

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
“___” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи Intrusion Prevention System
для протидії несанкціонованому мережевому доступу або
атакам на мережу”

Виконав здобувач вищої освіти
IV курсу, групи КІ-23-МБ
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Стеценко В.М.
«___» _____ 2026 р.

Керівник проекту
кандидат фізико-математичних наук, доцент
_____ Якименко Н.М.
«___» _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань 12 "Інформаційні технології"
Спеціальність 123 "Комп'ютерна інженерія"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 27 » лютого 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Стеценку Владиславу Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу

2. Керівник роботи Якименко Наталія Миколаївна, кандидат фізико-математичних наук, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 126-02 від 27.02.2026 року

3. Строк подання студентом роботи до захисту 23.05.2026 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою роботи є розробка програмного забезпечення системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

7. Дата видачі завдання « 27 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника

Якименко Н.М.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Стеценко В.М.
(прізвище та ініціали)

АНОТАЦІЯ

Стеценко В.М. Програмне забезпечення системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу.

Метою розробки є програмне забезпечення системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу.

Результат роботи – програмна реалізація системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу.

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 Python.

Ключові слова: комп'ютерна інженерія, Intrusion Prevention System

ABSTRACT

Stetsenko V.M. Software for the Intrusion Prevention System to counteract unauthorized network access or network attacks. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed that is intended for the Intrusion Prevention System to counteract unauthorized network access or network attacks.

The purpose of the development is the software for the Intrusion Prevention System to counteract unauthorized network access or network attacks.

The result of the work is the software implementation of the Intrusion Prevention System to counteract unauthorized network access or network attacks.

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 tools are provided.

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

The program is developed in the Python environment.

Keywords: computer engineering, Intrusion Prevention System

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	23
2.3 Розгорнута постановка завдання	26
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	27
3.1 Опис функціонування системи	27
3.2 Розробка структурної схеми.....	36
3.3 Розробка функціональної схеми	41
3.4 Розробка діаграми процесів.....	43
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	45
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	45
4.2 Захист розробленого програмного забезпечення.....	68
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	70
6 ОСНОВНІ ВИСНОВКИ.....	75
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	77

					ВКРБ-123.26.0038.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу	Літ.	Аркуш	Аркушів
Розроб.	Стеценко В.М.					Б	1	84
Перев.	Якименко Н.М.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-23-МБ		
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ІБ	—	інформаційна безпека
КМ	—	комп'ютерна мережа
КСАЗ	—	комплексна система антивірусного захисту
МЕ	—	міжмережевий екран
ПЗ	—	програмне забезпечення
ПК	—	персональний комп'ютер
ACL	—	Access Control List
FTP	—	File Transfer Protocol
ETM	—	Enterprise Threat Management
HTTP	—	HyperText Transfer Protocol
IDS	—	Intrusion detection system – системи виявлення вторгнень
IPS	—	Intrusion prevention system – системи запобігання вторгнень
NAC	—	контролю доступу в мережу
NBA	—	поведінковий аналіз мережі
POP3	—	Post Office Protocol Version 3
SMTP	—	Simple Mail Transfer Protocol
VA	—	аналіз уразливостей
VLAN	—	Virtual Local Area Network

ВСТУП

Актуальність теми. У сучасному світі системи виявлення й запобігання вторгнень (Intrusion detection system / Intrusion prevention system, IDS/IPS) – необхідний елемент захисту від мережеских атак. Основне завдання даних систем – виявлення фактів неавторизованого доступу в корпоративну мережу або несанкціоноване керування нею, з виконанням відповідних мір протидії (інформування адміністраторів про факт вторгнення, обрив з'єднання або переналаштування міжмережевого екрана для блокування подальших дій злоумисника й т.д.). Для рішення поставленого завдання IDS повинна виконувати наступні основні функції:

- моніторинг подій з метою виявлення інцидентів інформаційної безпеки (ІБ);
- запис інформації про дані інциденти як локально, так і з відправленням у будь-яку централізовану систему збору логів або SIEM-систему;
- повідомлення адміністраторів ІБ про інциденти (email, SNMP-Трапи, SMS, консоль керування системи IDS);
- створення звітів, що уточнюють або, навпаки, що узагальнюють інформацію з одного або декількох подіям.

Як ми бачимо, IDS виконує лише першу частину завдання (виявлення й інформування). Для рішення другої частини завдання (протидія) можна використовувати систему, що спеціалізується по усуненню конкретної погрози, або використовувати IPS із декілька розширеним функціоналом ніж в IDS:

- блокування атаки (обрив сесії порушника, закриття доступу до ресурсів, хостів, застосунків);
- зміна середи, яка захищається (конфігурування міжмережеских екранів і комунікаційного устаткування);

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

– зміна вмісту атаки (видалення з листа інфікованого файлу, аналіз вхідних запитів і видалення даних із заголовків пакетів).

За способом моніторингу IPS класифікуються на:

- Мережеві IPS (Network-based Intrusion Prevention, NIPS).
- IPS для бездротових мереж (Wireless Intrusion Prevention Systems, WIPS).
- Аналізатор поведінки мережі (Network Behavior Analysis, NBA).
- IPS для хостів (Host-based Intrusion Prevention, HIPS).

На сьогоднішній день у багатьох системах використовується функціонал IPS (міжмережеві екрани, антивіруси, SIEM і т.д.), але найчастіше ці функції або усічені, або для їхнього виконання виділяється дуже мало ресурсів.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу.

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

- Огляд існуючих систем Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу.
- Дослідження системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу.
- Програмна реалізація системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Мережеві системи запобігання вторгнень (IPS) можуть стати як ефективним інструментом для людей, що займаються безпекою, так і дорогим обладнанням, що порохиться без справи. Щоб IPS-система не стала розчаруванням, вона повинна як мінімум відповідати наступним вимогам, які необхідно враховувати при її виборі.

Система повинна:

- мати широке коло моделей, що задовольняють вимогам як маленьких регіональних офісів, так і основного підприємства із багатогігабітними мережами;
- підтримувати не тільки сигнатурний аналіз, але й аномальний аналіз протоколів, і, звичайно ж, поведінковий аналіз;
- давати ясну картину мережі й пристроїв, підключених у неї;
- забезпечувати роботу в режимі IDS, здійснювати поведінковий аналіз, мати інструментарій для проведення розслідувань;
- мати централізоване керування встановленими IPS / IDS-системами;
- мати гарні засоби аналізу для ефективного вдосконалення політик безпеки.

IDS / IPS-системи найчастіше підключають там, де є критичні ресурси. Але крім того, що необхідно блокувати атаки на ці ресурси, варто вести постійний контроль за ними, а саме: знати, які із цих ресурсів уразливі, як змінюється їхнє поведіння в мережі. Тому до IDS / IPS-систем необхідно додати додаткового функціонала, що дозволяє їм захищати необхідні ресурси більш надійно, знижуючи при цьому вартість володіння. Отже, захист може здійснюватися в три фази – IPS, Adaptive IPS, Enterprise Threat Management.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Функціонал кожної наступної фази містить у собі всі функції з попередньої фази й наращується більш новими.

Фаза 1. Ви можете контролювати й/або блокувати атаки, що використовують тисячі уразливостей, тобто це стандартні IPS-сенсори й центри керування ними.

Фаза 2. З'являється можливість дослідження мережі, пріоритезації подій і автоматизації налаштування IPS.

Фаза 3. Повний можливий функціонал для захисту корпоративної мережі до, під час і після атаки.

ETM – це перше втілення усвідомленості того, що, захищаючи інформаційні ресурси, необхідно працювати розумніше, а не більш посилено. З технологічної точки зору ETM – це комбінація чотирьох технологій керування погрозами й уразливостями, об'єднаних у єдине рішення, що управляється централізовано. У результаті це рішення надає більше можливостей, ніж кожний продукт окремо. ETM складається із системи запобігання вторгнень (IPS), поведінкового аналізу мережі (NBA), контролю доступу в мережу (NAC), аналізу уразливостей (VA), підсистеми обміну даними й централізованим керуванням.

1.2 Область застосування

У цей час захист, забезпечуваний файєрволом і антивірусом, уже не ефективна проти мережових атак і малварів. На перший план виходять рішення класу IDS/IPS, які можуть виявляти й блокувати як відомі, так і ще не відомі погрози.

Технології IDS/IPS

Щоб зробити вибір між IDS або IPS, варто розуміти їхні принципи роботи й призначення. Так, завдання IDS (Intrusion Detection System) складається у виявленні й реєстрації атак, а також оповіщенні при спрацьовуванні певного правила. Залежно від типу, IDS уміють виявляти різні види мережових атак,

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

виявляти спроби неавторизованого доступу або підвищення привілеїв, поява шкідливого ПЗ, відслідковувати відкриття нового порту й т.д. На відміну від міжмережевого екрана, що контролює тільки параметри сесії (IP, номер порту й стан зв'язків), IDS «заглядає» усередину пакета (до сьомого рівня OSI), аналізуючи передані дані. Існує кілька видів систем виявлення вторгнень. Досить популярні APIDS (Application protocol-based IDS), які моніторять обмежений список прикладних протоколів на предмет специфічних атак. Типовими представниками цього класу є PHPIDS, що аналізує запити до PHP-Застосунків, Mod_Security, що захищає веб-сервер (Apache), і GreenSQL-FW, що блокує небезпечні SQL-Команди.

PHPIDS блокує неправильні PHP-Запити

Мережеві NIDS (Network Intrusion Detection System) більше універсальні, що досягається завдяки технології DPI (Deep Packet Inspection, глибоке інспектування пакета). Вони контролюють не один конкретний застосунок, а весь трафік, який проходить, починаючи з канального рівня.

Для деяких пакетних фільтрів також реалізована можливість «заглянути усередину» і блокувати небезпеку. Як приклад можна привести проекти OpenDPI і Fwspport. Останній являє собою програму для перетворення бази сигнатур Snort в еквівалентні правила блокування для iptables. Але споконвічно файєрвол заточений під інші завдання, та й технологія DPI «накладна» для движка, тому функції по обробці додаткових даних обмежені блокуванням або маркіруванням строго певних протоколів. IDS усього лише позначає (alert) всі підозрілі дії. Щоб заблокувати атакуючий хост, адміністратор самотійно перенастроює брандмауера під час перегляду статистики. Природно, ні про яке реагування в реальному часі тут мови не йде. Саме тому сьогодні більше цікаві IPS (Intrusion Prevention System, система запобігання атак). Вони засновані на IDS і можуть самотійно перебудовувати пакетний фільтр або переривати сеанс, відсилаючи TCP RST. Залежно від принципу роботи, IPS може встановлюватися «у розрив» або використовувати дзеркалюванні трафіку (SPAN), одержуваного з декількох

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

сенсорів. Наприклад, у розрив встановлюється Hogwash Light BR, що працює на другому рівні OSI. Така система може не мати IP-адреси, а виходить, залишається невидимою для зломщика.

