявлення складних загроз та горизонтального переміщення, а також для автоматизації реагування на інциденти:

Аналіз поведінки користувачів та об'єктів

Аналітика поведінки користувачів та об'єктів (UEBA) використовує алгоритми та машинне навчання для виявлення аномальних моделей поведінки користувачів та пристроїв (наприклад, маршрутизаторів, серверів та кінцевих точок) у мережі. UEBA дозволяє вашій організації виявляти ширший спектр кіберзагроз, таких як атаки методом перебору, розподілені атаки відмови в обслуговуванні (DDoS).) та внутрішні загрози.

Оркестрація безпеки та автоматизоване реагування

Оркестрація безпеки та автоматизація реагування (SOAR) допомагають координувати та автоматизувати реагування на виявлені загрози за допомогою автоматизованих сценаріїв або робочих процесів. Вони також використовують штучний інтелект.(Штучний інтелект) вивчає моделі поведінки, щоб передбачати подібні загрози до їх виникнення.

Переваги SIEM-рішень

Рішення SIEM може допомогти керувати ризиками кібербезпеки вашої організації, підтримуючи виявлення загроз, дотримання вимог та управління інцидентами безпеки. Рішення SIEM дозволяють вашій команді безпеки:

– керувати безперервним надходженням даних журналів з багатьох різних джерел:

○ допомагає зменшити вартість окремих інструментів, що використовуються різними групами у вашій організації;

o централізує дані журналів в одному сховищі;

– співвідносити та аналізувати великі обсяги даних, щоб мати змогу проактивно виявляти потенційні загрози, оскільки вони залишають сліди в різних джерелах журналів;

– автоматизувати завдання безпеки, щоб зменшити навантаження аналітиків безпеки шляхом автоматизації повторюваних завдань;

– отримувати автоматичні сповіщення та відповідні дії за допомогою автоматичного тригера на основі конкретних випадків використання для забезпечення швидкого реагування на інциденти;

– отримуйте дані в режимі реального часу по всій організації, щоб допомогти вашій організації швидко виявляти та усувати сліпі зони вразливостей у вашій мережі;

– пошук історичних даних журналів для різних мережевих вузлів та періодів часу для підтримки аналізу першопричин та виявлення інцидентів після того, як стався витік;

– генерувати звіти для аудиторів, щоб продемонструвати дотримання нормативних вимог та виявляти потенційні порушення на ранній стадії, щоб їх можна було усунути;

– переглядати інформаційні панелі управління, які відображають дані про події в інформаційних діаграмах, щоб побачити закономірності незвичайної діяльності:

o допомагає вашій організації визначити пріоритети ресурсів для першочергового вирішення найважливіших загроз.

Рішення SIEM дозволяють вашій організації автоматизувати впровадження, оцінку та постійний моніторинг засобів контролю безпеки. Згідно зі спеціальною публікацією (SP) 800-137 Національного інституту технологічних стандартів (NIST), технології SIEM можуть допомогти організаціям автоматизувати багато специфічних засобів контролю безпеки. Ці технічні, операційні та управлінські засоби контролю безпеки описані в документі

Кіберцентру « Управління ризиками безпеки IT : підхід життєвого циклу» (ITSG-33).

– Технічні засоби контролю безпеки:
  - AC-5 Розподіл обов'язків;
  - AU-2 Події, що підлягають аудиту;
  - AU-6 Аудиторський огляд, аналіз та звітність;
  - Скорочення аудиту AU-7 та створення звітів.
– Контроль операційної безпеки:
  - Моніторинг інцидентів IR-5;
  - PE-6 Моніторинг фізичного доступу;
  - Моніторинг інформаційної системи SI-4.
– Контроль безпеки управління:
  - Оцінювання безпеки CA-2;
  - Безперервний моніторинг CA-7;
  - Оцінка ризиків RA-3;
  - Сканування вразливостей RA-5.

Хмарні SIEM-рішення

У сфері кібербезпеки перехід до хмарних SIEM-рішень змінює те, як організації керують своїми даними та взаємодіють з ними. У звіті Gartner за 2023 рік було оцінено, що до кінця року 90% SIEM-рішень пропонуватимуть можливості виключно в хмарі. На відміну від традиційних локальних SIEM-рішень, які вимагають спеціалізованого апаратного та програмного забезпечення у власній інфраструктурі організації, хмарне SIEM-рішення розміщується на серверах, що обслуговуються стороннім постачальником хмарних послуг (CSP).

Хмарні SIEM-рішення дозволяють вашій організації перекласти більшу частину управління інфраструктурою на постачальника послуг зв'язку (CSP) та зосередитися на використанні вашої системи для досягнення ваших цілей безпеки. На практиці це означає, що журнали даних з мережевих пристроїв та систем вашої організації збираються, передаються в хмару та безпечно

зберігаються на серверах CSP.

Після цього ваша організація може взаємодіяти з вашими даними через веб-інтерфейсабо інтерфейс прикладного програмування (API), що надається постачальником послуг криптографії (CSP). Цей API зазвичай містить набір інструментів для аналізу даних, візуалізації та звітності. Це дозволяє вашій організації виконувати складну аналітику для виявлення, розслідування та реагування на інциденти безпеки.

Хмарні SIEM-рішення часто оснащені можливостями машинного навчання та штучного інтелекту для кращого виявлення аномалій та потенційних загроз. Це відбувається в режимі реального часу та у великих масштабах, надаючи організаціям потужний, гнучкий та ефективний інструмент для управління своєю кібербезпекою.

Типи хмарних пропозицій

Існує два типи пропозицій для хмарних SIEM-рішень: керовані та некеровані.

Керований

Це ближче до моделі « SIEM як послуга», де постачальник SIEM-рішення відповідає за хмарну інфраструктуру та її обслуговування. Постачальник SIEM-рішення також надає клієнту послуги моніторингу інцидентів у режимі реального часу та виявлення загроз. Клієнт зазвичай має менше контролю над управлінням життєвим циклом SIEM-рішення, оскільки це є відповідальністю постачальника. Хоча керовані рішення можуть бути дорожчими, вони знімають з клієнта тягар впровадження та обслуговування SIEM-рішення.

Некерований

Клієнт відповідає за створення, підтримку, усунення несправностей та управління життєвим циклом усіх компонентів SIEM-рішення. Третя сторона може надавати додаткову допомогу, але загалом клієнт несе відповідальність за доступність SIEM-рішення.та стабільність. Некеровані рішення можуть бути підходящим варіантом для організацій з високочутливими активами, яким

потрібен повний контроль над своїм SIEM-рішенням.

Переваги хмарних SIEM-рішень

Хмарні SIEM-рішення можуть забезпечити вашій організації кілька переваг.

Масштабованість та гнучкість

У міру зростання вашої організації або коливань попиту, хмарні рішення можуть адаптуватися до ваших потреб. Така масштабованість також означає, що ви платите лише за те, що використовуєте, що може бути економічно вигідним вибором для багатьох компаній.

Зменшення операційних накладних витрат

Завдяки локальному рішенню SIEM ваша організація відповідає за обслуговування апаратного та програмного забезпечення, що може бути ресурсомістким. Хмарні рішення SIEM перекладають значну частину цієї відповідальності на постачальника послуг зв'язку (CSP). Це дозволяє вашій команді безпеки зосередитися на стратегічних завданнях, а не на обслуговуванні.

Аналітика

Хмарні SIEM-рішення часто включають комерційну готову аналітику (COTS), специфічну для CSP. Ця аналітика розроблена для оптимальної роботи в інфраструктурі постачальника послуг, потенційно пропонуючи покращені можливості виявлення загроз та аналізу даних. Наявність цієї аналітики може покращити можливості вашої організації щодо кіберзахисту, використовуючи спеціалізовані знання та ресурси постачальника.

Недоліки хмарних SIEM-рішень

Хоча хмарні SIEM-рішення можуть запропонувати багато переваг вашій організації, вам слід знати про потенційні недоліки.

Проблеми конфіденційності даних

Коли ви використовуєте хмарне SIEM-рішення, ваші дані зберігаються на серверах постачальника послуг зв'язку (CSP). Перш ніж переходити на хмарне рішення, переконайтеся, що ви повністю розумієте та знайомі з методами

обробки та зберігання даних вашого постачальника.

Зафіксованість постачальника

Перехід на хмарне SIEM-рішення може призвести до прив’язки до постачальника, тобто до ситуації, коли важко або дорого перейти до іншого постачальника або повернутися до локального рішення. Багато хмарних сервісів є власністю CSP, що може ускладнити перенесення даних. Перш ніж вибрати хмарне SIEM-рішення, переконайтеся, що ви розумієте умови надання послуг, зокрема, що передбачає зміна постачальників.

Вартість

Хмарні SIEM-рішення можуть забезпечити економію коштів, особливо з точки зору обслуговування та інфраструктури, але вони також можуть збільшити витрати. Це особливо актуально, якщо ваша організація використовує багато даних, оскільки багато CSP стягують плату залежно від обсягу оброблених даних.

3.2 Розробка структурної схеми

Безпечне розгортання та експлуатація SIEM-рішень є життєво важливими. SIEM-рішення слід розглядати як систему вищої цінності, таку як адміністративний контроль або контроль доступу системи. Через свою роль у моніторингу та виявленні інцидентів безпеки, слід приділяти особливу увагу забезпеченню безпеки як продукту, так і постачальника. У разі виникнення вразливості нульового дня, а також через чутливість даних та рівень доступу до рішення SIEM, Кіберцентр вважає за потрібне розробляти архітектуру SIEM на основі рішень кількох постачальників, а не бути прив’язаним до одного. Такий підхід покращує загальний рівень безпеки, зменшуючи ризики, пов’язані з вразливостями, характерними для певних постачальників.

Неправильно впроваджене SIEM-рішення може призвести до більшої кількості хибнопозитивних результатів, виявлення більшої кількості «аномальних» подій та генерування додаткових, некорисних сповіщень. Це може

створювати навантаження на ресурси вашої команди кібербезпеки. Впроваджуючи наведені нижче найкращі практики, ваша організація може отримати максимальну користь від вашого SIEM-рішення.

Загальні рекомендації

– Визначення варіантів використання для моніторингу, оповіщення та аудиту:

o З цих випадків використання визначте джерела журналів, які потрібно отримати та проаналізувати.

– Розгляньте можливість проведення перевірки концепції (POC), щоб оцінити, чи підходить SIEM-рішення для вашого середовища:

o Налаштуйте POC у тестовому середовищі, яке базується на чітко визначених сценаріях користувачів і є репрезентативною підмножиною вашої інфраструктури та даних.

– Визначте свої найважливіші ресурси, такі як дані та пристрої, і налаштуйте SIEM-рішення для їх моніторингу.

– Налаштуйте відповідний моніторинг джерел журналів та сповіщення, щоб отримувати сповіщення про проблеми зі збиранням журналів.

– Оцініть, скільки даних ви хочете зібрати, щоб отримати повне уявлення про вашу мережу.

– Як мінімум, вам слід збирати дані журналу про:

o транзакції авторизації (успішні та невдалі спроби);

o зміни привілеїв користувачів, включаючи зміни облікових записів користувачів (зокрема створення та видалення), зміни членства в групах та механізмів автентифікації (паролі та багатофакторна конфігурація), а також додавання або видалення привілейованого доступу;

o помилки програми;

o процеси згоди, такі як умови та положення;

o дії, що виконуються всіма користувачами з правами адміністратора;

o реєстрація нових пристроїв в інфраструктурі, включаючи будь-які

мобільні телефони та персональні пристрої, що дозволяють використовувати власні пристрої.

– Запобігайте збору конфіденційних даних вашим SIEM-рішенням, таких як:

o фінансова інформація (наприклад, банківські записи або дані кредитної картки);

o персональну інформацію (наприклад, ідентифікаційний номер, виданий урядом);

o паролі та ключі шифрування;

– Зрозумійте вимоги до дотримання вимог вашого бізнесу та налаштуйте SIEM-рішення відповідно до них.

– Регулярно перевіряйте та тестуйте своє SIEM-рішення, щоб переконатися, що воно правильно налаштоване на основі впроваджених заходів безпеки та політик.

– Розробіть план реагування на інциденти, щоб ваша організація була готова належним чином впоратися з подією, коли трапляється інцидент безпеки:

o Для отримання додаткової інформації зверніться до нашої публікації «Розробка плану реагування на інциденти» (ITSAP.40.003).

– Синхронізуйте всі мережеві пристрої з центральним сервером часу, щоб гарантувати, що записані журнали аудиту використовують одне й те саме джерело часу;

o Налаштуйте щонайменше 3 сервери часу для полегшення обслуговування та усунення несправностей.

– Підпишіться на зовнішні канали загроз і створюйте сповіщення на основі даних, що надсилаються, регулярно оновлюйте логіку виявлення для виявлення нових загроз:

o Кіберцентр надає IoC організаціям, включаючи партнерів у канадській критичній інфраструктурі, через автоматизовану систему під назвою AVENTAIL, яку можна інтегрувати безпосередньо у вашу SIEM.

Дані журналу якості

Щоб ваша організація отримувала найкориснішу інформацію про діяльність у вашій мережі, переконайтеся, що високоякісні дані журналів надходять до вашого інструменту SIEM.

Виберіть відповідні методи збору журналів

Рішення SIEM можуть збирати та зберігати журнали безпеки з різних джерел. Визначте, який метод збору журналів підходить для потреб вашої організації.