У звичайному житті двері не тільки замикають на замок, але й додатково захищають, залишаючи біля її охоронця, адже тільки в цьому випадку можна бути впевненим у безпеці. В ІТ у якості такого секьюриті виступають хостові IPS), що захищають локальну систему від вірусів, руткітів і злому. Їх часто плутають із антивірусами, що мають модуль проактивного захисту. Але HIPS, як правило, не використовують сигнатури, а виходить, не вимагають постійного відновлення баз. Вони контролюють набагато більше системних параметрів: процеси, цілісність системних файлів і реєстру, запису в журналах і багато чого іншого.

Щоб повністю володіти ситуацією, необхідно контролювати й зіставляти події як на мережевому рівні, так і на рівні хоста. Для цієї мети були створені гібридні IDS, які колектують дані з різних джерел (подібні системи часто відносять до SIM – Security Information Management). Серед OpenSource-проектів цікавий Prelude Hybrid IDS, що збирає дані практично із всіх OpenSource IDS/IPS і розуміючий формат журналів різних застосунків (підтримка цієї системи припинена кілька років назад, але зібрані пакети ще можна знайти в репозиторіях Linux і *BSD).

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Перші два типи систем (IPS – intrusion prevention system & IDS – intrusion detection system) з'явилися в 1986 році як результат наукової праці, і їхні базові принципи дотепер використовуються всюди – у системах запобігання й виявлення, в NGIPS і NGFW – словом у всіх системах, які були згадані в заголовку. У розділі ми розповімо, як IPS/IDS змінювалися згодом, з якими проблемами зіштовхувалися розроблювачі й що можна від них очікувати в майбутньому.

Отже, як ми вже сказали, системи виявлення погроз і системи запобігання погроз з'явилися після написання наукової розділі якоїсь Дороти Деннінг, і називалася ця стаття «Модель виявлення погроз», і завдяки цій розділі Стенфордський Дослідницький Інститут розробив щось за назвою Intrusion Detection Expert System / (IDES). Вільно це можна перевести як експертна система виявлення погроз. Вона використовувала статистичне виявлення аномалій, сигнатури й хостові / користувальницькі профілі для детектування нетипового поведіння в систем. Таким чином, вона могла визначити якщо такі протоколи як FTP або HTTP були використані некоректно й навіть могла визначати атаки з відмовою обслуговування (DoS).

2000 – 2005: виявлення переважніше запобігання

У ранніх 2000-х системи виявлення вважалися гарним тоном. А до цього міжмережеві екрани були дуже ефективні для ландшафту погроз божевільних 90х років. Файєрволи обробляли трафік відносно швидко, тому що в них не було глибокої інспекції пакетів, тобто ви не знали, що це за трафік приходить до вас у

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

мережу – файєрволи реагували тільки на встановлені в правилах (аркушах контролю доступу) порти, протоколи й / або мережеві адреси. На початку 2000-х з'явилися нові атаки, такі як SQL-ін'єкції та інші, і вони моментально завоювали місце на подіумі в арсеналі зломщиків. І от на цьому етапі IDS системи й придалися – а час систем запобігання погроз ще не настало.

У той час деякі організації боялися використовувати IPS тому що така система потенційно могла заблокувати нешкідливий трафік. Як ми більш докладно описували в нашій розділі про IPS і IDS, IPS відноситься «у розрив» і блокує підозрілі з'єднання, повністю розриваючи коннект і зв'язок між що відправляє й приймають сторонами. Але як ви могли зрозуміти, таке з'єднання могло стати підозрілим просто через якусь аномалію в підключенні й грубо говорячи «глюці». Таким чином, IDS системи просто повідомляли про таку аномалію й нічого не блокували, щоб сісадмін міг зреагувати й перевірити – чи правда це щось погане або ж це просто доброякісна аномалія. Із цієї причини в той час ринок для систем запобігання погроз був настільки малий, що існувало всього декілька IPS вендорів. Тобто ідеєю було що потрібно пропускати будь-який трафік, а розберемося, мол, уже після – ризик втрати гарного трафіку був більш страшний погрози злому.

У цей час сигнатури писалися для виявлення експлойтів, але не уразливостей – тобто для кожної уразливості було 100 різних способів експлойта. Як тільки зловмисники знаходили уразливість, вони змушували розроблювачів IDS виходити потім і писати сотні різних сигнатур для експлойтів – всі тільки для того, щоб система виявлення відправила тривогу адміну. І вендори IDS хвастилися кількістю наявних у них сигнатур, начебто це вигідно відрізняло їх від конкурентів – але як ви розумієте, це не було коректним критерієм оцінки. У загальному й цілому, механізми тоді нараховували наступне полчище методів – збіг по паттернам, рядкам, аномаліям і навіть евристичний аналіз.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Прийняття IPS – рік 2005

Коли в 2005 році системи запобігання почали стає популярніше, більша кількість вендорів стали змагатися за місце під сонцем на зростаючому ринку, і перестали хвастати самими довгими сигнатурами. Знову ж, через установку «у розрив», клієнти боялися, що всі ці сигнатури будуть сповільнювати мережа, тому що кожне з'єднання повинне бути пропущене через них. Таким чином, було вирішено перемінити вектор написання сигнатур на інші – ті, які будуть базуватися не на експлойті, а на самій уразливості. Було отримано досвідченим шляхом, що якщо в системі більше 3500 сигнатур, те це буде помітно позначатися на продуктивності. Сьогодні виробники усе ще поміщають у систему як нові сигнатури, так і якусь класику уразливостей, що зловмисники можуть використовувати.

2006 – 2010: настає час продуктивних IPS/IDS комбайнів

Вендори, які пропонували гібридні системи, швидко обійшли конкурентів – вони пропонували набагато більше продуктивні системи, аж до 5 Гбіт/сек, і могли мониторити сегментовані мережі, DMZ, серверні ферми з веб-додатками й площа усередині периметра. Приміром, сьогодні продуктивні IPS пристрою легко дають більше 40 гігабіт у секунду.

У підсумку, клієнти почали масово переходити на системи запобігання вторгнень і ринок почав дуже швидко рости. А коли з'явився стандарт безпеки PCI DSS почав жадати від організацій підтримати оплати картами установки або IDS, або MCE з можливістю фільтрації веб-застосунків, дуже багато організацій купили гібридні системи. І пройшло вже багато років з моменту народження технології, так що технологію порядно відлагодили, так що, ложно-позитивних спрацьовувань стало набагато менше. Однак, у цей же момент початку розповзатися епідемія ботнетів. І самим популярним способом стало приміщення зловливих застосунків на популярних сайтах, і, якщо який-небудь браузерний плагін начебто Java або Adobe Flash був з уразливістю, при кліку на відповідний документ вредонос тихенько завантажувався на комп'ютер. Крім того, в 2008 році

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

зловмисники активно використовували перенапрямні посилання на шкідливі сайти, так що IDS/IPS вендори почали також додавати списки IP-адрес шкідливих командних центрів і їх веб-адрес – якщо ці ресурси містили на собі вредоноси.

2011 – 2015: системи запобігання вторгнень наступного покоління

У ці роки був переломний момент для вендорів у сфері ІБ – тому що вони стали випускати системи запобігання погроз наступного покоління, які містили в собі такі функції як контроль користувачів і застосунків. Таким чином, традиційний IPS дивиться в мережевий трафік на предмет відомих атак і щось робить із цим трафіком, залежно від моделі розгортання, а IPS наступного покоління робить теж саме, але крім того він покриває набагато більше протоколів (аж до 7 рівня) для захисту від більшої кількості атак. Крім того, він також дозволяє гнучко контролювати доступ до застосунків – тобто, наприклад, щоб можна було лайкати фотки в Instagram, але не можна було їх заливати. І більше того – щоб це могли робити тільки певні групи користувачів.

Наступне доповнення до IDS/IPS систем з'явилося після злому RSA (компанії, що займається мультифакторною автентифікацією) в 2011 році – тоді новинні ресурси назвали це APT (Advanced Persistent Threat)-атакою, тобто складною постійною погрозою. Пізніше було сказано, що це була фішингова атака, у якій утримувався документ із вредоносом усередині. Клієнти стали запитувати ІБ вендорів, чи можуть вони їх захистити від подібних речей, якщо у вендора немає сигнатури на даний конкретний вредонос, і відповіддю вендорів було надання такої функції як емуляція й пісочниці – але це зажадало близько 18 місяців для більшості вендорів. Так що компанії FireEye і Fidelis виявилися у фазі бурхливого росту, тому що вони надавали такі технології пісочниці, до яких усім було дуже далеко. Тільки подумайте, пісочниці вперше за всю історію могли виявити дотепер невідому атаку нульового дня.

Як працює пісочниця: невідомий файл, який виконується, або документ спочатку попадає в пісочницю, де він запускається в різних операційних системах і алгоритм намагається імітувати дії користувача – клавіші стукають, мишка

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

єлозить і клікає, час прокручується – усе в надії на те, що вредонос вилупиться й себе покаже.

Вендори пішли ледве далі. Якщо вредонос себе проявляв, то його хеш-сума (MD5 або SHA) зберігалася для того, щоб у майбутньому завжди ловити такі файли. Відповідно, якщо інший клієнт на такій же системі одержував той же файл – те він не пропускався в мережу й звучала тривога. Такі системи одержали назву Next Generation Firewall – міжмережевих екранів наступного покоління.

Звичайно, Гартнер використовував цей термін ще в 2003 році й пророчив, що вони міжмережеві екрани будуть містити усередині себе складну IPS систему, але індустрія не приймала подібні пристрої аж до 2019 року.

2018 – і далі: міжмережеві екрани наступного покоління

Сьогодні більшість організацій використовують NGFW і список їх функцій тільки росте. Тому що ці МСЕ відрізняються різними функціями, організаціям прийдеться вибрати залежно від точності поставленого завдання і їхніх вимог.