– Потік журналів: Пристрої генерують журнали та надсилають їх безперервним потоком до колектора журналів рішення SIEM. Це забезпечує рішення SIEM інформацією в режимі реального часу.

– Надсилання журналів: Пристрій автоматично збирає журнали та надсилає (завантажує) їх безперервно або через регулярні проміжки часу до колектора журналів рішення SIEM. Колектор журналів налаштовано на прийом журналів у певному форматі та за певним протоколом (syslog, FTP тощо).

– Збір журналів: Як і при надсиланні журналів, цей метод використовує збирач журналів рішення SIEM для ініціювання підключення та запиту журналів. Цей метод часто використовується для збору журналів на рівні операційної системи за допомогою програмного агента.

Перегляд та оновлення аналізаторів журналів

Різні системи генерують журнали в різних форматах. Деякі формати журналів мають чітко визначену структуру та їх легко використовувати в SIEM-рішеннях, тоді як інші формати журналів менш узгоджені та складніші для аналізу та обробки в SIEM-рішеннях. Переконайтеся, що вибране вами SIEM-рішення може зрозуміти отримані журнали.

Формати журналів також можуть змінюватися з часом (наприклад, після оновлень програмного забезпечення), що може призвести до того, що SIEM не зможе аналізувати та індексувати журнали з певного джерела. Регулярно переглядайте аналізатори журналів та оновлюйте їх за потреби.

Правильне керування сховищем журналів

Дані журналів, отримані рішенням SIEM, зберігаються відповідно до налаштованих політик зберігання. Журнали можна надсилати до сховища для архівування або до механізму кореляції рішення SIEM, де вони будуть проаналізовані та зіставлені з іншими журналами. Така кореляція може надати важливу інформацію вашій IT- команді.

Залежно від обраного вами рішення SIEM, журнали можуть зберігатися або в отриманому вигляді, або у стиснутому форматі. Оскільки пошук стиснутих журналів займає більше часу, деякі рішення SIEM зберігають останні журнали в нестиснутому форматі. Після певного часу журнали стискаються, щоб зменшити використання пам'яті.

SIEM-рішення можуть отримувати тисячі журналів щосекунди, тому зберігання нестиснених журналів протягом тривалого періоду може призвести до високих витрат на зберігання. Якщо SIEM-рішення зберігає журнали в хмарі, витрати на зберігання також можуть значно зрости.

Видалення журналів після того, як вони більше не мають цінності, допоможе зменшити витрати на зберігання та продуктивність. Журнали, обсяг яких перевищує політику зберігання, можна відкинути або зберігати в дешевших рішеннях.

Зберігання даних журналу

Політики зберігання журналів можуть допомогти контролювати потреби в сховищі. Розробляючи політику зберігання журналів вашої організації, ретельно обміркуйте, як довго слід зберігати журнали безпеки. Як загальне правило, ми рекомендуємо зберігати важливі журнали вашої організації щонайменше 6 місяців. Для більш критичних журналів розгляньте період зберігання 13 місяців.

Термін зберігання залежатиме від вашого:

– галузеві стандарти організації;

– нормативні акти та закони;

– конкретні проблеми кібербезпеки, унікальні для вашого бізнес-

середовища;

– витрати на зберігання та доступність;

Багато компрометацій виявляються через довгий час після того, як стався витік. Згідно з публікацією IBM «Вартість звіту про витік даних за 2023 рік», середній час виявлення витоку становив 204 дні. Якщо у вашій організації стався витік, ваші журнали є важливим доказом, який допоможе вам виявити та розслідувати інцидент. Ретельно розробляйте політику зберігання журналів та періодично переглядайте її, щоб перевірити, чи потрібні коригування та чи зберігаються ваші журнали протягом належного часу.

Активація індексації найчастіше шуканих полів

Журнали з різних типів джерел містять різну інформацію та використовують різні формати. Рішення SIEM використовують аналізатори журналів для розуміння форматів журналів та інформації, яку вони містять. Це може включати сам журнал, інформацію про дату та час, а також розташування імені користувача або імені машини в потоці журналів. Ці поля можна індексувати, що призведе до швидшого пошуку.

Індексування журналів пришвидшує пошук, але вимагає додаткових ресурсів сховища та центрального процесора (CPU), що може вплинути на продуктивність рішення SIEM. Ми рекомендуємо індексувати лише ті поля, які часто шукаються.

Рішення SIEM повинно надавати інформацію про пошукові запити, зокрема, які поля шукаються та чи індексуються ці поля. Використовуючи цю інформацію, адміністратор SIEM може активувати або деактивувати індексацію залежно від того, як часто виконується пошук у полях.

Нормалізувати дані журналу

Нормалізація журналів важлива для кореляції подій та розслідування інцидентів. Рішення SIEM може отримувати журнали в різних форматах. Наприклад, ваша мережа може мати пристрої в різних часових поясах, деякі журнали можуть використовувати 12-годинний формат, а інші – 24-годинний, або

журнали Active Directory (AD) можуть містити імена користувачів, тоді як хмарні журнали відображають адресу електронної пошти користувача як його ім'я користувача.

Рішення SIEM повинно мати можливість нормалізувати якомога більше полів, щоб обмежити кількість рядків пошуку, що вказують на одного й того ж користувача або ресурс. Під час розслідування інцидентів пошук подій, що відбулися протягом певного періоду, повинен повертати журнали з усіх пристроїв, незалежно від часового поясу, на який налаштовано рішення SIEM.

Налаштування правил кореляції та порогових значень

Кореляція подій стосується аналізу подій у бізнес-контекстах та встановлення зв'язків між ними на основі набору попередньо визначених правил. Ці правила дозволяють вашому SIEM-рішенню визначати, які підозрілі дії слід розглядати як потенційні загрози безпеці. Для точного виявлення інцидентів механізм кореляції SIEM-рішення має бути налаштований належним чином. Налаштуйте правила кореляції та встановіть порогові значення на основі конкретних випадків використання або бізнес-потреб вашої організації. Ви можете почати зі стандартних правил конфігурації SIEM-рішення та деактивувати й активувати параметри відповідно до того, що ви хочете корелювати.

Архітектура нульової довіри

Термін «нульова довіра» (ZT) являє собою систему безпеки для захисту інфраструктури та даних. Центральний принцип ZT полягає в тому, що жоден суб'єкт (додаток, користувач чи пристрій) в інформаційній системі не є довіреним за замовчуванням. Довіра має оцінюватися та перевірятися щоразу, коли суб'єкт запитує доступ до нового ресурсу. Ступінь наданого доступу динамічно коригується залежно від рівня довіри, встановленого з суб'єктом. ZT передбачає прийняття нового підходу до безпеки, завжди припускаючи порушення та зосереджуючись на захисті ресурсів (наприклад, послуг та даних). Архітектура нульової довіри (ZTA) – це корпоративний підхід до проектування систем, у яких

безпека базується на принципах ZT. У стандарті NIST SP 1800-35B «Впровадження архітектури нульової довіри» описано приклади рішень для впровадження ZTA. Ці рішення припускають, що технологія SIEM є однією з базових функцій кібербезпеки організації, можливості якої поступово додаються в міру розвитку ZTA. Рішення SIEM підтримують впровадження ZTA, оскільки зібрані ними дані можуть бути використані в механізмі політик ZTA для прийняття рішень щодо динамічного доступу.

Великі організації та підприємства стикаються з постійно мінливим ландшафтом кіберзагроз. Щоб пом'якшити атаки з боку передових зловмисників, ваша організація повинна інвестувати в інструменти безпеки, які надають аналітику активності у вашій мережі в режимі реального часу. Інструменти кібербезпеки, такі як рішення SIEM, можуть забезпечити вам єдиний інтерфейс для отримання цієї аналітики. Рішення SIEM може допомогти вашій організації виявляти, аналізувати та реагувати на кіберзагрози, перш ніж вони порушать вашу бізнес-операцію. Як і у випадку з будь-яким важливим IT- рішенням, вам слід зважити всю інформацію, представлену в цій публікації, з урахуванням конкретних потреб та обставин вашої організації, щоб визначити, чи є рішення SIEM найкращим для вас.

Інтеграція сканера уразливостей з SIEM дозволяє сполучити кілька методів виявлення погроз і значно підвищити ймовірність своєчасного виявлення. Приміром, SIEM може виявити аномалію через baseline, але без інформації про те, що на активі є уразливість, SIEM не зможе сказати, із чим саме ця аномалія зв'язана. При наявності відомостей від сканера уразливості, SIEM зможе зробити вивід про те, що виробляється експлуатація уразливості. Маючи інформацію про уразливість, а також про критичність активів від сканера уразливостей, система SIEM здатна пріоритизувати інциденти по їхній критичності. Це дозволить у першу чергу реагувати на значимі інциденти, важливі для бізнесу.

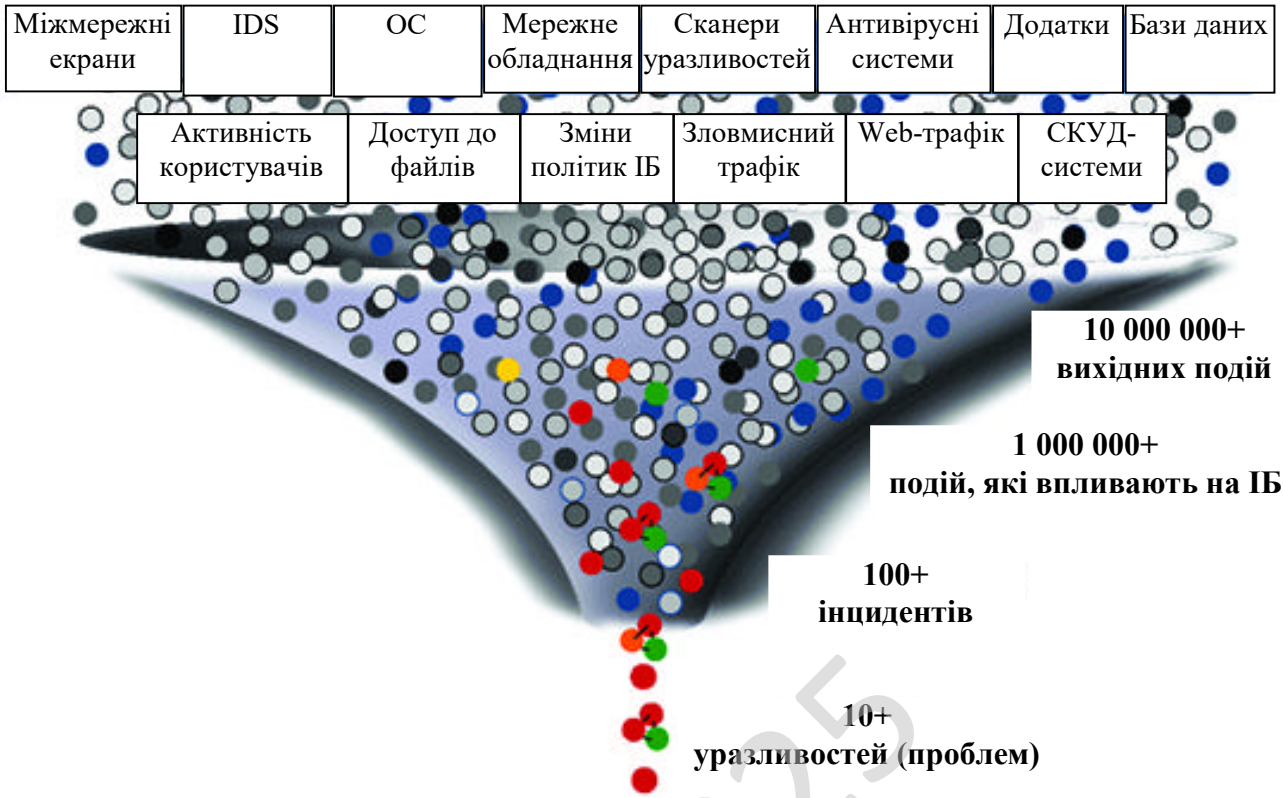

Рисунок 3.1 – Структурна схема системи

Сканер уразливостей є відмінним постачальником інвентаризаційної інформації для SIEM, наприклад, про версії програмного забезпечення і його конфігурацій. Ця інформація може використовуватися при виявленні інциденту, при з'ясуванні причин його виникнення. Використання убудованого в SIEM механізму перевірки відповідності внутрішнім політикам і високорівневим стандартам без інтеграції зі сканером уразливостей не дає повноцінної картини, тому що використовується дуже мала частка технічних вимог.

Жоден джерело не надасть більше детальної й повної інформації про наявність уразливості й про можливість її експлуатації (з урахуванням топологічної структури мережі й конфігурацій) краще, ніж сканер уразливостей. Уразливість може бути присутнім, але бути при цьому неексплуатований (закритий мережний порт, зупинена служба, на активному мережному встаткуванні організована VLAN або правилами міжмережного екрана

заблокований трафік на даний порт). Інформація про це може істотно знизити залишкові ризики, допоможе витрачати кошти на дійсно необхідні засоби захисту, а також виключити помилкові інциденти.