Знову ж, є за й проти МСЕ наступного покоління: за – потрібно купити тільки пару пристроїв замість майже десятка. Проти – це все один вендор, і його мудрість обмежена, тобто не існує кращого вендора, що знав би всі й відразу. У такий спосіб дуже непоганою практикою є комбінувати пристроєм захисту від різних виробників і розбавляти їх «мудрість» між собою. Важливо пам'ятати, що будь-який пристрій захисту завжди добре тільки настільки, наскільки багаті знання й досвід, що коштують за цим пристроєм. Є навіть спеціальний термін – Threat Intelligence. Такі системи й бази знань є у всіх більших ІБ вендорів. Більше того, вони є повністю безкоштовні й відкриті – наприклад, VirusTotal.

Сьогодні ландшафт погроз постійно міняється й більшість вендорів сконцентрована на машинному навчанні, щоб алгоритми аналізу файлів завжди поліпшувалися, а кількість шуму й помилкових спрацьовувань прагнуло до мінімуму.

Але це нескінченна гра в кішки-мишки, і на кожний хід виробників хакери придумують що-небудь нове, що пізніше зможуть нейтралізувати вендори.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Cisco Systems

Надійні рішення, мають відмінну підтримку, але важкі в налаштуванні, сигнатурний аналіз дає багато помилкових спрацьовувань, інтерфейс керування при великій кількості подій не дозволяє адекватно розбирати зафіксовані події. Для повноцінного функціонування необхідні додаткові капіталовкладення в Cisco Security Monitoring, Analysis and Response System (CS-MARS).

TippingPoint

Системи цього виробника зручні в налаштуванні й установці. Мають непоганий інтерфейс керування, але можуть підключатися тільки в розрив, тобто без пасивного виявлення. Не дозволяють розширити функціонал і являють собою просто IDS / IPS-систему.

На одній з конференцій представник TippingPoint у своєму виступі сказав, що їхнє устаткування можна встановити й забути про нього – і така їхня стратегія захисту.

Можливо, хтось її й розділяє, але мені з нею складно погодитися. Будь-який засіб безпеки повинне бути контрольоване, інакше належної віддачі ви від нього ніколи не одержите. Наприклад, якщо хтось завзято намагається зламати ваш партнерський портал і йому не вдалося цього зробити в перші два рази завдяки IPS-системі, то втретє йому це вдасться, а без контролю за IPS-системою ви цього не довідаєтеся й запобігти наступним спробам у вас не вийде.

Juniper Networks

Що б не писали аналітики Gartner або інших видань, сказати щось гарне про їхні продукти складно. Система жахливо складна в налаштуванні. Консоль керування NSM дуже сильно обмежена. Відображення результатів виробляється таким чином, що складається враження, що розроблювачі постаралися зробити так, щоб на неї дивилися як можна рідше й сподівалися на те, що атаки дійсно відбуваються.

Sourcefire

Мабуть, краща система. Зручно все. Функціонал неймовірно широкий. До того ж система вже має убудовані можливості детального збору даних, про атакуючих і вузлах, які атакуються, а не тільки IP- і MAC-адреси, що сильно знижує час аналізу й розбору подій. До такої інформації відноситься й історія з'єднань, відкриті й потім закриті порти, типи переданої адреси, імена користувачів, якщо, наприклад, ішла передача по FTP або e-mail, і, звичайно ж, сам e-mail-адреса. У більших мережах може стати незамінним засобом захисту. Випускають свої рішення з 2001 р., але на український ринок вийшли недавно.

IBM security network intrusion prevention system

Рішення від IBM забезпечує блокування від мережеских атак, ботнетів, хробаків, комплексний захист застосунків, веб-застосунків і даних, при збереженні високої пропускної здатності й мінімальних затримок. Функції брандмауера дозволяють дозволити доступ тільки по певних портах і IP, без необхідності залучення додаткового пристрою. Інтегрований модуль DLP забезпечує моніторинг конфіденційної інформації минаючої через мережу. Система може заглянути усередину протоколу, визначити й блокувати паразитний мережеский трафік (начебто P2P і IM).

Як і інші розробки, IBM Security для детектування використовує запатентовану технологію аналізу протоколів превентивний захист, що забезпечує, у тому числі й від 0day погроз. Його основою є PAM (Protocol Analysis Module, модуль аналізу протоколів, goo.gl/3tKl) сигнатурний метод, що сполучить у собі традиційний, виявлення атак (Proventia OpenSignature) і поведінковий аналізатор, що включає в себе велика кількість евристичних механізмів (Shellcode Heuristics, Injection Logic Engine і т.д.), що дозволяє не покладатися на статичні сигнатури. Усього для аналізу трафіку використовується більше 3000 алгоритмів, 200 з них розпізнають DoS атаки. Також PAM розрізняє більше 200 протоколів рівня застосунків (атаки на HTTP, RPC, VoIP, RPC і т.д.) і формати даних (DOC, XLS, PDF, ANI, JPG) і може вгадувати місце, куди може

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

бути впроваджений шкідливий код. У цей час актуальною в пристроях версією прошивання є 4.6 (вийшла наприкінці березня 2019), що включає в себе ряд нових функцій. В основі лежить SNORT 2.9.3.1 з новими політиками за замовчуванням. Формат журналу SNORT змінений, щоб краще відслідковувати установки політик. При аналізі події можна переглянути правило його що викликало. Також тепер відображаються дані URL HTTP запитів (конфігурується в Secure Protection Settings > Advanced IPS > SNORT Configuration and Rules). Формат даних переданих зовнішньої SIEM системи змінений на IPFIX (IP Flow Information Export), а для переадресації журналу використовуються LEEF (Log Event Enhanced) формати, які сумісні з QRadar SIEM (відповідні налаштування перебувають в Manage System Settings > Appliance > Remote Data Flow Collection і LEEF Log Forwarding (syslog)). Для роботи цієї функції потрібно також відновлення модуля PAM. Правда деякі моделі не підтримують новий формат (GX6116, GX7412, GX 7412-05 і др.). У політику Security Interfaces (перебуває Manage System Settings > Network) з'явився параметр Report link який показує стан порту, що відображається в даних про стан агента. Його відключення дозволить наприклад не виводити помилки невикористовуваних портів. В Tuning Parameters (Secure Protection Settings > Advanced IPS) додано майже 20 нових параметрів які дозволяють більш тонко налаштувати роботу IPS, збір статистики й журналювання пакетів. Змінено деякі види звітів. Для GV200 і GV1000 офіційно підтримуються VMware ESXi 5.0 і 5.1. Підтримка v4.3 буде припинена у квітні 2021, v4.4 – серпень 2021, v4.5 – квітень 2022 року.

Розробкою PAM займається одна з найвідоміших дослідницьких груп безпеки у світі – IBM X-Force (xforce.iss.net), що разом із центром забезпечення безпеки GTOC (webapp.iss.net/gtoc) цілодобово відслідковує погрози, відповідають і за відновлення правил. Сигнатури можуть створювати самі користувачі, інтерфейс надає все необхідне, а процес докладно описаний у документі «OpenSignature User Guidelines» (goo.gl/yLwZT).

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Модульність РАМ дозволяє оперативно додавати підтримку протоколів і специфічних оброблювачів, тому його функціональність постійно розвивається. У цей час РАМ підтримує кілька технологій:

- IBM Virtual Patch – дозволяє блокувати відомі уразливості в додатках і ОС перше ніж їхні розроблювачі випустять офіційний патч, що відповідає правило може бути створено, як у рамках X-Force, так і адміністратором мережі. За замовчуванням функція активована (для всіх), але її можна відключити або налаштувати більш тонко по даним бази сканера безпеки ISS X-Press Updates (XPU);

- Client-side application protection – захист від атак спрямованих на користувальницькі додатки (MS Office, веб-браузери, Adobe PDF і т.п.);

- Advanced network protection – захист від мережових атак і атак на службу DNS;

- Data security – по суті модуль DLP, що робить моніторинг і ідентифікацію не зашифрованих персональних і конфіденційних даних, що забезпечує контроль спроб передачі або для відстеження переміщення інформації усередині мережі, що дозволяє оцінювати ризики й блокувати витік. За замовчуванням розпізнається 8 загальних типів даних (номера кредиток, телефони, поштова адреса й т.д.), іншу специфічну для організації інформацію адмін налаштовує самостійно за допомогою регулярних виражень.

- Web application security – дуже рідка функція в сучасних IPS, являє собою аналог файєрвола рівня застосунків (web application firewalls) захищаючий веб-застосунки й СУБД від специфічних атак (SQL injection, LDAP injection, XSS, JSON hijacking, PHP file-includers, CSRF і ін.);

- Application control – модуль контролю застосунків що дозволяє розпізнавати, обмежувати пропускну здатність або блокувати Skype, P2P, IM, Active елементи, спроби тунелювання трафіку й т.д.

- Network anomaly detection – збір даних для виявлення аномалій за допомогою SIEM рішень (Security Information and Event Management) начебто

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

IBM Security QRadar що представляє собою систему для збору, аналізу, архівування й зберігання мережевих журналів мережі й журналів подій безпеки .

Як видно можливості IPS від IBM набагато ширше ніж традиційні рішення які забезпечують в основному аналіз мережевого трафіку, але не вміють заглядати усередину пакета, забезпечуючи безпеку конфіденційної інформації, веб-трафіку й блокування застосунків. З огляду на, що веб-ресурси відкриті й доступні в режимі 24/7 атаки на них найпоширеніші, а специфіка вимагає особливого підходу. Тому модуль Web application security дуже затребувана функціональність.

Аналіз мережевих пакетів паралельно модулю PAM робить і IDS/IPS SNORT. Причому з огляду на, що використовується єдина черга, першим обробку й видачу рішення може зробити як PAM, так і SNORT, у журналах можуть з'являтися зайві записи. Наприклад, SNORT перевірів пакет і згенерував подію, що потрапило в журнал, а PAM його відразу відкинув. Нічого страшного в такому поводженні ні, просто потрібно бути в курсі.

Крім IBM Security QRadar підтримується інтеграція й з деякими іншими продуктами IBM. Наприклад із прошивання 4.1 з'явилася можливість інтеграції з IBM Security AppScan (раніше IBM Rational AppScan), що дозволяє тестувати застосунки використовуючи різні можливі вектори атаки, на предмет наявності уразливостей, дозволяючи виявити пробіли в безпеці й створювати на основі отриманих даних нові профілі захисту. Інтеграція з IBM Security Network IPS дозволяє клієнтам, що використовують AppScan для створення політик захисту веб-застосунків показати, що повинне бути захищене.