Процес керування конфігураціями, що так складно реалізувати, стає простим при використанні зв'язування SIEM і сканера уразливостей. Ви можете аналізувати, що змінилося, ким і коли були зроблені зміни, а також можете автоматично оцінити, на що вони вплинули. Для цього необхідно лише скласти найпростіші правила кореляції в SIEM і налаштувати параметри переданої інформації від сканера; всю іншу логіку здійснить сама система SIEM. Природно, що чим більше ефективних джерел інформації в SIEM, тим більше ймовірність виявлення погрози на ранній стадії її виникнення. Ви можете використовувати SIEM або сканер уразливостей окремо. Але описане зв'язування значно мінімізує ризики.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. У багатьох пристроях для корпоративних мереж, наприклад інтегрованих маршрутизаторах, часто є багатофункціональні програмні файрволи. Такі файрволи звичайно реалізують трансляцію мережних адрес (NAT), динамічний аналіз пакетів (SPI), а також фільтрацію по IP-адресах, додатках і веб-сайтах. Додатково вони підтримують функції DMZ. Інтегрований маршрутизатор дозволяє налаштувати примітивну DMZ для доступу до внутрішнього сервера з вузлів за межами мережі. Для цього сервер повинен мати статичну IP-адресу, що вказується в конфігурації DMZ. Інтегрований маршрутизатор ізолює трафік, що пересилається на зазначений IP-адрес. Цей трафік пропускається тільки на той порт комутатора, до якого підключений сервер. На всі інші вузли як і раніше поширюється захист файрволу.

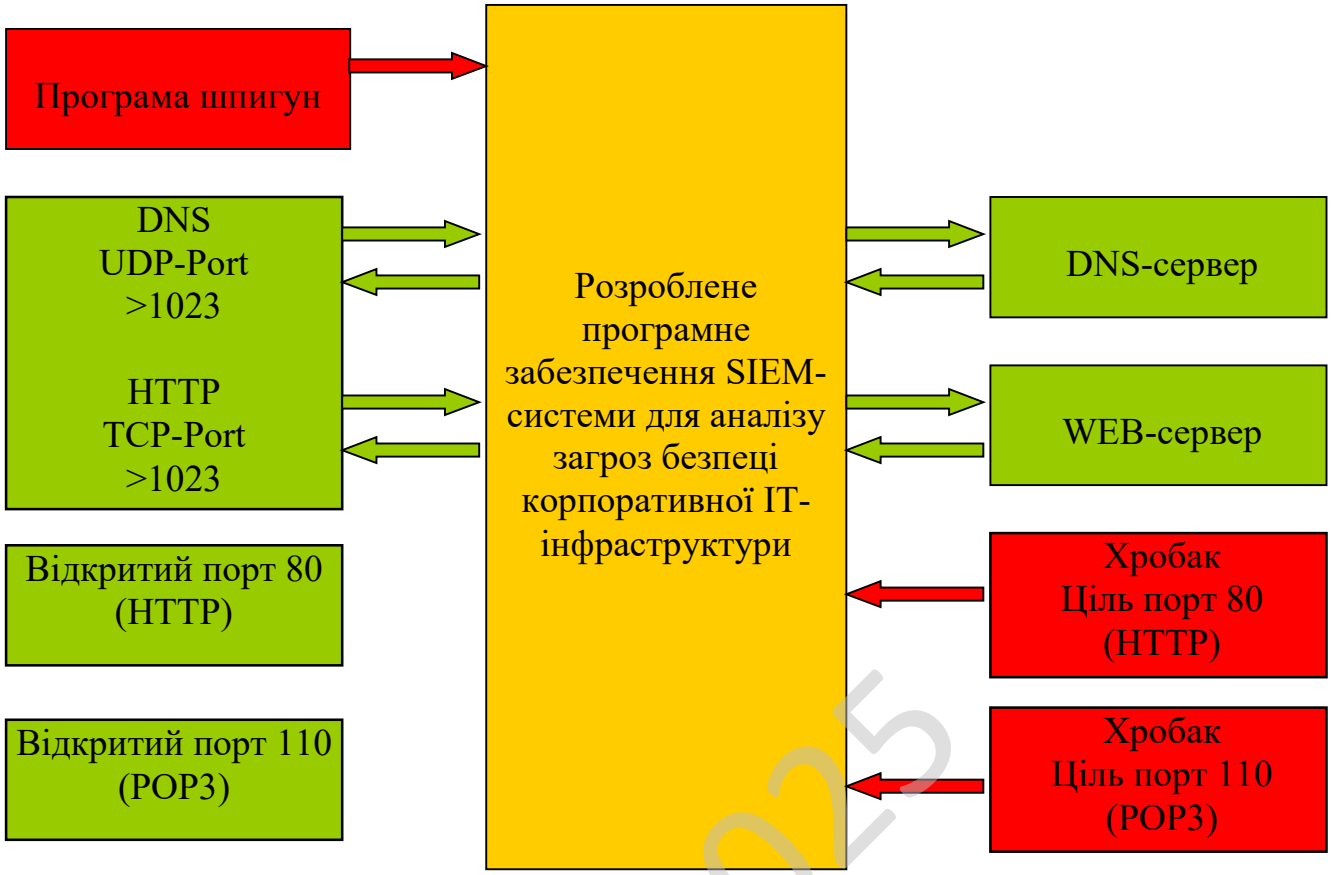

Рисунок 3.2 – Функціональна схема системи

При активації DMZ у найпростішому виді зовнішні вузли одержують доступ до всіх портів сервера, наприклад 80 (HTTP), 21 (FTP) і 110 (POP3 для електронної пошти).

Функція переадресації портів дозволяє налаштувати більш строгу конфігурацію DMZ. У цьому випадку вказуються порти, які повинні бути доступні на сервері. Пропускається тільки трафік, спрямований на ці порти. Весь інший трафік виключається.

Бездротова точка доступу в складі інтегрованого маршрутизатора часто вважається частиною внутрішньої мережі. Необхідно усвідомлювати, що при роботі точки доступу в незахищеному режимі всі користувачі, що підключилися до неї, одержують доступ до внутрішньої захищеної мережі без проходження файрволу. Зловмисники можуть у такий спосіб одержати доступ до внутрішньої мережі, минаючи всі засоби захисту.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється. Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходе взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Рисунок 3.3 – Діаграма взаємодії процесів

Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схеми є основою ПЗ. Тому від точності і детальності проробки блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об’єктно-орієнтована що вимагає оптимізації, також те, що при розробці програми слід надати особливу увагу модулю SIEM-системи для аналізу загроз безпеці корпоративної IT-інфраструктури.

Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні блоки можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірки поточного стану та поверненням на початок схеми чи з завершенням роботи розробленого ПЗ.

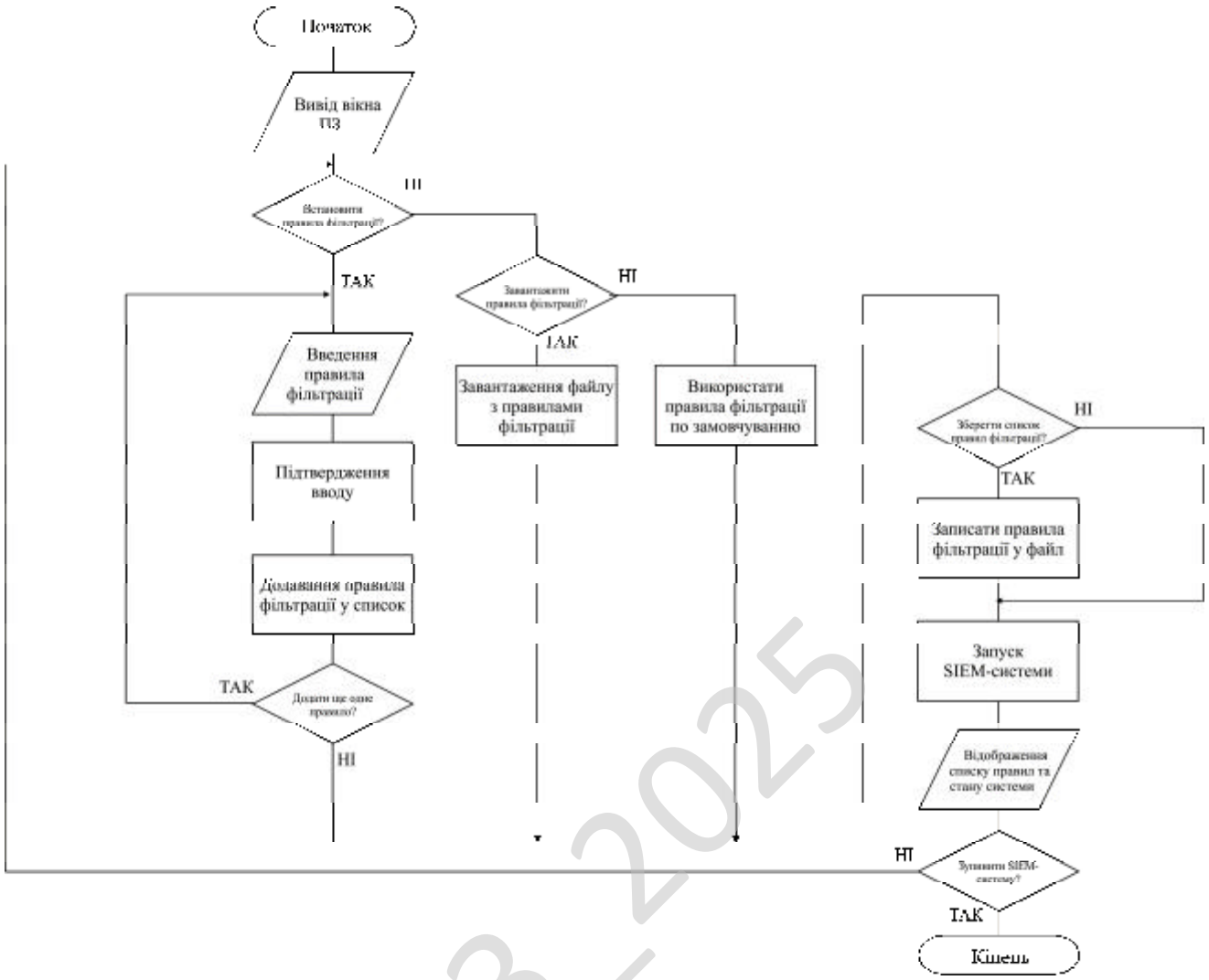

Рисунок 4.1 – Блок-схема основної програми

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем.

UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

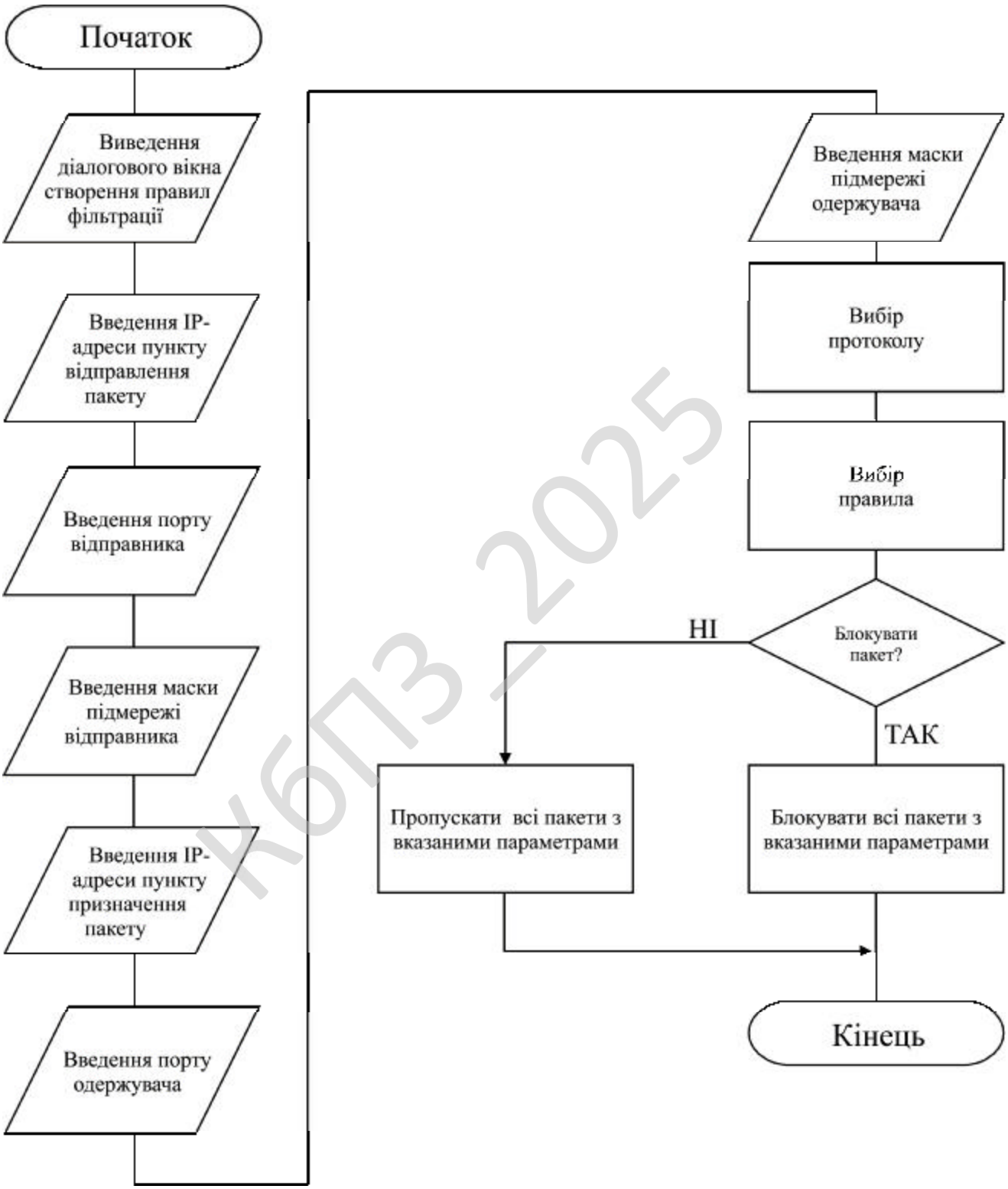

Рисунок 4.2 – Блок-схема роботи підпрограми

Розглянемо використані технології та їх основні компоненти що підтверджують правильність використаних проектних рішень.