Місце в мережі

Пристрої IBM Network IPS працюють на другому (канальному) рівні моделі OSI, тому їхнє впровадження відбувається прозоро й не вимагає внесення змін у конфігурацію мережі. Просто підключаємо в «розрив» і IPS готова до роботи. І головне таку систему безпеки не має IP, сутужніше виявити й розпізнати. Реалізовано три режими роботи: моніторинг – пасивне виявлення

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

(режим IDS), активний захист (Prevention, режим IPS) і імітація (Simulation, режим навчання IPS, без блокування). При виявленні атаки передбачений ряд дій: блокування або приміщення в карантин вилученого вузла (за замовчуванням 1 година), відправлення попередження, ігнорування, запис трафіку атаки (файл сполучимо з tcpdump) і дія, що налаштовується користувачем. Один пристрій підтримує декількох зон безпеки, включаючи VLAN і може захищати декілька доменів. Передбачений дуже широкий діапазон по установках політик, роздільні політики вказуються для кожного пристрою, порту, тега VLAN, IP-адреси або IP-діапазону.

При наявності в мережі двох і більше пристроїв IPS активація режиму High Availability гарантує, що у випадку виходу з ладу одного з них трафік піде через інше, установлені з'єднання при цьому не перервуться. Пристрою при цьому можуть працювати в одному із двох режимів роботи – active/active (кластер) або active/passive (обробляє тільки один, другий перебуває в гарячому резерві). Висока готовність забезпечується, завдяки можливості перемикання «на лету» між географічно розподіленими пристроями IPS. У цьому випадку другий пристрій підхоплює всі з'єднання, і що немаловажно, що поточні сесії при цьому не перериваються (для цього трафік із другого пристрою дзеркалюється). Причому можливі тонке налаштування параметрів інспектування пакетів у режимі HA. Наприклад за замовчуванням SNORT і PAM аналізують всі з'єднання прийшли із зеркальованого порту (без відповідної реакції), це дозволяє збільшити охопат і дозволяють із більшою ймовірністю виявити атаку, хоча й нерідко приводить до створення дублюючих повідомлень в SiteProtector (у локальній консолі події із дзеркального порту не відображаються). При необхідності таку функцію легко відключити.

Інтерфейс керування

Керування всіма версіями IBM IPS виробляється за допомогою веб-консолі (Local Management Interface, LMI), побудованої з використання Java технологій (за замовчуванням працює на стандартному https/443 порту). З його

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

допомогою можна: контролювати стан пристрою, налаштувати режими роботи IPS, параметри брандмауера й політики безпеки, переглядати оповіщення й т.д. Офіційно підтримуються веб-браузери IE або FF, також потрібно встановлений JRE 1.6/1.7 (x64 JRE не підтримуються). Для запуску LMI потрібно близько 1 Гб ОЗП, тому комп'ютер адміністратора повинен мати відповідні ресурси й працювати під керуванням Win XP-7/10. Варто відзначити, що при таких потужних можливостях IPS, керування залишається дуже простим і інтуїтивним.

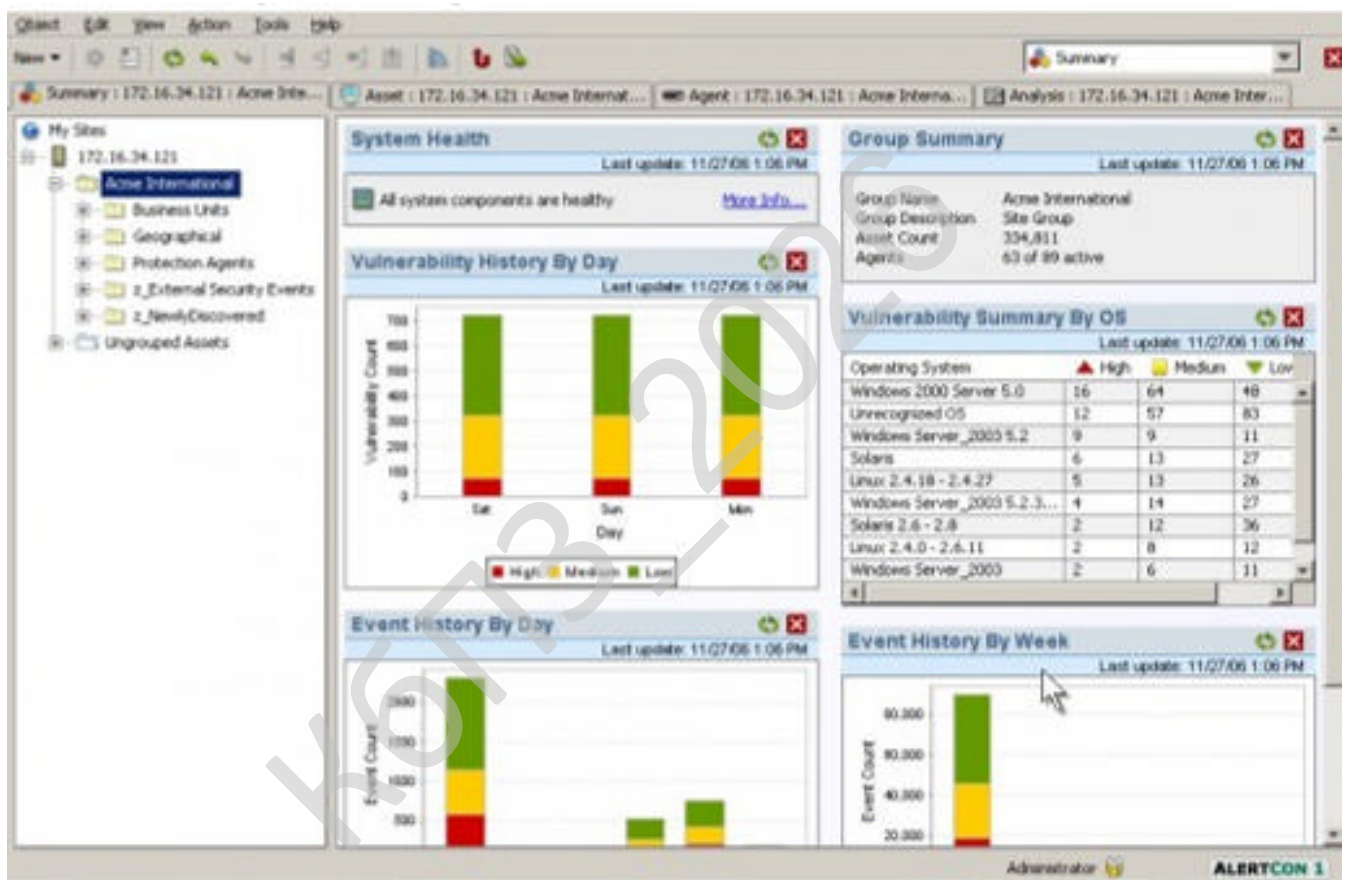


Рисунок 2.1 – Інтерфейс користувача

Зовнішній вигляд веб-консолі можна назвати класичним. Після входу тебе зустрічає Appliance Dashboard вивідну основну інформацію й графіки з інформації про стан пристрою, мережі, кількості інцидентів, заповнювання кореневого розділу жорсткого диска/ОЗП й т.п. На самому видному місці

показується ТОП 10 атак, що атакують і атакуємих. Панелі що налаштовуються, це дозволяє вивести затребувані установки й ключові показники на видне місце й відслідковувати їхній стан коли буде потрібно. У багатьох питаннях, у тому числі й при редагуванні політик допомагають майстра й щоб додати або змінити параметр необхідно лише заповнити запропоновані поля.

Призначення інших вкладок зрозуміло з назви: Monitor (стан і статистика), Secure (установки безпеки), Manage (установки системи) і Review (аналіз і діагностика). При наведенні на кожній із зазначених пунктів з'являється спливаюче меню, також візуально розділене на підпункти, що дозволяє швидко одержати доступ до потрібної функції або налаштування. Кожний підпункт також містить панель Dashboard у якому показані всі основні зв'язані події, але більш що повно розкриває проблеми. Наприклад у розділі Monitor знаходимо підпункти Network (стан мережі й пропускна здатність), Security (інформація з інцидентів, ТОП атак, що атакують і атакуємих ресурсів і ін.) і System (стан, доступні й зайняті ресурси). Вікно перегляду подій безпеки Security Events показує не тільки події (протокол, важливість, опис і т.п.), але й дозволяє відразу встановити дія – ігнорувати або блокувати. При додаванні користувальницької події з'являється вікно майстра в якому задається назва, домен, дія й т.д. Але є тут одна тонкість – для визначених користувачем подій (User Overridden), до них у наслідку не можна застосувати X-Force Virtual Patch. Політики Web application security дозволяють захистити декілька доменів. Щоб активувати відповідну реакцію досить просто перейти в підменю й установити відповідні прапорці (включити, відобразити або заблокувати). Аналогічно просто настроюються правила firewall, натискаємо Add і в заповнюємо/вибираємо значення у вікні Add Firewall Rules.

Усе знаходиш там де шукаєш. У підсумку не дивлячись на велику кількість функцій, розібратися з конфігурацією представляючи, що потрібно робити дуже легко. Щоб освоїтися досить витратити годинка другий часу. Про простоту налаштувань говорить те що курси навчання займають усього один день.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Для керування передбачено кілька типів облікових записів (локально дозволено входити тільки убудованим root і admin), для автентифікації може використовуватися LDAP, у тому числі й Active Directory.

Крім LMI можливе керування рядом налаштувань за допомогою консолі (підключення по SSH) і кнопок лицьової панелі. Для одержання статусу підтримується інтеграція по SNMP із продуктами третіх фірм.

При наявності декількох пристроїв звичайно здобувається IBM Proventia Management SiteProtector, що забезпечує централізоване керування всією лінійкою продуктів ISS і третіх фірм, збір і відображення подій у реальному режимі часу, їхня фільтрація й аналіз, створення звітів.

Поставка

Лінійка пристроїв IBM Security Network IPS складається з апаратних рішень із продуктивністю від 10 Мбіт/с до 15 Гбіт/сек і підтримуючі 10 Гбіт інтерфейси, програмний модуль який забезпечує на платформі Crossbeam швидкість до 40 Гбіт/с. Апаратна платформа гарантовано забезпечує високу пропускну здатність (до 160 000 з'єднань), низьку затримку (< 150 мікросек) і великий час безвідмовної роботи. Молодші моделі GX3002 і GX4002 призначені для вилучених філій (GX3002 поставляється в десктопному формфакторі), старші – для захисту більших мереж. Усередині пристроїв всі підсистеми дубльовані: RAID, два блоки живлення, два вентилятори охолодження. Убудовані й зовнішні bypass модулі (наприклад, IBM Security Network Active Bypass) дозволяють передавати дані в обхід пристрої у випадку системної помилки або збоїти в електроживленні. Також представлені два варіанти IPS (GV200 і GV1000) у вигляді віртуальної системи (Virtual Appliance) призначений для захисту як віртуальних, так і фізичних мереж з тим уже рівнем безпеки, що й апаратні рішення. Для установки використовуються продукти VMware. Під час установки Virtual Appliance буде створено три мережевих інтерфейси: керування (Management port), що захищається мережі (Two Layer 2 sensing) і IPS (для скидання TCP з'єднань).

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Можливості IBM Security Network IPS вражають, цей продукт не зрячи відзначали під час численних тестів проведених незалежними експертами по безпеці й комп'ютерних виданнях. Також можна відзначити дійсно просте впровадження, якісну документацію (на англійському), наявність декількох навчальних роликів, що дозволяють швидше розібратися з налаштуваннями пристроїв.

Не варто впроваджувати цілий ряд нових продуктів, що вирішують тільки одну проблему. Статичні засоби безпеки не можуть захистити динамічне середовище. Необхідно берегти час ваших співробітників і їхнього зусилля. Дозвольте їм працювати більш якісно, а не більш напружено. Скорочуйте витрати на підтримку гетерогенного середовища. Знижуйте час, затрачуваний на аналіз даних з безлічі консолей і звітів. Витрачайте гроші з розумом, поки системи безпеки не стали обходитися вам дорожче ризиків, від яких ви захищаєтеся.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – високорівнева мова програмування, яку називають другою за популярністю в світі. Її використовують для розробки вебзастосунків, програмного забезпечення, машинного навчання. Python застосовують для вирішення робочих завдань у компаніях Google, Instagram, Facebook, IBM, NASA, Dropbox, Netflix та інших. Розробники цінують цю мову програмування за простоту у вивченні, ефективність та мультиплатформність.

Python – скриптова мова програмування з досить простим синтаксисом. Для розуміння достатньо порівняти принципи написання найпростішої програми, яка виводить на екран текстове повідомлення. Саме тому мова програмування Python більш доступна для новачків, а професіонали встигли адаптувати її для вирішення великої кількості завдань. Це мультиплатформне рішення, тому знання Python дає можливість працювати у різних сферах: від розробки

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

мобільних застосунків до ігрової індустрії та штучного інтелекту. У мови програмування динамічна типізація: є можливість передавати до функцій будь-який тип даних без попереднього вказання. Інтерпретованість дозволяє знаходити помилки у коді ще до повної збірки у робочий застосунок. При цьому Python дуже чітко дає зрозуміти, де та через що виникла помилка.

Це мова об'єктноорієнтованого програмування (ООП). Програмне забезпечення на Python оформлене у вигляді моделей, які можуть бути зібраними у пакети. Тип та структуру кожного об'єкта можна запитати під час виконання програми. Для кожного з об'єктів можна отримати всю інформацію щодо його внутрішньої структури. Окрім того:

- у мови логічний синтаксис, завдяки чому вихідний код легко читати та розуміти;
- гнучкість та масштабованість Python дозволяє адаптувати високорівневу логіку та розширяти складні застосунки, як тільки виникне така необхідність;
- розробка на Python у більшості випадків проходить швидше, ніж на інших мовах програмування;
- Python – інтерпретована мова програмування. Це значить, що код можна написати у будь-якому текстовому файлі на будь-якій платформі, і потім успішно запустити;
- у Python – колосальна спільнота однодумців. Тож будь-які складнощі конкретних розробників вирішуються колективно.

Проте є декілька особливостей, які можна віднести до недоліків. Це повільність (ця мова програмування хоч і універсальна, проте повільніша за інші), велика кількість ресурсів, необхідних для роботи та «прив'язаність» до системних бібліотек.

Мова програмування Python використовується у наступних сферах:

1. Розробка програмних застосунків будь-якого напрямку.
2. Розробка серверної частини мобільних застосунків (найпопулярніший напрямок).

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

3. Ігри. Багато сучасних ігор для комп'ютерів (наприклад, World of Tanks) частково чи повністю написані на Python.

4. Вбудовані системи для різних пристроїв. Дуже часто Python використовують для написання внутрішніх платформ управління банкоматами.

5. Скрипти та плагіни до уже реалізованих програм для автоматизації процесів чи створення інших рішень.

6. Тестування (автоматизація цього процесу).

7. Машинне навчання. – основна мова для написання алгоритмів і аналітичних застосунків у сфері Machine Learning.

Бібліотеки Python

Різні бібліотеки Python використовують для виконання конкретних завдань. Наприклад, Matplotlib підходить для відображення даних у двовимірній та тривимірній графіці. Pandas підходить для зручної роботи з даними. NumPy дозволяє створювати масиви та керувати ними. Requests використовується для веброзробки. OpenCV-Python відкриває можливості для обробки зображень з метою оптимізації систем «машинного зору».

Найвідоміші фреймворки для мови програмування Python

Фреймворки Python допомагають створити зручне та функціональне середовище для розробки. У них міститься набір інструментів, модулів та бібліотек, корисних для виконання конкретних завдань. Це значно полегшує роботу: наприклад, дає змогу не витратити час на розписування дій, які повторюються, а використати релевантний інструмент. Тож є можливість позбутися рутинних процесів та сконцентруватися на логіці проєкту.

Серед найпопулярніших фреймворків для Python:

– Django – найстаріший та найвідоміший. Створений для реалізації великих інтерактивних проєктів;

– Pyramid – зручний у налаштуваннях, і дає можливість реалізувати складні нестандартні ідеї;

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) 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) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

IPS (Intrusion Prevention System) українською – система запобігання вторгнень. Причому акцент саме на Prevention (тобто запобігання)! Одне з головних кредо компанії Check Point це: “We Prevent, not detect!”. Особисто я згодний з такою позицією. Яка користь від детекта, якщо вас атакував наприклад шифрувальник? Зашифрований комп'ютер і так вас повідомить, що була атака. У поточних реаліях потрібно подбати саме про Prevent. І IPS тут може дуже добре допомогти.

Однак, останнім часом спостерігається якась зневага цим класом захисту, мол “IPS більше не актуальний і використовувати його безглуздо”. На мій погляд ця думка є непрощеною помилкою. Властиво в цьому уроці я постараюся описати основні існуючі омани на рахунок IPS. Потім у рамках лабораторної роботи покажу яким образом IPS допоможе підсилити захист вашої мережі. Ну й звичайно ж постараюся розповісти, як одержати максимум від цього корисного інструмента, на які налаштування звернути увагу й про що потрібно пам'ятати включаючи IPS.

Урок вийшов досить довгим тому я розбив його на дві частини. Перша частина буде чисто теоретична, а друга вже повністю присвячена практиці у вигляді лабораторної роботи. Сподіваюся, що буде цікаво.

Хотілося б почати з деяких особливостей. Насправді IPS є часткою случаємо IDS (Intrusion Detection System – Система виявлення вторгнень або СОВ, як її коротко називають в Україні). Ідея про створення IDS з'явилася після виходу розділі “Computer Security Threat Monitoring and Surveillance” Джеймса Андерсона, аж в 1980 році! Досить занятна стаття й саме головне актуальна

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

донині. Через 6 років, в 1986 році Дороті Деннінг і Пітер Нейман опублікували першу теоретичну модель IDS, що напевно дотепер є основою для сучасних систем. Далі було досить багато різних розробок, але всі вони в основному зводилися до використання трьох методів виявлення вторгнень:

- Сигнатурний аналіз.
- Евристичний аналіз.
- Виявлення аномалій.

Мабуть одним із ключових моментів у розвитку IDS стала поява бібліотеки librsar в 1998 році. Розробили її хлопці з Національної лабораторії ім. Лоуренса в Беркли. У цьому ж році був розроблений сніфер пакетів APE пакет, що використовує, librsar. Через місяць APE був перейменований у всім відомий Snort. Автором Snort-а є Мартін Реш.

Пізніше була заснована компанія Sourcefire (в 2001 році) і проект Snort продовжив свій стрімкий розвиток уже в рамках компанії Sourcefire і став фактично стандартом серед IDS рішень. Snort має відкритий вихідний код, ніж і користується більшість сучасних виробників ІБ рішень (особливо вітчизняні компанії).

В 2003 р. компанія Gartner констатувала неефективність IDS і необхідність переходу на IPS системи (тобто перемінити детект на превент). Після цього розроблювачі IDS стали оснащувати свої рішення режимом IPS. Snort природно може працювати як в IDS, так і в IPS режимі (тобто на запобігання).

Безумовно варто також відзначити бурхливий ріст ще одного проекту з відкритим вихідним кодом – Suricata. Заснували цей проект вихідці з Snort-а. Перший бета реліз був в 2009 році. За розробку відповідає компанія Open Information Security Foundation (OISF). На даний момент Suricata є дуже популярним рішенням (хоч і уступає поки ще Snort-У по популярності). Насправді їх досить часто використовують спільно.

В 2019 році Sourcefire була поглинена компанією Cisco. При цьому Snort продовжує залишатися проектом з відкритим вихідним кодом, а його комерційна

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

версія продається під брендом Cisco FirePower. З вашого дозволу ми не будемо говорити про відмінності вільної й пропрієтарної версії. Цікавий момент. Ще в 2005 році Check Point намагалися купити SourceFire за 225 млн \$, однак уряд США не схвалили цю угоду. І як я сказав раніше, Cisco купили SourceFire в 2019 році аж за 2,7 млрд \$. Непогане подорожчання за 8 років) більш ніж в 12 разів.

Природно я перелічив тільки малу частину рішень. Паралельно розвивалася величезна кількість комерційних пропрієтарних рішень (Check Point один з них). Найчастіше IPS входив до складу UTM або NGFW рішення, рідше в якості окремо варті “залозки” (Cisco IPS – яскравий приклад).