Управління вимогами це процес запису, аналізу, трасування, пріоритезації і узгодження вимог та контролю змін і доведення до їх зацікавлених сторін. Це безперервний процес протягом всього життя проекту. Вимога – якість, якій мають відповідати результати проекту (продукту або послуги).

Мета управління вимогами полягає в тому, щоб переконатися, що організація відповідає потребам і очікуванням своїх клієнтів, внутрішніх або зовнішніх зацікавлених сторін. Управління вимогами починається з аналізу і виявлення цілей і обмежень організації. Управління вимогами додатково включає в себе підтримку планування вимог, інтеграції вимог і організації роботи з ними (атрибути для вимог).

Управління вимогами передбачає спілкування між членами проектної групи і зацікавленими сторонами, і адаптацію до змін у вимогах протягом всього проекту. Щоб запобігти перетину поля одного класу вимог з іншим, постійні зв'язки між членами команди розробників є критичними. Наприклад, при розробці програмного забезпечення для внутрішнього використання у бізнесу можуть бути настільки сильні потреби, що він може проігнорувати вимоги користувачів, або вважати, що створені сценарії використання покриють також і користувальницькі вимоги.

Відслідковування вимоги фактично означає документування всього життєвого циклу вимоги. Часто необхідно дізнатися першоджерело кожної вимоги. Для цього всі зміни вимог повинні бути задокументовані, щоб досягти стану повного відстеження. Відстежувати треба бути навіть використання реалізованих вимог.

Вимоги мають різні джерела, такі як ділова людина, що замовляє продукт, менеджер зі збуту і фактичний користувач. У всіх цих людей є різні вимоги до продукту. Використовуючи відслідковування вимог, реалізована в системі функція може бути простежена назад до людини або групі, яка замовляла її під

час збору вимог. Ця особливість може, наприклад, використовуватися в процесі розробки для пріоритезації вимог, визначаючи, наскільки цінною є дана вимога для певного користувача.

Відслідковування може також використовуватися після розгортання продукту. Наприклад, коли вивчення використання системи показує, що якась функція не використовується, можна визначити навіщо вона була потрібна спочатку.

Завдання управління вимогами

На кожному етапі процесу розробки існують ключові методи і задачі пов'язані з управлінням вимогами. Для ілюстрації, розглянемо наприклад стандартний процес розробки з п'ятьма фазами: дослідженням, аналізом здійсненності, дизайном, розробкою та тестуванням і випуском.

Дослідження. Під час фази дослідження збираються перші три класи вимог від користувачів, бізнесу і команди розробників. У кожній області задають однакові питання: які цілі, які обмеження, які використовуються процеси та інструменти і так далі. Тільки коли ці вимоги добре зрозумілі, можна приступати до розробки функціональних вимог.

Тут необхідне застереження: незалежно від того, як сильно група намагається це зробити, вимоги не можуть бути повністю визначені на початку проекту. Деякі вимоги змінюються, або тому що вони просто не були знайдені спочатку, або тому що внутрішні чи зовнішні сили торкаються проекту в середині циклу. Таким чином, учасники групи повинні спочатку погодитися, що головна умова успіху – гнучкість у мисленні та діях.

Результатом стадії дослідження є документ – специфікація вимог, схвалений усіма членами проекту. Пізніше, в процесі розробки, цей документ буде важливий для запобігання розповзання меж проекту або непотрібних змін. Оскільки система розвивається, кожна нова функція відкриває світ нових можливостей, таким чином специфікація вимог прив'язує команду до оригінального бачення системи і дозволяє контрольоване обговорення змін.

У той час як багато організацій все ще використовують звичайні документи для керування вимогами, інші управляють своїми базовими вимогами, використовуючи програмні інструменти.

Ці інструменти керують вимогами використовуючи базу даних, і зазвичай мають функції автоматизації відстеження (наприклад, дозволяючи створювати зв'язки між батьківськими і дочірніми вимогами, або між тестами і вимогами), управління версіями, і управління змінами. Зазвичай такі інструментальні засоби містять функцію експорту, яка дозволяє створювати звичайний документ, експортуючи дані вимог.

Аналіз здійсненності

На стадії аналізу здійсненності визначається вартість вимог. Для користувальницьких вимог поточна вартість роботи порівнюється з майбутньою вартістю встановленої системи. Задаються питання такі як: «Скільки нам зараз варті помилки введення даних?» Або, «Яка вартість втрати даних через помилки оператора пов'язаної з використовуваним інтерфейсом?». Фактично, потреба в новому інструменті часто розпізнається, коли подібні питання потрапляють до уваги людей, що займаються в організації фінансами.

Ділова вартість включає відповіді на такі питання як: «У якого відділу є бюджет на це?» «Який рівень повернення коштів від нового продукту на ринку?» «Який рівень скорочення внутрішніх витрат на навчання і підтримку, якщо ми зробимо нову, більш просту в використанні систему?»

Технічна вартість пов'язана з вартістю розробки програмного забезпечення та апаратною вартістю. «Чи є у нас потрібні люди, щоб створити інструмент?» «Чи потребуємо ми нове устаткування для підтримки нової системи?»

Подібні питання дуже важливі. Група повинна з'ясувати, чи буде новий автоматизований інструмент мати достатню ефективність аби перенести частину тягаря користувачів на систему і зекономити час людей.

Ці питання також вказують на основну суть управління вимогами. Людина і інструмент формують систему, і це розуміння особливо важливе, якщо

інструмент – комп'ютер або новий додаток на комп'ютері. Людський розум вкрай ефективний у паралельній обробці та інтерпретації тенденцій з недостатніми даними. Комп'ютерний процесор ефективний у послідовній обробці і точному математичному обчисленні. Основна мета управління вимогами для програмного проекту полягала б у тому, щоб гарантувати, що автоматизована робота призначена «правильному» процесору.

Наприклад, «не змушуйте людину пам'ятати, де вона знаходиться в системі. Примусьте інтерфейс завжди повідомляти про місцезнаходження людини в системі ». Або «не змушуйте людини вводити ті ж самі дані в два екрани. Примусьте систему зберігати дані і заповнювати їх де необхідно автоматично». Результатом стадії аналізу здійсненності є бюджет і графік проекту.

Дизайн. Припускаючи, що вартість точно визначена і переваги, які будуть отримані, є досить великими, проект може перейти до стадії проектування.

На стадії дизайну основна діяльність управління вимогами полягає в тому, щоб перевіряти чи відповідають результати дизайну документу вимог, щоб упевнитися, що робота залишається в межах проекту.

І знову, гнучкість є ключем до успіху. Ось класичний приклад змін проекту, які відмінно працювали. Проектувальники Форда на початку 1980-х очікували, що ціни на бензин піднімуться до 3,18 дол за галон до кінця десятиліття. На середині процесу дизайну автомобіля Ford Taurus, ціни встановилися приблизно на рівні 1,50 дол за галон. Колектив дизайнерів вирішив, що вони могли б створити більший, більш зручний, і більш потужний автомобіль, якщо б ціни на бензин залишилися низькими. Таким чином, вони перепроектувати автомобіль. Коли новий автомобіль вийшов, він встановив загальнонаціональні рекорди продажів.

У більшості випадків, однак, відступ від оригінальних вимог до такої міри не працює. Таким чином документ вимог стає ключовим інструментом, який допомагає команді приймати рішення про зміни дизайну.

Розробка та тестування. На стадії розробки і тестування, основна

діяльність управління вимогами – це гарантувати, що робота і ціна залишаються в межах графіка і бюджету, і що створюваний інструмент дійсно відповідає вимогам. Основним інструментом, використовуваним на цій стадії, є створення прототипу і ітераційне тестування. Для програмного додатка користувацький інтерфейс може бути створений на папері і перевірений з потенційними користувачами, в той час як створюється основа програми. Результати цих тестів записуються в керівництві по дизайну користувацького інтерфейсу і передаються колективу дизайнерів. Це економить їх час і робить їх завдання набагато простіше.

При розробці ПЗ було використано підходи ри́зик-ме́неджменту – це система управління ризиками, яка включає в себе стратегію та тактику управління, направлені на досягнення основних цілей. Ефективний ризик-менеджмент включає:

– систему управління;

– систему ідентифікації і вимірювання;

– систему супроводження (моніторингу та контролю).

Сучасна наука представляє ризик як вірогідну подію, в результаті настання якої можуть відбутися позитивні, нейтральні або негативні наслідки. Якщо ризик припускає наявність як позитивних, так і негативних результатів, він відноситься до спекулятивних ризиків. Якщо ж наслідки негативні, або відсутні взагалі, такий ризик іменується чистим.

Мета ризик-менеджменту – підвищення конкурентоспроможності господарюючих суб'єктів за допомогою захисту від реалізації чистих ризиків.

Теорія ризик-менеджменту ґрунтується на трьох базових поняттях: корисності, регресії і диверсифікації.

У 1738 швейцарський математик Даніель Бернуллі доповнив теорію вірогідності методом корисності або привабливості того або іншого результату подій. Ідея Бернуллі полягала в тому, що в процесі ухвалення рішення люди приділяють більше уваги розміру наслідків різних результатів, ніж їх вірогідність.

В кінці XIX століття англійський дослідник Ф. Гальтон запропонував вважати регресію або повернення до середнього значення універсальною статистичною закономірністю. Суть регресії трактувалася ним як повернення явищ до норми з часом. Згодом було доведено, що правило регресії діє в найрізноманітніших ситуаціях, починаючи з азартних ігор та розрахунку вірогідності виникнення нещасних випадків, і закінчуючи прогнозуванням коливань економічних циклів.

У 1952 аспірант Університету Чикаго Гарі Марковіц в статті «Диверсифікація вкладень» («Portfolio Selection») математично обґрунтував стратегію диверсифікації інвестиційного портфеля, зокрема, він показав, як шляхом продуманого розподілу вкладень мінімізувати відхилення прибутковості від очікуваного показника. У 1990 Г. Марковіцу присуджена Нобелівська премія за розробку теорії і практики оптимізації портфеля фондових активів.

Етапи ризик-менеджменту

У ризик-менеджменті прийнято виділяти декілька ключових етапів:

– на першому етапі відбувається виявлення ризику з супутньою оцінкою вірогідності його реалізації і масштабу наслідків;

– на другому етапі здійснюється розробка ризик-стратегії з метою зниження вірогідності реалізації ризику і мінімізації можливих негативних наслідків;

– на третьому етапі вибираються методи і інструменти управління виявленим ризиком;

– на четвертому етапі проводиться безпосереднє управління ризиком;

– на завершальному етапі оцінюються досягнуті результати і коректується ризик-стратегія.

За ключовий етап ризик-менеджменту вважається етап вибору методів і інструментів управління ризиком.

Методи і інструментарій ризик-менеджменту

Базовими методами ризик-менеджменту є відмова від ризиків, зниження, передача і ухвалення.

Ризик-інструментарій значно ширший. Він включає політичні, організаційні, правові, економічні, соціальні інструменти, причому ризик-менеджмент як система допускає можливість одночасного застосування декількох методів і інструментів ризик-управління.

Найбільш часто вживаним інструментом ризик-менеджменту є страхування. Страхування припускає передачу відповідальності за відшкодування передбачуваного збитку сторонній організації (страхової компанії).

Прикладами інших інструментів можуть бути відмова від надмірно ризикової діяльності (метод відмови), профілактика або диверсифікація (метод зниження), аутсорсинг витратних ризикових функцій (метод передачі), формування резервів або запасів (метод ухвалення).

Підпрограма виводу головного вікна:

```
//ініціалізація й створення вікна і його компонентів
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
//створення вікна
  if (CFrameWnd::OnCreate(lpCreateStruct) == -1)
       return -1;
//створення ToolBar
if (!m_wndToolBar.CreateEx(this,TBSTYLE_FLAT,WS_CHILD|WS_VISIBLE|CBRS_TOP
  | CBRS_GRIPPER | CBRS_TOOLTIPS | CBRS_FLYBY | CBRS_SIZE_DYNAMIC) ||
  !m_wndToolBar.LoadToolBar(IDR_MAINFRAME))
  {
  TRACE0("Failed to create toolbar\n");
  return -1;      // fail to create
  }
//створення StatusBar
  if (!m_wndStatusBar.Create(this) ||
!m_wndStatusBar.SetIndicators(indicators, sizeof(indicators)/sizeof(UINT)))
  {
       TRACE0("Failed to create status bar\n");
       return -1;      // fail to create
  }
```

```
    m_wndToolBar.EnableDocking(CBRS_ALIGN_RIGHT);
    EnableDocking(CBRS_ALIGN_ANY);
    DockControlBar(&m_wndToolBar);
//Установка заголовка
    this->SetWindowText("Firewall");
    return 0;
}
```

Потім користувач або завантажує попередньо збережений файл з правилами фільтрації, або вводить правила вручну, чи використовує значення за замовчуванням.