IPS у корпоративних мережах

Тепер, якщо торкнутися історії поширення IPS рішень у корпоративних мережах, то виходить приблизно наступна картина:

На початку 2000-х компанії досить скептично ставилися до цього нового класу рішень захисту. Більшість уважали IPS якоюсь екзотичною штукою, що не особливо те й потрібна.

Уже після 2005-го більшість усвідомили користь і необхідність IPS. Почався бум впроваджень по усьому світі.

До 2010 року IPS став фактично необхідним стандартним засобом захисту корпоративної мережі.

Ближче до 2015-му ринок IPS відносно охолонув. IPS по стандарті був практично у всіх UTM/NGFW рішеннях. Усе перемкнулися на SIEM, захист від таргетованих атак, пісочниці, хонейпоти й т.д. При цьому зовсім забувши про важливість IPS. Але це ми вже обговоримо ледве далі.

Тепер, коли ми небагато освіжили знання про історію появи IPS, хочеться обговорити ще один момент. А саме класи IPS. У грубому наближенні всі IPS-Рішення можна розділити на два класи:

– NIPS працює на рівні мережі, тобто сканує минаючий/транзитний трафік.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

– HIPS працює на рівні комп'ютера користувача, тобто з тим трафіком, що призначений безпосередньо цьому комп'ютеру, ну або генерує сам хост.

Ми будемо обговорювати саме перший варіант.

Два варіанти використання NIPS

Давайте розглянемо архітектурне застосування IPS. Тут теж все доволі просто, є два варіанти використання мережевого IPS:

1. Inline mode. По-українському ми це називаємо “у розрив”. Тобто реальний мережевий трафік проходить через IPS. IPS у цьому випадку працює як звичайний бридж (тобто на другому рівні моделі OSI). Даний режим самий оптимальний з погляду захисту. У випадку виявлення атаки, IPS може відразу заблокувати сесію й комп'ютер атакуючі. Є звичайно й негативні моменти у вигляді false positive спрацьовувань, ну тобто помилкових спрацьовувань, коли IPS блокує нормальний трафік. Але це окрема тема, ми декілька пізніше її обговоримо.

2. Promiscuous mode. Знову ж, по-українському – режим моніторингу. Як правило в такому режимі IPS “вішається” на SPAN-порт, що “зеркалює” на пристрій копію! трафіку. У цьому варіанті, IPS автоматично перетворюється в IDS, тому що він працює вже не з реальним трафіком і в нього немає можливості оперативно блокувати атаки. Тобто це поганий варіант, коли потрібна максимальна захищеність.

Ми не будемо докладно розглядати особливості застосування того або іншого режиму. Це досить велика тема, що виходить за рамки нашого уроку. Давайте краще зосередимося на inline режимі, тому що саме він у більшості випадків використовується в Check Point-E.

IPS в UTM пристроях. Packet Flow

З погляду обробки трафіку, пакети спочатку перевіряються firewall-ом і якщо вони дозволені відповідними аксес аркушами, те тільки тоді включається IPS і починає перевіряти трафік, який проходить. Властиво цей алгоритм обробки

трафіку розкриває концептуальне розходження між Міжмережевим екраном і Системою запобігання вторгнень.

Міжмережевий екран прагне запобігти проходженню того або іншого трафіку. IPS же працює із уже минулим трафіком.

Це логічно не тільки з погляду безпеки, але й з погляду продуктивності. Навіщо досліджувати трафік, що може бути швидко відкинутий файрволом з мінімальними ресурсними витратами. Це до питання, чи варто ставити IPS перед firewall-ом. Однозначно немає! Тільки представте скільки “лівого” трафіку буде на нього сипатися від різних ботів, які сканують в Інтернеті всі підряд.

Що ж, на цьому ми закінчуємо наш тривалий ліричний відступ. Давайте перейдемо до типових оман на рахунок IPS. Я постараюся їх розвінчати на прикладі Check Point-a.

Типові омани на рахунок IPS

1. IPS захищає тільки від атак мережевого рівня

Мабуть це найпоширеніший міф. Історично звичайно IPS у першу чергу захищав від мережеских атак, таких як сканування портів, брутфорс, деякі види ddos-а ну й звичайно боротьба з аномаліями. Однак! Багато хто дотепер не знають, що IPS може перевіряти й завантажувані файли! І якщо файл містить exploit, те IPS заблокує його завантаження швидше ніж Anti-Virus, т.к. IPS працює в потоці трафіку, а Anti-Virus змушений чекати поки в буфер скачає весь файл. Тобто IPS перевіряє такі файли як pdf, doc, xls і багато чого іншого. Я обов'язково покажу це в лабораторній роботі. Тому, включений IPS істотно підвищить захист ваших користувачів, які качають із інтернету різні файли. Не зневажайте цим додатковим рівнем захисту!

2. IPS уже не актуальний і ні від чого не захищає

Ще один популярний міф. Безумовно, останнім часом професійні хакери намагаються не використовувати класичні інструменти атаки, такі як сканування портів, брутфорс і т.д. І всі тому, що такі атаки відразу помітні й генерують величезне кількість алертів на класичних засобах захисту. Однак! Це має місце

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

бути тільки при дуже складних і таргетованих атаках, коли за справу береться справжній професіонал. 99% всіх успішних атак – автоматизовані боти, які сканують мережа на предмет відомих уразливостей, які потім і експлуатують. IPS все це бачить! Більше того, знову ж згадавши wappasгу, після виявлення цієї уразливості, Check Point випустив IPS-сигнатуру буквально через пару днів. Microsoft же випустив заплатку куди пізніше (через пару тижнів). Включений IPS з актуальними сигнатурами відмінно відбиває подібні автоматизовані атаки, які дотепер переважають (та й навряд чи щось зміниться в найближчому майбутньому).

3. IPS не треба часто обновляти

Властиво в попередньому пункті я констатував, що включений IPS саме з актуальними сигнатурами забезпечує захист від автоматизованих атак. Чомусь багато хто вважають, що регулярно обновляти потрібно тільки антивірусні бази, при цьому геть-чисто забуваючи про IPS. Але ж сигнатури для IPS з'являються або обновляються буквально щодня. Для приклада можна скористатися ресурсом Check Point. Як бачите тільки за останні пару днів вийшло кілька нових сигнатур. Або були оновлені раніше створені.

IPS з актуальними базами це дуже важливо. Як я вже сказав раніше, IPS сигнатури виходять швидше ніж патчи від вендорів. Якщо у вас старі сигнатури, то ваш IPS просто в порожню молотить трафік і безцільно витрачає системні ресурси. І не вірте вітчизняним виробникам COB, які говорять, що відновлення раз на місяць це нормально (як правило саме такий період вони ставлять і швидше за все це пов'язане з тим, що вони використовують бази snort, які для безкоштовної версії обновляються із затримкою в 30 днів).

4. IPS істотно знижує продуктивність пристрою

Що я можу сказати про цей міф. І та й немає. Безумовно, включення IPS збільшує навантаження й на процесор і на оперативну пам'ять. Але всі не так

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

драматично, як прийнято вважати. Колосальне підвищення навантаження при включенні IPS як правило проявляється у двох випадках:

– Ви (або Вам) неправильно підібрали рішення. Будь те апаратний аплайнс або віртуальне рішення. Це до питання, що все необхідно тестувати перед покупкою. Знову ж, отут багато чого залежить від того, хто підбирає рішення. Приділяйте цьому особливу увагу. Уважно користуйтеся даташитами на пристрої. Там обов'язково вказується пропускна здатність пристрою при включеному IPS. Ну й знову ж, в Check Point-а є інструмент сайзинга, що дає досить реалістичні рекомендації із приводу підходящої моделі. Обов'язково користуйтеся сайзингом! Якщо у вас немає доступу до цього інструмента, просите своїх партнерів.

– Друга й найпоширеніша причина підвищеного завантаження через IPS – включене занадто багато сигнатур. Це одна із крайностей використання IPS, коли включаються абсолютно всі доступні сигнатури, через те, що споконвічно не зрозуміло, які саме потрібні. Це дуже груба помилка приводить до просто дикого завантаження пристрою й великій кількості помилкових спрацьовувань. Напорювшись на це багато адміністраторів вирішують відключити IPS, тому що з ним прикордонний шлюз просто “загинається”. Як бути? Ну для початку потрібно визначити, що саме ви збираєтеся захищати. Логічно припустити, що якщо у вас в DMZ перебуває поштовий сервер на Linux-E, те напевно не варто включати для цього сегмента сигнатури пов'язані з уразливостями Microsoft, MacOS, Android, Wordpress і т.д. Думаю загальний зміст зрозумілий. Молотити трафік всіма сигнатурами не перебираючи – дуже дорого. Щоб IPS ефективно витрачав ресурси, необхідно включити тільки потрібні сигнатури й виключити непотрібні. Звучить просто. Але реалізація представляється досить складної. Звідси росте наступний міф.

5. IPS важко налаштовувати

Почасти ця справедлива думка, які мало місце бути. Насправді багато які IPS рішення дотепер досить складні в освоєнні. Але це не про Check Point.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Давайте один по одному. Як звичайно виглядає сигнатура? Як правило це щось начебто:

IGSS SCADA ListAll Function Buffer Overflow

WebGate Multiple Products WESPMonitor Stack Buffer Overflow

І спробуй розберися, що це за сигнатура, для чого вона, чи сильно навантажить шлюз її включення, наскільки вона критична? На щастя в Check Point-E є докладний опис кожної сигнатури. Опис видно прямо в SmartConsole і вам не прийдеться гуглити кожну назву. Більше того, для зручності, Check Point привласнив кожній сигнатурі кілька тегів:

- Severity (тобто рівень небезпеки, що може закрити дана сигнатура).
- Confidence level (рівень вірогідності сигнатури, тобто яка ймовірність адекватності спрацьовування цієї сигнатури. Це дуже важлива характеристика особливо для поведінкового аналізу).
- Performance Impact (цей тег показує на скільки сильно включених сигнатур буде вантажити пристрій).
- Vendor – ви можете відфільтрувати сигнатури по Вендору (наприклад уразливості зв'язані з усіма продуктами компанії Adobe або Microsoft).
- Product – можливо відфільтрувати по продукті (наприклад Microsoft Office або WordPress).

Теги Вендор і Продукт іноді перетинаються, тому що досить часто в компанії є всього лише один продукт.

За допомогою цих тегів, можна досить швидко сформувати список сигнатур, які ви хочете включити. І для цього вам не треба мати сім п'ядей у чолі. Ну й природно перед цим вам потрібно зробити невеликого аудита мережі й зрозуміти, який софт використовують ваші співробітники.

6. Робота з IPS це нудотна й рутинна робота

Інформаційна безпека це не результат, а безперервний процес. Не можна один раз налаштувати систему й забути про неї. Потрібна безперервна, систематична доробка. Теж саме стосується IPS. І отут звичайно виникають

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

труднощі. Практично кожна більше менш адекватна IPS система генерує величезну кількість логів.

А тепер представте, що цих логів набралось кілька тисяч за тиждень. Яким образом вам аналізувати їх? Як зрозуміти які події відбуваються найчастіше? Які сигнатури можливо треба виключити або які хости можливо варто заблокувати на рівні firewall-a?

Отут видно й Severity і Performance Impact і Confidence Level. Однак це не вирішує проблему аналізу такої великої кількості подій. Як правило саме тут згадують про SIEM системи, які покликані агрегувати, корелювати й виконувати первинний аналіз подій. На щастя в Check Point є убудована SIEM система – Smart Event. Цей блейд дозволяє бачити логи IPS уже в обробленому й агрегованому виді.

Як ви розумієте, із цим уже набагато простіше працювати. Головна цінність SIEM у тому, що ця система дозволяє “побачити” вашу захищеність у кількісному вираженні. У будь-якій роботі потрібно бачити результат і тут набагато простіше орієнтуватися на цифри. У нашій прикладі ми бачимо, що є присутнім досить велика кількість логів з Severity рівня Critical. Саме з ними й потрібно починати роботу. Більше того, ми бачимо, що найбільша кількість подій пов'язане із сигнатурою GNU Bash Remote Code Execution. Розбір варто почати саме із цих подій. Провалившись далі ми можемо визначити:

- Які хости атакують?
- Який хост атакують?
- З якої країни йде атака?

Може вийти, що сигнатура спрацьовує на трафік, що генерує наш внутрішній вузол. Це вже привід для розгляду. Або він дійсно заражений, або це помилкове спрацьовування. Далі ми або лікуємо хост, або виключаємо цю сигнатуру конкретно для цього хоста.

Якщо ж атака йде із зовнішньої мережі, то може бути всі ці логи створює всього один атакуючий вузол (швидше за все це якийсь бот), причому з якого-небудь Сінгапуру. У цьому випадку ми можемо додати цей хост (або взагалі всю

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

мережу Сінгапуру) у блок-аркуш, щоб він блокувався на рівні firewall-а й не доходив до обробки трафіку IPS.

Також ми можемо помітити, що сигнатура відловлює атаки для linux-дистрибутивів, при цьому можливо атакується windows-хост. У цьому випадку буде також логічно відключити linux-сигнатури конкретно для цього хоста.

Як бачите, процес нагадує якесь розслідування. Після подібної постійної оптимізації істотно знизиться навантаження на шлюз, тому що:

- Частина атакуючих хостів буде блокуватися ще на рівні firewall-а.
- Будуть відключені непотрібні сигнатури, які витрачають ресурси.

При цьому SmartEvent дозволяє відразу побачити результат у вигляді зменшеної кількості логів. Це саме та, кількісна характеристика, що показує ефективність нашої проробленої роботи. В ідеалі ми повинні позбутися від так званого “шуму” і бачити в логах тільки дійсно важливі IPS події (тобто справжні атаки).

3.2 Розробка структурної схеми

Системи виявлення й запобігання вторгнень, які пропонуються провідними вендорами, адаптуються до погроз, що змінюються, а IDPS нового покоління еволюціонували у відповідь на появу нових і цілеспрямованих погроз, які IDPS першого покоління виявляти не можуть.

Програмні або програмно-апаратні системи виявлення й/або запобігання вторгнень (Intrusion Detection System / Intrusion Prevention System, IDS/IPS або IDPS) – один з найважливіших елементів систем інформаційної безпеки мереж будь-якого сучасного підприємства. Ріст числа проблем, пов'язаних з ІБ, привів до того, що такі системи дуже швидко стали ключовим компонентом будь-якої стратегії мережевого захисту.

Завдання IDPS – припиняти спроби несанкціонованого мережевого доступу, атаки зовнішніх хакерів або інсайдерів, а також запобігати можливий

збиток, що шкідливі програми можуть нанести корпоративним ІТ-системам. Для цього застосовуються різні методи виявлення аномальної мережевої активності, спроб зміни повноважень доступу до корпоративних ресурсів або конфіденційного даним, використання уразливостей у програмному забезпеченні та ін. Системи IDPS повинні мати наступні основні функції:

- виявлення вторгнень або мережевих атак;
- запобігання вторгнень і атак;
- автоматичне блокування мережевих атак;
- недопущення подальшого розвитку мережевих атак;
- визначення джерел атак;
- прогнозування можливих атак;
- виявлення уразливостей і контроль безпеки;
- збір даних для запобігання атак і усунення їхніх наслідків;
- документування існуючих погроз.

Види IDPS

Такі системи містять у собі модулі збору, зберігання й аналізу подій, модулі моніторингу, керування, налаштування й адміністрування, використовують бази даних про уразливості і репутаційні сервіси, але можуть розрізнятися типом сенсорів і механізмами аналізу подій. Функціонують вони на рівні мережі або на рівні окремого сервера.

Фізично IDPS можуть являти собою або пристрою, у тому числі багатофункціональні міжмережеві екрани нового покоління (NGFW) і комплексні системи безпеки (UTM), або віртуальні машини або програмний продукт. Вони можуть використовуватися як на площадці замовника, так і у віртуальних середовищах провайдерів або публічних хмар.

Здійснюючи глибокий аналіз протоколів і кореляцію подій, IDPS працюють у режимі реального часу, впливаючи на пропускну здатність і затримки в мережі. Вони часто застосовуються для перевірки трафіку, що вже пройшов через пристрої захисту периметра мережі – міжмережеві екрани й

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

плюзи безпеки. Виділені IDPS установлюються в різних місцях: за міжмережевим екраном як додатковий рівень захисту, за контролером доставки застосунків (балансувальником навантаження) для перевірки трафіку, а також у різних точках корпоративної мережі для захисту внутрішніх активів.

Методи виявлення

Виявити вторгнення можна декількома способами: наприклад, по сигнатурах і аномаліям протоколу, із застосуванням різних методів аналітики, за допомогою моніторингу, евристичних методів, виявлення аномального поведіння, за допомогою вдосконалених методів захисту від погроз (Advanced Threat Defense, ATD) і аналізу погроз (Threat Intelligence, TI). Ці просунуті методи розширюють можливості IDPS і одночасно допомагають скоротити число попереджень і помилкових спрацьовувань.

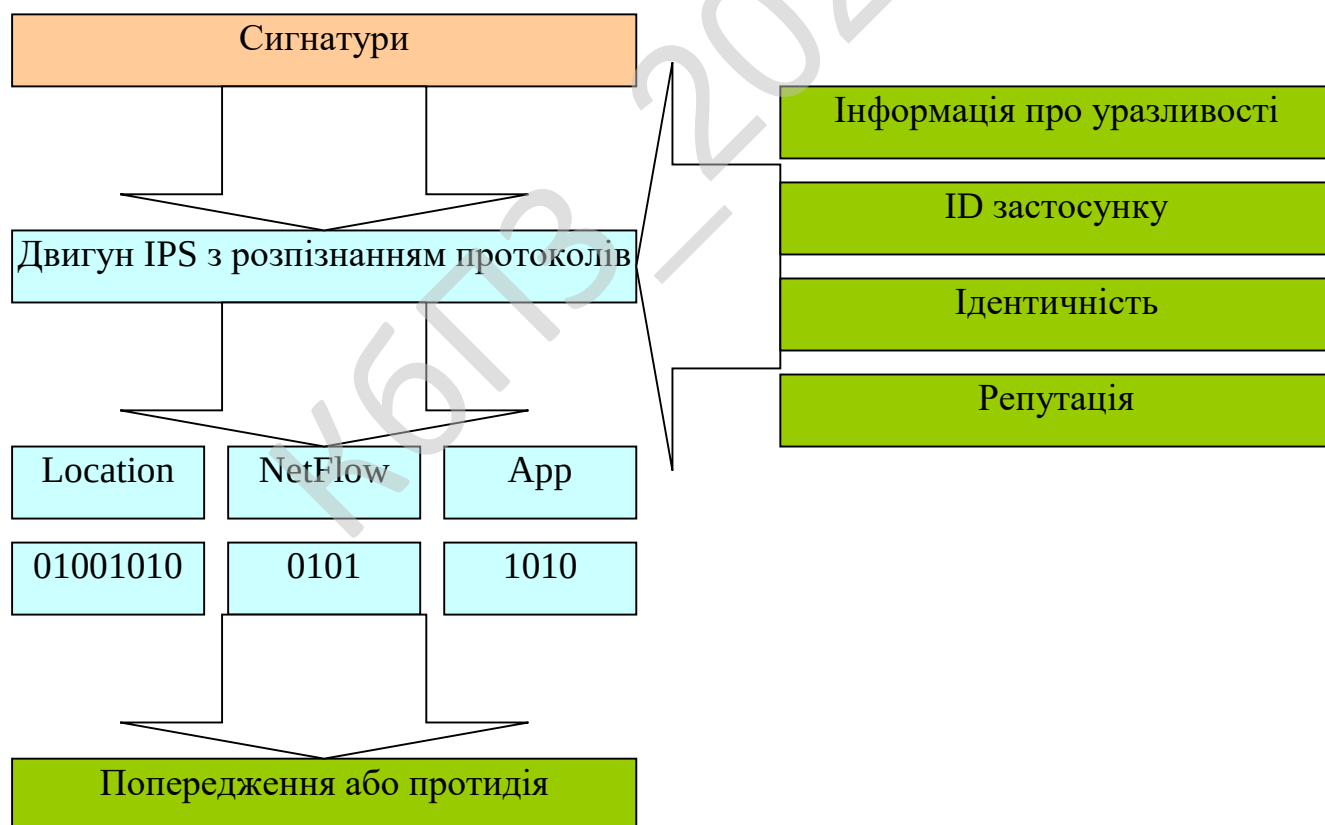


Рисунок 3.1 – Структурна схема системи

Аналіз на відповідність сигнатурам одержав широке поширення, але не завжди ефективний: для його застосування може знадобитися попередня обробка трафіку, а якщо для атаки використовується уразливість «нульового дня», те сигнатури виявляються марними.