Після створення списку правил, його можна зберегти у файлі, для подальшого використання. Підпрограма збереження правил у файлі:

```
void CMainFrame::OnSaveRules()
{
CFirewallAppDoc *doc = (CFirewallAppDoc *)GetActiveDocument();
if(doc->nRules == 0)
   {
        AfxMessageBox("There isnt Rules to Save.");
        return;
   }
CFileDialog dg(FALSE, NULL, NULL, OFN_HIDEREADONLY | OFN_CREATEPROMPT,"Rule
Files(*.rul)|*.rul|all(*.*)|*.*||", NULL);
if(dg.DoModal()==IDCANCEL) return;
   CString nf=dg.GetPathName();
   if(nf.GetLength() == 0)
   {
        AfxMessageBox("This file name isn't valid.");
        return;
   }
   CFile file;
   CFileException e;
if( !file.Open( nf, CFile::modeCreate | CFile::modeWrite, &e ) )
{
        AfxMessageBox("Error openning the file.");
        return;
}
PFFORWARD_ACTION action = pckFilter.GetDefaultAction();
file.Write(&action, sizeof(PFFORWARD_ACTION));
unsigned int i;
for(i=0;i<doc->nRules;i++)
```

```
        {
            file.Write(&doc->rules[i], sizeof(RuleInfo));
        }
    file.Close();
    }
```

Підпрограма завантаження списку правил з файлу:

```
    void CMainFrame::OnLoadRules()
    {
        CFile file;
        CFileException e;
        DWORD nRead;
        CFirewallAppDoc *doc = (CFirewallAppDoc *)GetActiveDocument();
        CFileDialog dg(TRUE,NULL, NULL, OFN_HIDEREADONLY | OFN_CREATEPROMPT,"Rule
Files(*.rul)|*.rul|all(*.*)|*.*||", NULL);
        if(dg.DoModal()==IDCANCEL)
            return;
        CString nf=dg.GetPathName();
        if(nf.GetLength() == 0)
        {
            AfxMessageBox("This file name isn't valid.");
            return;
        }
        if( !file.Open(nf, CFile::modeRead, &e ) )
        {
            AfxMessageBox("Error openning the file.");
            return;
        }
        doc->ResetRules();
        PFFORWARD_ACTION action;
        file.Read(&action, sizeof(PFFORWARD_ACTION));
        if(action != pckFilter.GetDefaultAction())
        {
            pckFilter.RemoveAll();
            pckFilter.SetDefaultAction(action);
            doc->defaultAction = action;
        }
            RuleInfo rule;
           do
          {
            nRead = file.Read(&rule, sizeof(RuleInfo));
            if(nRead == 0)
```

```
            break;
        if(doc->AddRule(rule.sourceIp,
                        rule.sourceMask,
                         rule.sourcePort,
                         rule.destinationIp,
                         rule.destinationMask,
                         rule.destinationPort,
                         rule.protocol,
                         1)  != 0)
        {
            AfxMessageBox("Error adding a rule.");
            break;
        }
      }while (1);
    CFirewallAppView *view = (CFirewallAppView *)GetActiveView();
    view->UpdateList();
}
```

Після того як створені чи відкриті правила фільтрації можна запустити файрвол. Після запуску файрволу, програма починає фільтрувати пакети по вказаним правилам. Підпрограма запуску файрволу:

```
void CMainFrame::OnButtonstart()
{
    CFirewallAppDoc *doc = (CFirewallAppDoc *)GetActiveDocument();
    unsigned int i;
    DWORD result;
    PIP_ADAPTER_INFO pAdapterInfo = NULL, aux;
    IP_ADDR_STRING *localIp;
    unsigned long len = 0;
      //Пошук адаптера мережної карти
    GetAdaptersInfo(pAdapterInfo, &len);
    pAdapterInfo = (PIP_ADAPTER_INFO) malloc (len);
    result = GetAdaptersInfo(pAdapterInfo, &len);
    if(result != ERROR_SUCCESS)
    {
        AfxMessageBox("Error getting adapters info.");
        return;
    }
    // Посилка правил на інтерфейс адаптера
    for(i=0;i<doc->nRules;i++)
    {
        // на всі знайдені адаптери
```

```
        for(aux=pAdapterInfo;aux != NULL;aux=aux->Next)
        {
             // на кожний IP адаптера
for(localIp=&aux->IpAddressList;localIp!=NULL;localIp=localIp->Next)
{
pckFilter.AddFilter(CharToIp(localIp->IpAddress.String),
   ANY_DIRECTION,
   doc->rules[i].sourceIp,
   doc->rules[i].sourceMask,
doc->rules[i].destinationIp,
   doc->rules[i].destinationMask,
   doc->rules[i].sourcePort,
   doc->rules[i].destinationPort,
   doc->rules[i].protocol);
   }
}
   }
   started = TRUE;
}
```

Для припинення фільтрації пакетів слід зупинити файрвол. Підпрограма зупинки файрволу виглядає наступним чином:

```
void CMainFrame::OnButtonstop()
{
   pckFilter.RemoveAll();
   started = FALSE;
}
```

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

– набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;

– набір клієнтів, які використовують сервіси, що надаються серверами;

– мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до

серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

– рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;

– прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;

– рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

– модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

– модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузерами.

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка ПЗ. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення –звичайний браузер. Це позбавляє його необхідності завантажувати та інсталювати спеціальні програми (хоча інколи така необхідність все-таки виникає).

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний

інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються. Слід згадати про технології EJB (Enterprise Java Beans), CORBA, а також про .NET – порівняно нову ініціативу компанії Microsoft. Для зберігання даних та їх передачі часто використовується так звана розширювана мова розмітки XML (Extensible Markup Language).

Jira – була використана комерційна система відслідковування помилок, призначена для організації взаємодії з користувачами, хоча в деяких випадках використовується і для управління проектами. Розроблено компанією Atlassian, є одним з двох її основних продуктів (поряд з вікі-системою Confluence). Має веб-інтерфейс.

Назва системи отримано шляхом усічення слова «Gojira» – Японського імені монстра Годзилла, що, в свою чергу, є відсиланням до назви конкуруючого продукту – Bugzilla; створювалася в якості заміни Bugzilla і багато в чому повторює її архітектуру. Система дозволяє працювати з декількома проектами. Для кожного з проектів створює і веде схеми безпеки і схеми оповіщення.

До версії 3.13.5 (включно) розрізнялися редакції Enterprise, Professional і Standard, після – Залишилася тільки редакція Enterprise (для великих організацій).

Система заснована на Java EE і працює на кількох популярних системах управління базами даних і операційних системах.

Основний елемент обліку в системі – завдання (ticket або issue). Завдання містить назву проекту, тему, тип, пріоритет, компоненти і зміст. Завдання може бути розширена додатковими полями (також і нові призначені для користувача поля можуть бути визначені), додатками (наприклад – Фотографіями, скриншотами) або коментарями. Завдання може редагуватися або просто

змінювати статус, наприклад, з «відкритий» в «закритий». Які переходи між станами можливі, визначається через настроюється потік операцій. Будь-які зміни в задачі записуються в журнал.

Jira має велику кількість можливостей конфігурації: для кожної програми може бути визначений окремий тип завдання з власним workflow, набором статусів, одним або декількома видами уявлення (screens). Крім того, за допомогою так званих «схем» можна визначити для кожного індивідуального Jira-проекту власні права доступу, поведінку і видимість полів і багато іншого.

Завдяки універсальному підходу можна пристосувати Jira для багатьох непрофільних завдань, наприклад, керування вимогами, керування ризиками, аж до реалізації невеликої системи бронювання, автоматизації процесу рекрутингу.

Для інтеграції з зовнішніми системами підтримує інтерфейси SOAP, XML-RPC і REST. Поставляється із засобами інтеграції з такими системами управління версіями, як Subversion, CVS, Git, Clearcase, Team Foundation Server, Mercurial і Perforce. Існують доповнення, що дозволяють вбудувати Jira в інтегровані середовища розробки, в тому числі Eclipse і IntelliJ IDEA. Перекладена багатьма мовами, включаючи російську, англійську, японську, німецьку, французьку, іспанську. Для сторонніх розробників надаються кошти розробки розширень системи – плагінів. Розробники розширень можуть викладати плагіни для продажу на спеціальний розділ сайту Atlassian. Є комерційним продуктом, який може бути ліцензований для роботи на локальному сервері або доступний в якості віддаленого додатки. Ціноутворення залежить від максимального числа користувачів, при цьому близько $50 за користувача для локального і $7 на місяць за користувача для віддаленого доступу є типовими цінами. Для академічних і комерційних клієнтів доступний повний вихідний код під ліцензією розробника. Для проектів з відкритим вихідним кодом Atlassian надає спеціальну безкоштовну ліцензію при дотриманні наступних правил:

– проект використовує ліцензії, схвалені Open Source Initiative;

– Вихідний код проекту доступний для скачування;

– у проекту є публічно доступна веб-сайт;

– програмне забезпечення від Atlassian є на веб-сайті проекту.

Була використана водоспадна (каскадна) модель життєвого циклу ПЗ (waterfall model) – послідовний метод розробки програмного забезпечення, названий так через діаграму схожу на водоспад. Ця модель розробки запозичена з системної інженерії у виробництві та будівництві – областях, в яких зміни на пізніх етапах дуже дорогі, або неможливі. Наприклад, для створення складних інженерних конструкцій (споруд, літаків, мостів і т.п.). Зміни в проекті фундаменту будинку після того, як покладений дах коштують дуже дорого, тому перфекціонізм на початкових етапах проектування просто необхідний. Інженери, які починали займатись розробкою програмного забезпечення перейшовши з інших галузей, просто адаптували звичну модель, тому що на ранніх етапах розвитку комп'ютерної техніки не було методологій створених саме для програмування. Проте, схожі методології застосовуються для програмного забезпечення й далі, у випадках коли вимоги фіксовані, і вимагається висока якість та надійність, наприклад в системах для військових чи медичних потреб. Перший формальний опис водоспадної моделі, після якої вона стала популярною був здійснений В.В. Ройсом у 1970. Попри те, що стаття містить переважно критику методу, на неї часто посилаються.

Переваги методу:

– Ніяких переробок.

– Гарна специфікація перетікає в гарну документацію.

– Зрозуміла модель.

– Розробники можуть мати низьку кваліфікацію.

Недоліки:

– Необхідний перфекціонізм на кожному етапі.

– Важко вносити зміни (якщо взагалі можливо).

– Надлишкове проектування.

– Поділ розробників на "perfect" та "code monkeys".

Модифікації. Через те що цей метод погано підходить для розробки саме ПЗ, частіше використовують його модифікації.

Найвідоміша модифікація – Sashimi. Названа так через японську страву сашимі (суші нарізане і сервіроване так, що складені рядочком шматочки накладаються один на одного). В моделі розробки Сашимі фази життєвого циклу йдуть одна за одною, але при цьому перекриваються одна з одною в часі.

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму FEAL – блоковий шифр, запропонований Акіхіро Симідзу і Седзі Міягуті.

У ньому використовуються 64-бітовий блок і 64-бітовий ключ. Його ідея полягає і в тому, щоб створити алгоритм, подібний DES, але з більш сильною функцією етапу. Використовуючи менше етапів, цей алгоритм міг би працювати швидше. На жаль, дійсність виявилася далекою від цілей проекту.

Як вхід процесу шифрування використовується 64-бітовий блок відкритого тексту. Спочатку блок даних підлягає операції XOR з 64 бітами ключа. Потім блок даних розщеплюється на ліву і праву половини. Об'єднання лівої і правої половин за допомогою XOR утворює нову праву половину. Ліва половина і нова права половина проходять через N етапів (спочатку 4). На кожному етапі половина об'єднується за допомогою функції F[1] з 16 бітами ключа і за допомогою XOR – з лівою половиною, створюючи нову праву половину. Вихідна права половина (на початок етапу) стає новою лівою половиною. Після N етапів (ліва і права половини не переставляти після N-го етапу) ліва половина знову об'єднується з допомогою XOR з правою половиною, утворюючи нову праву половину, потім ліва і права об'єднуються разом в 64-бітове ціле. Блок даних об'єднується за допомогою XOR з іншими 64 бітами ключа і алгоритм завершується.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ SIEM-системи для аналізу загроз безпеці корпоративної IT-інфраструктури яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

– Навігаційне меню: Файл; SIEM-дані; Формати; Фільтри; Налаштування; Довідка.

– Функції поточного налаштування SIEM.

– Розділу обрання режиму роботи.

– Розділу журналу роботи ПЗ.

– Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.

– Функціональних кнопок ПЗ.

SIEM-система не тільки автоматизує аналіз різних системних подій. Немаловажно, що з її допомогою можна виявити дії, які зовні виглядають цілком необразливими, але в сукупності являють загрозу. Наприклад, якщо довірений користувач відправляє конфіденційні дані на email-адресу, що лежить поза звичайним колом адресатів, те DLP-система не завжди виявляє такі дії, однак SIEM згенерує інцидент на базі накопиченої статистики.

Діапазон завдань, які здатна вирішити SIEM-система, дійсно дуже широкий. По-перше, про що вже згадувалося раніше, це автоматизація моніторингу й аналізу всіх подій, які відбуваються в численних системах захисту. Друге важливе завдання, цілей, заради якої використовуються SIEM-технології: у випадку інциденту SIEM здатні надати всю необхідну доказову базу, придатну як для внутрішніх розслідувань, так і для суду. Третє важливе призначення – SIEM допомагає проводити аудитів на відповідність різним галузевим стандартам.

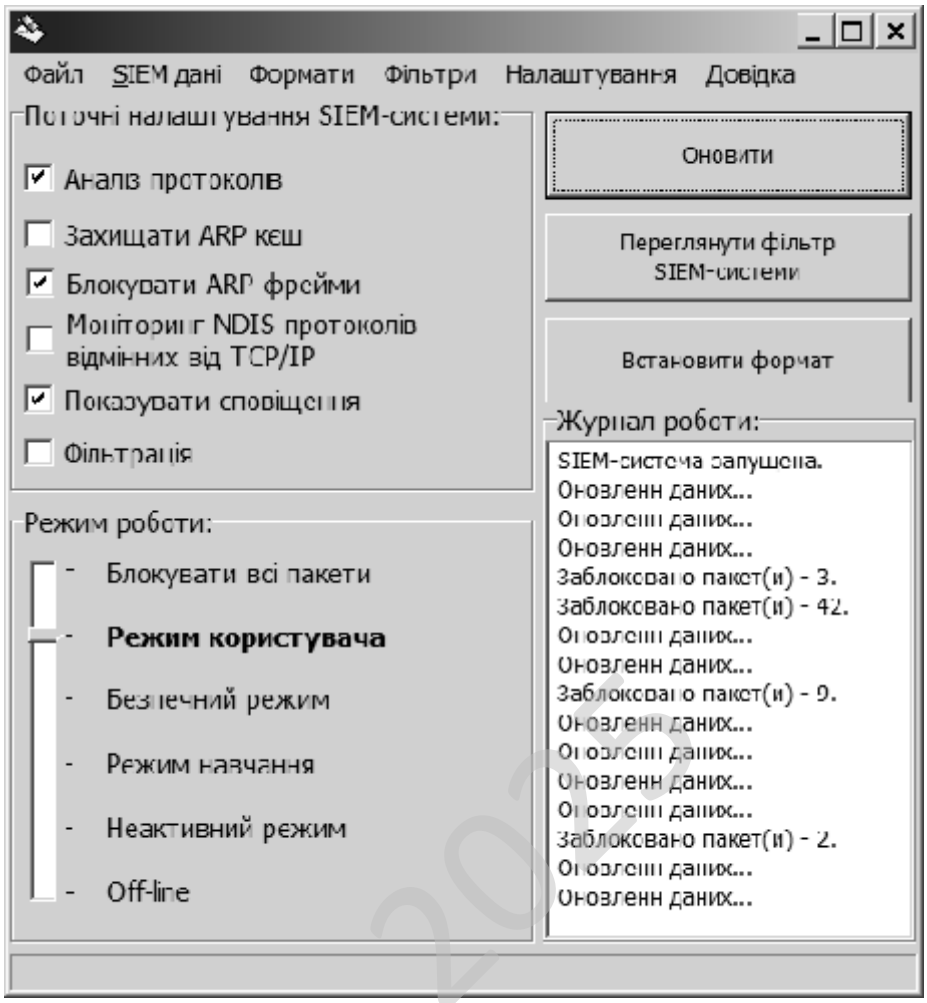

Рисунок 5.1 – Головне вікно ПЗ

Варто відзначити, що SIEM можна назвати інструментом не тільки відділу ІБ, але й ІТ-департаменту в цілому. Адже завдяки потужним кореляційним механізмам з'являється можливість забезпечувати безперервність роботи ІТ-сервісів, виявляти збої в роботі інформаційних і операційних систем, а також апаратного забезпечення. Тим самим, можна забезпечити безперервність бізнесу в цілому. Простий приклад, актуальний для більшості корпоративних мереж: конфлікт IP-адрес. За рахунок найпростішого правила можна довідатися про інцидент задовго до дзвінка користувача. При цьому усунення причини вимагає набагато менше часу, а отже, зменшуються можливі фінансові втрати бізнесу.

Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним

середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

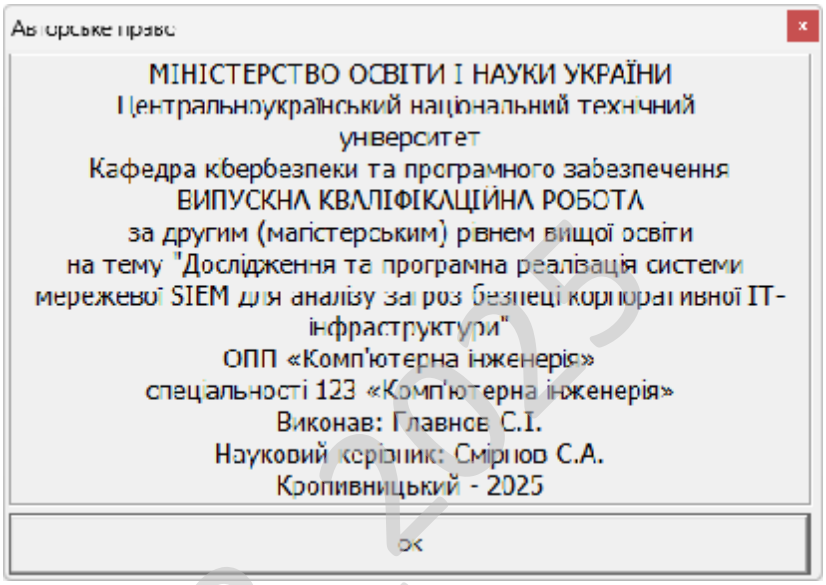

Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити.

Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

– Випуск.

– Встановлення та активація.

– Деактивація.

– Адаптація.

– Обновлення.

– Вмонтування.

– Відстежування версій.

– Видалення.

– Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників.

Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

– відповідності поставленим вимогам;

– правильна відповідь для усіх можливих вхідних даних;

– виконання функцій за прийнятний час;

– практичність;

– сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

– Гарантується перевірка всіх незалежних маршрутів програми.

– Знаходяться гілки True, False для всіх логічних рішень.

– Виконуються всі цикли (у межах їхніх кордонів та діапазонів).

– Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

– Кількість незалежних маршрутів може бути дуже велика.

– Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

– У програмі можуть бути пропущені деякі маршрути.

– Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

– Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

– Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

– При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Обрано умови розповсюдження – proprietary software.

Програмне забезпечення, на яке зберігаються як немайнові, так і майнові авторські права. Отримавши або придбавши таке програмне забезпечення, користувач отримує обмежені права користування ним: може бути заборонено або закрито доступ до коду (вивчення), внесення змін, тиражування, розповсюдження та перепродаж. Програмне забезпечення вважається власницьким, якщо наявне хоча б одне з перелічених обмежень.

Найчастіше основним методом захисту майнових прав на власницьке ПЗ, поза ліцензійною угодою, власник обирає закриття сирцевого коду, захищаючи свій продукт від модифікації і вбудовуючи системи обмеження користування через авторизацію. Таке програмне забезпечення називається закритим. Проте, код власницького продукту може бути і відкритим, але власник може обмежити права користувача умовами користувацької ліцензії.

Власницьке програмне забезпечення та комерційне програмне забезпечення не є синонімами – власницьким може бути і безплатне (тобто, некомерційне) програмне забезпечення.

На противагу власницькому ПЗ існує вільне програмне забезпечення, автори і власники якого дозволяють вивчати, модифікувати і поширювати свій продукт. Саме визначення власницького програмного забезпечення виникло в результаті діяльності громадського руху вільного програмного забезпечення (представленого Фондом вільного програмного забезпечення та іншими організаціями) і осмислення умов свободи користування програмами. Визначенням власницького програмного забезпечення є не невідповідність хоча б одній з базових умов вільного програмного забезпечення. Сама назва власницьке ПЗ підкреслює визначальне значення власника у способі використання і можливостях розвитку цього програмного забезпечення.

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Метою розробки є дослідження та програмна реалізація системи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Об'єктом дослідження є процес мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Предметом дослідження є методи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Методи дослідження базуються на методах захисту інформації у мережі, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

– Розроблено вітчизняний продукт мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури, який має більш широкі можливості, на відміну від існуючих аналогів.

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та розробки системи мережевої SIEM (Security Information and Event Management) можуть бути цікавими насамперед компаніям, які працюють у сфері інформаційних технологій, кібербезпеки та управління корпоративними даними. У сучасних умовах, коли кібератаки стають дедалі складнішими, а кількість інцидентів зростає, системи SIEM набувають особливої важливості. Для великих підприємств, які мають складну ІТ-інфраструктуру, така система дозволяє централізовано відстежувати події безпеки, швидко реагувати на підозрілі активності та мінімізувати можливі ризики. Саме тому результат проєкту може зацікавити як технічних фахівців, так і керівників відділів інформаційної безпеки.

Державні установи також можуть бути зацікавлені у впровадженні подібних рішень, адже вони зберігають великі обсяги конфіденційної інформації. Для них система SIEM є не лише технічним інструментом, а й частиною національної стратегії кіберзахисту. Вона допомагає відповідати вимогам законодавства щодо захисту даних та забезпечувати стабільність роботи державних сервісів.

Освітні заклади та наукові установи можуть розглядати систему як навчальну або дослідницьку платформу. Вона дозволяє моделювати загрози, аналізувати типові вразливості та вивчати поведінку атак у контрольованому середовищі. Це робить систему цінним ресурсом для підготовки майбутніх спеціалістів із кібербезпеки.

Крім того, малі та середні підприємства, які часто не мають власного відділу ІТ-безпеки, також можуть бути потенційними користувачами. Хмарна

версія або SaaS-модель такої системи дозволила б їм отримати доступ до високого рівня захисту без значних інвестицій у обладнання та персонал. Таким чином, аудиторія потенційних зацікавлених сторін охоплює як великі корпорації, так і невеликі бізнеси, які прагнуть підвищити рівень цифрової безпеки.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для визначення привабливості системи мережевої SIEM було проведено експертне опитування серед фахівців у галузі інформаційної безпеки, адміністраторів IT-інфраструктури та представників бізнесу, які працюють із захистом даних. Кожен експерт оцінював систему за критеріями ефективності виявлення загроз, швидкості аналізу подій, масштабованості, зручності користування, а також економічної доцільності впровадження. За результатами аналізу середня інтегральна оцінка привабливості склала 9,1 бала з 10, що свідчить про високу потенційну цінність розробки.

Фахівці особливо високо оцінили здатність системи автоматично збирати журнали подій із різних джерел – серверів, мережевих пристроїв і додатків – та аналізувати їх у реальному часі. Це дозволяє виявляти нетипову поведінку користувачів або підозрілі активності ще до того, як вони переростуть у серйозні інциденти. Експерти також відзначили зручність графічного інтерфейсу, який дозволяє навіть неспеціалістам у кібербезпеці швидко орієнтуватися в аналітичних звітах.

Високі оцінки система отримала за гнучкість налаштувань, що дозволяє адаптувати її під специфіку різних підприємств – від банків до виробничих компаній. Разом із цим, експерти визнали важливою перевагою можливість інтеграції з іншими рішеннями для моніторингу безпеки. Такий підхід забезпечує більш комплексний захист, що є вагомим фактором у конкурентному середовищі.

Результати експертного оцінювання свідчать, що система має високий потенціал комерційного впровадження. Вона поєднує в собі сучасні технології аналізу даних, автоматизацію процесів та зручність для користувача, що робить її привабливою для широкого кола підприємств, які прагнуть посилити контроль над інформаційною безпекою.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи SIEM найдоцільніше застосувати витратний метод у поєднанні з елементами дохідного підходу. Витратний метод дозволяє визначити загальні витрати на розробку, тестування, впровадження та технічну підтримку системи. Це включає оплату праці команди розробників, вартість ліцензування додаткових компонентів, витрати на інфраструктуру та обладнання. Такий підхід забезпечує реалістичне розуміння собівартості продукту.

Однак, враховуючи комерційний потенціал SIEM-системи, доцільно також оцінити її з точки зору доходів, які вона може принести. Використовуючи дохідний підхід, можна розрахувати потенційний прибуток від продажу ліцензій, підписок або надання послуг SaaS. Це дозволяє не лише визначити цінність системи для розробника, а й прогнозувати її рентабельність для інвесторів.

Поєднання цих двох підходів дає найповнішу картину вартості, адже охоплює як витратну, так і прибуткову складову. Завдяки цьому можна визначити оптимальну ціну продукту, яка забезпечить окупність протягом короткого періоду, не знижуючи при цьому доступності для кінцевих користувачів.

Таким чином, комбінований метод оцінки вартості дозволить приймати зважені рішення як у межах економічного планування розробки, так і в процесі подальшого маркетингового просування. Він особливо актуальний для проєктів у сфері кібербезпеки, де інвестиції в розробку часто значні, але віддача може бути високою завдяки довготривалому використанню продукту.

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

Результати розрахунку зведемо до таблиці 7.1.

Таблиця 7.1 – Економічна ефективність впровадження системи

Показник	До впровадження	Після впровадження	Економічний ефект
Кількість інцидентів безпеки на рік	12	3	–9
Середні втрати від одного інциденту	50 000 грн	15 000 грн	–35 000 грн
Річні витрати на аудит і моніторинг	180 000 грн	120 000 грн	–60 000 грн
Витрати на впровадження системи (одноразово)	—	—	350 000 грн
Річний економічний ефект	—	—	375 000 грн
Термін окупності	—	—	0,93 року (~11 місяців)

Упровадження системи SIEM дозволяє знизити кількість інцидентів безпеки у чотири рази, зменшити витрати на аудит на третину та окупити витрати менш ніж за рік. Крім того, компанія отримує нематеріальні переваги: зростання

рівня довіри клієнтів, підвищення ефективності реагування на загрози та покращення корпоративної репутації.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Просування проєкту системи SIEM має починатися з глибокого розуміння її цільової аудиторії. Основними користувачами є компанії, що мають складну мережеву інфраструктуру та обробляють великі обсяги конфіденційних даних. Перший крок – створення демонстраційної версії системи або пілотного проєкту, який дозволить потенційним клієнтам побачити реальні результати аналізу загроз і оцінити ефективність автоматизації процесів. Візуалізація даних, приклади реальних сценаріїв реагування та наочні графіки стануть переконливим аргументом на користь рішення.

Далі слід зосередитись на комунікації через професійні спільноти та конференції з кібербезпеки. Виступи на форумах, публікації аналітичних матеріалів або кейсів використання системи дадуть змогу сформувати довіру серед фахівців галузі. Водночас важливо підтримувати онлайн-присутність через офіційний сайт, сторінку проєкту в LinkedIn, спеціалізовані форуми та освітні платформи. Це забезпечить доступність інформації про продукт для широкого кола користувачів.

На наступному етапі можна запропонувати гнучку модель ліцензування. Наприклад, надати можливість безкоштовного тестового періоду для компаній, які бажають оцінити функціонал перед купівлею. Такий підхід сприяє формуванню лояльності та довіри. Важливо також підготувати набір матеріалів для технічних директорів і IT-менеджерів, у яких чітко показати, як система скорочує час реагування на загрози, знижує кількість інцидентів і дозволяє ефективніше використовувати ресурси команди безпеки.

Фінальним кроком має стати формування партнерських програм з провайдерами хмарних сервісів, консалтинговими компаніями та інтеграторами

ІТ-рішень. Це розширить канали просування системи, дозволяючи їй виходити на нові ринки без значних маркетингових витрат. Також варто забезпечити підтримку користувачів і регулярні оновлення продукту – це допоможе не лише утримати клієнтів, а й створити довгострокову екосистему навколо розробки.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту системи SIEM повинна бути побудована на поєднанні прямих і партнерських шляхів реалізації. Основну ставку варто зробити на співпрацю з компаніями, які вже працюють у сфері ІТ-консалтингу або надають послуги з адміністрування корпоративних мереж. Вони можуть виступати посередниками, які пропонуватимуть систему своїм клієнтам як частину комплексних рішень із кібербезпеки. Така стратегія дозволяє не лише зменшити витрати на прямий маркетинг, а й розширити охоплення аудиторії через уже сформовану базу довіри партнерів. Також ефективним інструментом може стати використання моделі підписки (SaaS). Це дає змогу компаніям почати користуватись системою без великих одноразових витрат, сплачуючи за фактичне використання. Такий підхід особливо привабливий для малого та середнього бізнесу, який часто має обмежений бюджет на безпеку, але потребує якісного захисту. Ще одним напрямом оптимізації є створення відкритого API, що дозволить іншим розробникам інтегрувати SIEM у власні рішення. Це розширює потенційну клієнтську базу та створює додатковий ефект взаємодії з іншими системами безпеки. Крім того, варто приділити увагу роботі з навчальними закладами – співпраця з університетами може не лише розширити знання про продукт, а й підготувати майбутніх фахівців, які у своїй професійній діяльності обиратимуть знайомі технології. Загалом, оптимізація збуту повинна спиратись на поєднання партнерських програм, цифрових каналів комунікації та освітніх ініціатив. Такий підхід створює не лише ринкову присутність продукту, а й підвищує його цінність у професійному середовищі.

7.7 Визначення ключових факторів успіху конкретного проєкту

Успіх проєкту системи мережевої SIEM багато в чому визначається її здатністю забезпечити надійність, точність і масштабованість. У першу чергу, система має ефективно збирати й аналізувати величезні обсяги даних у режимі реального часу, виявляючи потенційні загрози до того, як вони завдадуть шкоди. Висока продуктивність і мінімальна кількість хибних спрацьовувань – це ті показники, які безпосередньо впливають на рівень довіри користувачів і конкурентоспроможність продукту. Не менш важливим є зручний інтерфейс і можливість адаптації системи під специфіку конкретного підприємства. Якщо рішення легко інтегрується у вже існуючу IT-інфраструктуру, воно стає привабливим як для технічних фахівців, так і для управлінців, які цінують простоту впровадження. Система, яка не потребує складного налаштування, має більші шанси на швидке поширення. Стабільне оновлення бази загроз і постійна підтримка користувачів також є визначальними факторами успіху. У сфері кібербезпеки зволікання може мати серйозні наслідки, тому здатність оперативно реагувати на нові виклики робить продукт конкурентним і надійним.

Не можна оминути увагою й репутацію розробників. Відкрита комунікація з клієнтами, публічність результатів тестування, участь у профільних конференціях – усе це формує довіру до компанії й підвищує авторитет продукту на ринку. Коли клієнти бачать, що система розвивається, отримує оновлення і вдосконалюється, вони сприймають її як живий, надійний і перспективний інструмент. Таким чином, поєднання технічної якості, гнучкості, професійної підтримки та репутаційної стабільності формує основу успіху будь-якої SIEM-системи. Це не просто програмний продукт, а стратегічне рішення, яке допомагає компаніям вибудовувати культуру безпеки та впевнено рухатись у цифровому середовищі.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона здоров'я працівників, забезпечення безпеки умов праці, ліквідація професійних захворювань і виробничого травматизму повинна складати одну з головних завдань роботодавця.

Основою охорони праці є науковий аналіз умов праці, технологічних процесів, виробничого обладнання, робочих місць, трудових операцій, організації виробництва з метою виявлення шкідливих і небезпечних виробничих факторів, їх властивостей, особливостей виливу на організм людини. На підставі такого аналізу розробляються заходи та засоби, спрямовані на мінімізацію несприятливого впливу виробничих факторів, створення безпечних та нешкідливих умов праці.

Основою охорони праці є науковий аналіз умов праці, технологічних процесів, виробничого обладнання, робочих місць, трудових операцій, організації виробництва з метою виявлення шкідливих і небезпечних виробничих факторів, їх властивостей, особливостей виливу на організм людини. На підставі такого аналізу розробляються заходи та засоби, спрямовані на мінімізацію несприятливого впливу виробничих факторів, створення безпечних та нешкідливих умов праці

Для того, щоб об’єктивно проаналізувати відповідність умов праці діючим нормативно-правовим актам, необхідно здійснити санітарно-гігієнічну характеристику умов праці відділу, в якому працює програміст, над розробкою даного програмного продукту.

В зв’язку з цим необхідно сконцентрувати увагу на небезпечних і шкідливих чинниках пов’язаних з постійною роботою за комп’ютером.

Електробезпека є одним із критичних питань для співробітників, що працюють із технікою, яка одержує живлення з електричної мережі. При невиконанні норм електробезпеки можливе враження електричним струмом.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машина (ПЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють програмісти необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

При роботі з використанням ПЕОМ відзначають наступні небезпечні та шкідливі фактори:

– ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території;

– ризик виникнення пожежі;

– негативний вплив на органи зору людини;

– ризики ураження електричним струмом;

– недостатня, або надмірна освітленість робочого місця;

– електромагнітні (у тому числі високочастотні) випромінювання (коливання);

– несприятливі мікрокліматичні умови;

– нервово-емоційна напруженість праці;

– інтелектуальні навантаження;

– монотонність праці;

– невідповідність ергономічних показників робочого місця діючим вимогам;

– шум;

– статичні навантаження на кістково-м’язовий апарат;

Відповідно до ст.14 Закону «Про охорони праці» [1] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування засобів індивідуального захисту працівників; відповідні вимоги охорони праці, умови праці на кожному робочому місці; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам виконання робіт; інструктаж з охорони праці; організацію контролю над станом умов праці на робочих місцях; проведення атестації робочих місць за умовами праці.

Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – одна з головних задач охорони праці.

8.3 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

– розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до

колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга) [9].

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

8.4 Пожежна безпека

Вимоги до пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам.

Зокрема, вказуються конкретні території, ділянки, зони, об'єкти, цілі будівлі і їх частини, поверхи, на яких відповідальний співробітник повинен проводити такі організаційні роботи.

Відповідальні особи зобов'язуються розробити, впровадити та підтримувати виконання певних інструкцій і положень на ввірених їм об'єктах. протипожежний режим відповідно до вимог, викладених в нормативних актах.

Передбачено також створення підрозділу добровільної пожежної охорони та пожежно-рятувальної команди в його складі.

Встановлений режим включає порядки з описом місць спеціального призначення та правила їх використання та утримання, наприклад:

– евакуаційних шляхів;

– так званих «курилок»;

– місць складування продукції та сировини;

– стоянки транспорту.

Також встановлюється порядок роботи та технічного обслуговування:

– вентиляційного устаткування;

– засобів пожежогасіння і захисту від загорянь;

– нагрівальних приладів;

– електрообладнання.

Розробляються і впроваджуються правила роботи з відкритим вогнем і горючими матеріалами. Створюються графіки проходження інструктажів з пожежної безпеки співробітників, а також порядок і терміни перевірок знань пожежно-технічного мінімуму, в тому числі, тих працівників, які відповідальні за цю ділянку роботи на підприємстві. При цьому можуть передбачатися внутрішні лекції, семінари, тренінги та практичні заняття на підприємстві, а також зовнішні – на базі спеціалізованих навчальних центрів з професійними викладачами.

Важливою складовою протипожежного режиму на будь-якому об’єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електроустановки, що і в якій послідовності необхідно робити співробітникам.

Відповідно, для кожного об’єкта, кожного приміщення (крім коридорів, санвузлів, басейнів і подібних приміщень), окремих видів робіт складаються інструкції, за якими повинен працювати персонал, залучений на певних ділянках і в виконанні окремих видів робіт. За інструкціями проводиться навчання (інструктаж) персоналу з подальшим контролем знань. Детально про те, як розробити протипожежний режим, прописати порядки та інструкції, пояснюють на тематичних курсах і семінарах. [4] Відповідно ДБН В 1.1-7-2016 «Пожежна безпека об’єктів будівництва» будинок можна віднести до II групи за ступенем вогнестійкості й до категорії Д за ступенем пожежонебезпеки.

Від розподільного щита по праву й ліву сторони встановлені кондиціонери, зовнішня електропроводка поміщена в ізольований кабель. Висота проводки становить 2,2 м від рівня підлоги. Біля кожного столу організований розподільний щит, розташований на текстолітовій пластинці, закріпленій на стіні на рівні 1м від підлоги. Усього до складу входять п'ять розеток і дві клеми заземлення. Всі обчислювальні машини з'єднані із клемами заземлення. Чотири з п'яти розеток забезпечують подачу напруги 220 V, а одна, забезпечує подачу напруги в 36 V. Про це є відповідні написи на кожному розподільному щиті.

Робота обслуговуючого персоналу полягає в інсталяції необхідного програмного забезпечення й наступному його використанні в діалоговому режимі роботи з ЕОМ. Іноді може виникати необхідність написання допоміжних програм для поліпшення роботи вузла або для зниження витрат. З погляду забезпечення умов праці й вимог техніки безпеки для роботи програміста необхідно наступне: достатнє висвітлення екрана дисплея й робочого місця; повна технічна справність устаткування, його електробезпечність; достатня пожежна безпечність приміщення; оптимальний мікроклімат, що сприяє продуктивній роботі; відповідність робочого місця вимогам ергономіки. До небезпечних і шкідливих факторів, дії яких піддається програміст, можна віднести: можливість поразки електричним струмом, при електронесправності устаткування, порушенні заземлення або техніки безпеки; робота в мікрокліматі з неприпустимими параметрами; робота при недостатній освітленості екрану дисплея й робочого місця.

Відповідно НПАОП 40.1-1.21-98 “Правил безпечної експлуатації електроустановок споживачів” [6], приміщення можна віднести до приміщень без підвищеної небезпеки, оскільки це приміщення сухе, з нормальною температурою й ізолюючими підлогами, що не має заземлених металоконструкцій.

Персональні ЕОМ можна віднести до першого класу електротехнічних виробів за способом захисту людини від поразки електричним струмом, оскільки

їхні корпуси зроблені з ізолюючої пластмаси й кожен пристрій має заземлення. Відповідно правилам пристрої електроустановок ЕОМ можна віднести до електроустановок з робочою напругою до 1000 В.

Однією з достовірних причин пожежі в приміщенні з обчислювальною технікою може бути коротке замикання, що спричиняє спалах електропроводки. Для його попередження вся обчислювальна техніка, а також інші електричні пристрої повинні бути обладнані плавкими запобіжниками, а на вході електромережі повинен бути передбачений автомат захисту. Не слід користуватися електричними подовжувачами й трійниками, що не мають сертифікатів відповідності вимогам безпеки.

Необхідно передбачити наявність у межах досяжності первинних засобів гасіння пожежі (вогнегасників) для локалізації вогню власними силами до приїзду команди пожежної охорони. Повинен бути розроблений план екстреної евакуації персоналу при виникненні загоряння. Кількість евакуаційних виходів повинне бути не менш двох. Допускається використання одного евакуаційного виходу, якщо відстань найбільш віддаленого робочого місця до цього виходу не перевищує 25 м.

8.5 Розрахункова частина

Запорукою безпечної роботи в ІТ-сфері є виконання вимог електричної безпеки, оскільки все офісне обладнання заживлюється від електричної мережі. Одним з необхідних засобів електричної безпеки є встановлення захисного заземлення.

Початкові дані, необхідні для розрахунку захисного заземлення:

– допустимий опір розповсюдженню струму в землі від заземлю-вального пристрою $R_{зн} = 10$ Ом;

– питомий опір грунту в місці встановлення заземлювача $\rho_з = 100$ Ом/м;

– тип грунту – суглинок;

– тип заземлювача – труба, діаметром d=0.058 м і довжиною l =2.1 м; – конструкція заземлювача – розташування заземлювачів по контуру. Розрахунок проводимо за стандартною методикою [7].

Визначимо розрахунковий опір землі:

$$\rho_{рз} = \phi \cdot \rho_3$$

де ϕ – коефіцієнт сезонності, що враховує коливання питомого опору при зміні вологості ґрунту протягом року; при використанні заземлювача довжиною l = 2.1 м при глибині закладання від вершини h = 0.4 м ϕ = 1.1 для четвертої кліматичної зони.

Схема розташування заземлювачів показана на рисунку 8.1.

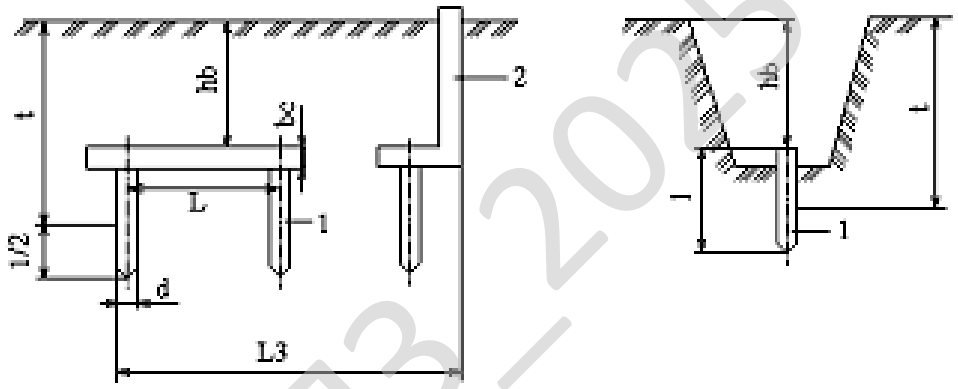

Рисунок 8.1 – Схема розташування заземлювачів

Опір землі:

$$\rho_{рз} = 1{,}1 \cdot 100 = 110 \text{ Ом·м}$$

Опір Rв, розповсюдженню струму в землі від одного вертикального заземлювача:

$$R_B = \frac{\rho_{рз}}{2\pi \cdot l}\left(\ln\frac{2 \cdot l}{d} + 0.5\ln\frac{4t + l}{4t - l}\right)$$

де

l – довжина заземлювача (l = 2.1 м);

d = 0.058 м – діаметр заземлювача при U<1 кВ та при S<100 кВА;

t – відстань від поверхні до середини заземлювача:

$$t = h+l/2=0.4+2.1/2=1.45 \text{ м.}$$

$$R_{в} = \frac{110}{2\cdot 3.14\cdot 2.1}\left(\ln\left(\frac{2\cdot 2.1}{0.058}\right)+0.5\cdot\ln\left(\frac{4\cdot 1.45+2.1}{4\cdot 1.45-2.1}\right)\right) = 38.72 \text{ Ом}$$

Визначаємо потрібну кількість заземлювачів:

$$n' = \frac{R_в}{R_{зн}} = \frac{38.72}{10} = 3.9 \approx 4 \text{ шт.}$$

Коефіцієнт використання вертикальних заземлювачів враховує ефект екранування. При вибраному значенні k = a/l, де a – відстань між вертикальними заземлювачами, м; k = 1 при a = 1.5 м. Коефіцієнт використання вертикального заземлювача за довідковими даними дорівнює $\eta_В = 0,6$.

Кількість вертикальних заземлювачів з урахуванням коефіцієнту використання $\eta_В$ приблизно складає

$$n = \frac{R_в}{R_{зн}\cdot\eta_в} = \frac{38.72}{10\cdot 0.6} = 6.45 \approx 7 \text{ шт.}$$

Довжина горизонтального заземлювача, необхідна для розміщення вертикальних заземлювачів по контуру

$$L = a\cdot n = 1.5\cdot 7 = 10.5 \text{ м}$$

Опір горизонтального заземлювача $R_Г$, Ом, прокладеного на глибині h = 0.4 м від поверхні землі буде

$$R_{г} = \frac{R_{рз}}{2\cdot 3.14\cdot L}\cdot\ln\frac{2\cdot L^2}{b\cdot h} = \frac{110}{2\cdot 3.14\cdot 10.5}\cdot\ln\frac{2\cdot 10.5^2}{0.058\cdot 0.4} = 15.7 \text{ Ом}$$

де b = 0.04 м – ширина сталевої смуги, з якої виготовлений заземлювач.

Обчислюємо загальний опір:

$$R_{з} = \frac{R_{в}\cdot R_{г}}{n\cdot R_{г}\cdot\eta_в + R_в\cdot\eta_г} = \frac{38.72\cdot 15.7}{6\cdot 15.7\cdot 0.6+38.72\cdot 0.34} = 8.13 \text{ Ом}$$

де $\eta_Г$ – коефіцієнт використання горизонтального заземлювача ($\eta_Г = 0.34$).

Маємо 8.13 Ом < 10 Ом, отже нормативне обмеження для потужності генераторів та трансформаторів 100 кВт і менше $R_з < R_{з.норм}$ виконується.

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

– Досліджена система мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

– На основі отриманих результатів досліджень створена програмна реалізація системи мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання мережевої SIEM для аналізу загроз безпеці корпоративної IT-інфраструктури.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створені програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Visual C++. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм FEAL.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проекту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Главнов С.І. Дослідження та програмна реалізація системи мережевої SIEM для аналізу загроз безпеці корпоративної ІТ-інфраструктури // Збірник праць молодих науковців ЦНТУ. – Вип. 15. – Кропивницький: ЦНТУ, 2025.

2. Awais Rashid, Howard Chivers, George Danezis, Emil Lupu, Andrew Martin. CyBOK The Cyber Security Body of Knowledge. The National Cyber Security Centre. 2019. 854 p.

3. Loren Kohnfelder. Designing Secure Software. No Starch Press. 2022. 332 p.

4. Samir Kumar Rakshit. Ethical Hacker's Penetration Testing Guide. BPB Online. 2022. 509 p.

5. Corey J. Ball. Hacking APIs. No Starch Press. 2022. 353 p.

6. Kevin Beaver. Hacking for Dummies. John Wiley & Sons. 2022. 419 p.

7. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p

8. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.

9. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.

10. Alex Matrosov, Eugene Rodionov, Sergey Bratus. Rootkits and Bootkits. No Starch Press. 2019. 450 p.

11. Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед та ін. «Системи інформаційної зброї та технології інформаційної війни»: підручник / Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед, Жарков Я.М., Смірнов О.А, Буравченко К.О., Давидюк А.В., Кононович В.Г., Корчинский В.В., Кудирко В.М., Фесенко А.О.; за заг. ред. В.М. Петрика, М.М. Присяжнюка.– К.: Видавничий центр “Кафедра”, 2025.– 320 с.

12. Усік, П.С., Смірнова, Т.В., Буравченко, К.О., Смірнов, О.А., Улічев,

О.С., Смірнов, С.А. «Дослідження технологій забезпечення кібербезпеки банківських систем з використанням штучного інтелекту». *Кібербезпека: освіта, наука, техніка.* 2025. Том 1 № 29. С.704–716, 2025

13. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Arnesano, M., Smirnov, O. «A secure biometric authentication architecture for blockchain-driven cyber-physical systems». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 193–224.

14. Kuznetsov, O., Atzeni, G., Arnesano, M., Randieri, C., Smirnov, O. «Secure IoT-based smart wheelchair system: From implementation to security enhancement strategy». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 225–257.

15. Kuznetsov, O., Smirnov, O., Kuznetsova, T., Shaikhanova, A., Svatowsky, I. «Privacy-utility trade-offs in IoT networks: A comparative analysis of differential privacy mechanisms for sensor data aggregation». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 589–622.

16. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with Fuzzy Information». *8th International Symposium on Intelligent Informatics, ISI 2023*, 2025. vol 389. pp 377-389. Springer, Singapore.

17. Kuznetsov O., Frontoni E., Kuznetsova Y., Smirnov O., Moskovchenko I. «Trust-Based Security Architecture for Edge Computing: A Simulation Study of Dynamic Trust Evolution and Attack Detection». *CEUR Workshop Proceedings,* 2024, 3909, pp. 227–241.

18. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.

19. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447.

20. Ткаченко, О., Ільєнко, А., Улічев, О., Мелешко, Є., Смірнов, О. «Правові засади поширення інформаційних впливів в соціальних мережах». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»,* 2024. № 2(26), С. 170–188.

21. Смірнова Т.В., Конопліцька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка.* 2024. №4(24), С. 6-27.

22. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка.* 2024. №3(23), С. 111-131.

23. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології,* 2024, № 13, с. 28-35.

24. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

25. Akhalaia, G., Iavich, M., Iashvili, G., Prysiazhnyy, D., Smirnova, T. «Secure Encrypted Connection on Georgian Website». *CEUR Workshop Proceedings,* 2023, 3550, pp. 313-320.

26. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56

27. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings,* Volume 3530, 2023, pp. 256-265.

28. Kuznetsov, O., Kandiy, S., Frontoni, E., Smirnov, O. «Trade-offs in Post-Quantum Cryptography: A Comparative Assessment of BIKE, HQC, and Classic McEliece». *CEUR Workshop Proceedings,* Volume 3504, 2023, pp. 1-11.

29. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». *In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies*, vol 131. 2023. Springer, Singapore. pp. 21-34.

30. Смірнова Т.В., Гнатюк С.О., Бердибаєв Р.Ш., Сидоренко В.М., Жигаревич О.К., «Система корелювання подій та управління інцидентами кібербезпеки на об'єктах критичної інфраструктури». *Кібербезпека: освіта, наука, техніка*, №*3*(19), 2023, С. 176-196.

31. Смірнов О.А. Козлов Я.О., Смірнова Т.В. «Дослідження застосування SIEM-систем для забезпечення кібербезпеки та захисту інформації». *ІІ Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи hозвитку інформаційних технологій (ІПШРІТ-2023)»* м.Черкаси 6 грудня 2023 року – Черкаси: ЧДТУ.– 2023. – С.251-252.

32. Козлов Я.О., Смірнова Т.В., Смірнов О.А. «Дослідження SIEM-систем для забезпечення кібербезпеки». *VII міжнародна науково-практична конференція "Інформаційна безпека та комп'ютерні технології" до 30-ти річчя кафедри кібербезпеки та програмного забезпечення,* м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 26.

33. Козлов Я.О., Козірова Н.Л., Смірнов О.А. «Дослідження структури та принципу роботи SIEM-системи». *VII міжнародна науково-практична конференція "Інформаційна безпека та комп'ютерні технології" до 30-ти річчя*

кафедри кібербезпеки та програмного забезпечення, м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 59.

34. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

35. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings,* Volume 3187, 2022,

36. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». *Journal of Ambient Intelligence and Humanized Computing* Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477.

37. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління,* № 2(70). 2022. С. 28-37.

38. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку,* 2022, № 3(69). С. 93-98.

39. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»,* № 2 (307). С. 46-52. 2022.

40. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А.,

Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

41. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021,* Cracow, Poland, 22-25 September 2021. P. 414-418

42. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». *4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) – 2021*, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

43. Smirnov O., Kuznetsov A., Girzheva O., Kiian A., Nakisko O., Kuznetsova T. «Advanced Code-Based Electronic Digital Signature Scheme». *2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020,* Kharkiv, 6 October 2020-9 October 2020, P. 358-362.

44. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». *CEUR Workshop Proceedings* Volume 2805, 2020, Pages *44-58.*

45. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». *International Journal of Computing*; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

46. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

47. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and

cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

48. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131.

49. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 1-14.

50. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». *Lecture Notes in Networks and Systems*, vol 152. Springer, Cham. 2021, pp 66-84.

51. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

52. Smirnov O., Kuznetsov A., Kiian A., Babenko V., Perevozova I.,Chepurko I. «New Approach to the Implementation of Post-Quantum Digital Signature Scheme». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT),* Ukraine, Kyiv, May 14-18. 2020. P. 166-171.

53. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT),* Ukraine, Kyiv, May 14-18. 2020. P. 172-177.