Один з більше прогресивних методів – автоматичне навчання на основі аналізу поведінки користувача. Спочатку, ґрунтуючись на накопичуваних даних, система будує модель нормального поведінки, а потім відслідковує відхилення від неї. Оскільки модель поведінки користувача нестатичне, необхідно постійне навчання.

Системи нового покоління (NGIDPS) функціонують у режимі реального часу, здійснюють моніторинг застосунків, можуть задіяти сторонні бази сигнатур і уразливостей. Інновації в даній області фокусуються також на просунутій аналітиці й підтримці публічних хмар.

NGIDPS працюють на рівні застосунків, використовують інформацію з різних джерел для прийняття рішень про блокування трафіку, аналізують виконувані й інші типи файлів, а гнучкий движок дозволяє задіяти нові джерела інформації й методи виявлення вторгнень. Наприклад, такі додаткові методи, як розширена аналітика поведінки (User and Entity Behavior Analytics, UEBA), служать для виявлення погроз, які не були виявлені іншими системами захисту. Подібні механізми виявлення аномального поведінки використовують у своїх продуктах Hillstone Networks і McAfee. До числа нових можливостей відноситься також аналітика подій безпеки або даних, що збираються в мережі даних телеметрії (наприклад, IDPS Trend Micro).

Ще одна тенденція – інтеграція IDPS зі службами інтелектуального аналізу погроз і реагування на інциденти (приклад – продукти Cisco, FireEye), використання репутаційних сервісів (NSFOCUS). Інтеграція зі службами запобігання погроз дозволяє наділити IDPS інтелектуальними функціями виявлення шкідливого ПЗ. Новітні системи використовують евристичні методи, застосовують для обробки подій алгоритми штучного інтелекту й машинного навчання (Alert Logic), що поліпшує їхня здатність виявляти погрози й інциденти.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

Еволюція IDPS: галузеві тенденції

Розроблювачі продовжують удосконалювати системи IDPS, щоб підвищити їхню ефективність і зменшити число помилкових спрацьовувань. Згідно із прогнозами Gartner, до кінця 2020 року 60% IDPS будуть постачені аналітичними методами, зокрема, мова йде про машинне навчання й аналіз поведінки користувачів і мережевих елементів (сьогодні такими можливостями володіють менш 10% систем).

Функції виявлення й запобігання вторгнень усе більшою мірою передаються міжмережним екранам (NGFW), але, незважаючи на таку тенденцію, IDPS поки забезпечують кращу ефективність виявлення й централізоване виконання функцій попередження, виявлення вторгнень і відповідних дій у мережі.

Ряд виробників пропонують IDPS не тільки як фізичні пристрої для локального розгортання, але й у вигляді віртуальних машин, що розгортаються в хмарних середовищах; у числі таких продуктів – Alert Logic і Venustech. За прогнозом Gartner, до кінця 2020 року 70% нових автономних систем виявлення й запобігання вторгнень будуть розміщатися не традиційно, за міжмережним екраном, а в хмарі (публічному або частці).

Наприклад, хмарні сервіси IDPS уже надають Alert Logic, FireEye, Cisco і McAfee. Їхні рішення можуть розгортатися в хмарі як ВМ. Ряд виробників пропонують IDPS для Amazon Web Services (AWS) і Microsoft Azure.

Комбінування застосовуваних в IDPS методів допомагає розроблювачам переборювати недоліки кожного з них. Актуальним напрямком залишається підвищення продуктивності алгоритмів для поліпшення співвідношення якості виявлення / швидкість спрацьовування.

Паралельно йде робота над точністю алгоритмів для зменшення числа помилкових спрацьовувань.

3.3 Розробка функціональної схеми

Використання мобільних і хмарних обчислень, розширення способів доступу до корпоративних ресурсів і застосунків приводить до того, що в корпоративній мережі більше немає єдиної точки входу, і захист мережі необхідності розгортати як на границі, так і усередині мережі.

Системи виявлення й запобігання вторгнень у корпоративну мережу (Intrusion Detection System/Intrusion Prevention System – IDS/IPS) призначені для глибокого аналізу мережевого трафіку, виявлення й запобігання мережевих атак, несанкціонованої мережевої активності. Маючи інформацію про всі мережеві потоки в реальному часі, системи IDS/IPS виділяють і реагують на події, які можуть вплинути на IT-безпеку.

Блок аналітики IDS / IPS-систем:

- спрогнозувати можливі атаки й виявити пов'язані з ними уразливості;
- забезпечити контроль якості адміністрування з погляду безпеки, особливо в більших і складних мережах;
- одержати корисну інформацію про проникнення, які мали місце, з метою відновлення й коректування факторів, що викликали проникнення..

Системи запобігання вторгнень IPS можуть в автоматичному режимі блокувати мережеву атаку, наприклад, шляхом обриву з'єднання TCP/IP, або запобігти атаці, наприклад, видаливши шкідливий код з переданих даних.

Блок методів виявлення атак:

- аналіз сигнатур (шаблонів, по яких визначають атаку);
- аналіз аномального поведінки користувачів на основі статистичних методів і порівнянні характеристик мережевої активності із заданими граничними значеннями.

Основними проблемами при впровадженні систем IDS/IPS довгий час були великий відсоток помилкових спрацьовувань і необхідність тривалого тонкого налаштування під конкретне мережеве середовище, профілі трафіку.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

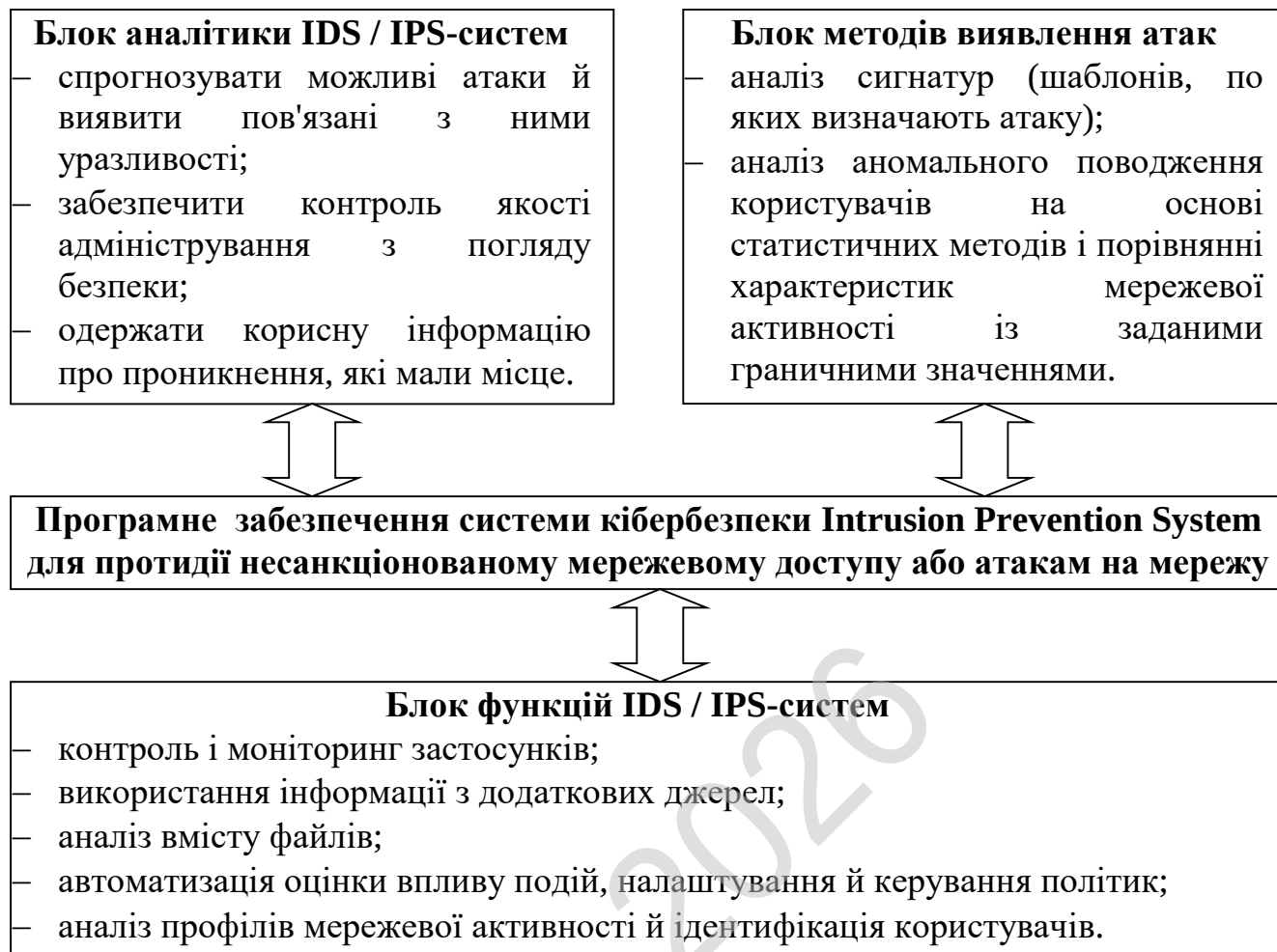


Рисунок 3.2 – Функціональна схема системи

Сучасні системи «наступного покоління» (Next Generation IPS – NG IPS) виступають як єдина платформа, що поєднує в собі переваги традиційної IPS і нові можливості.

Блок функцій IDS / IPS-систем:

- контроль і моніторинг застосунків;
- використання інформації з додаткових джерел (глобальні бази уразливостей, геолокація й т.п.);
- аналіз вмісту файлів;
- автоматизація оцінки впливу подій, налаштування й керування політик;
- аналіз профілів мережевої активності й ідентифікація користувачів.

Системи IDS/IPS нового покоління мають високий рівень автоматизації процесів полегшуючи роботу адміністраторів і надаючи рекомендації із захисту від погроз, характерних у конкретної IT-інфраструктурі.

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей. Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

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.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграми потоків даних містять чотири типи елементів:

- Зовнішні по відношенню до системи сутності.
- Потoki даних між елементами трьох попередніх типів.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).

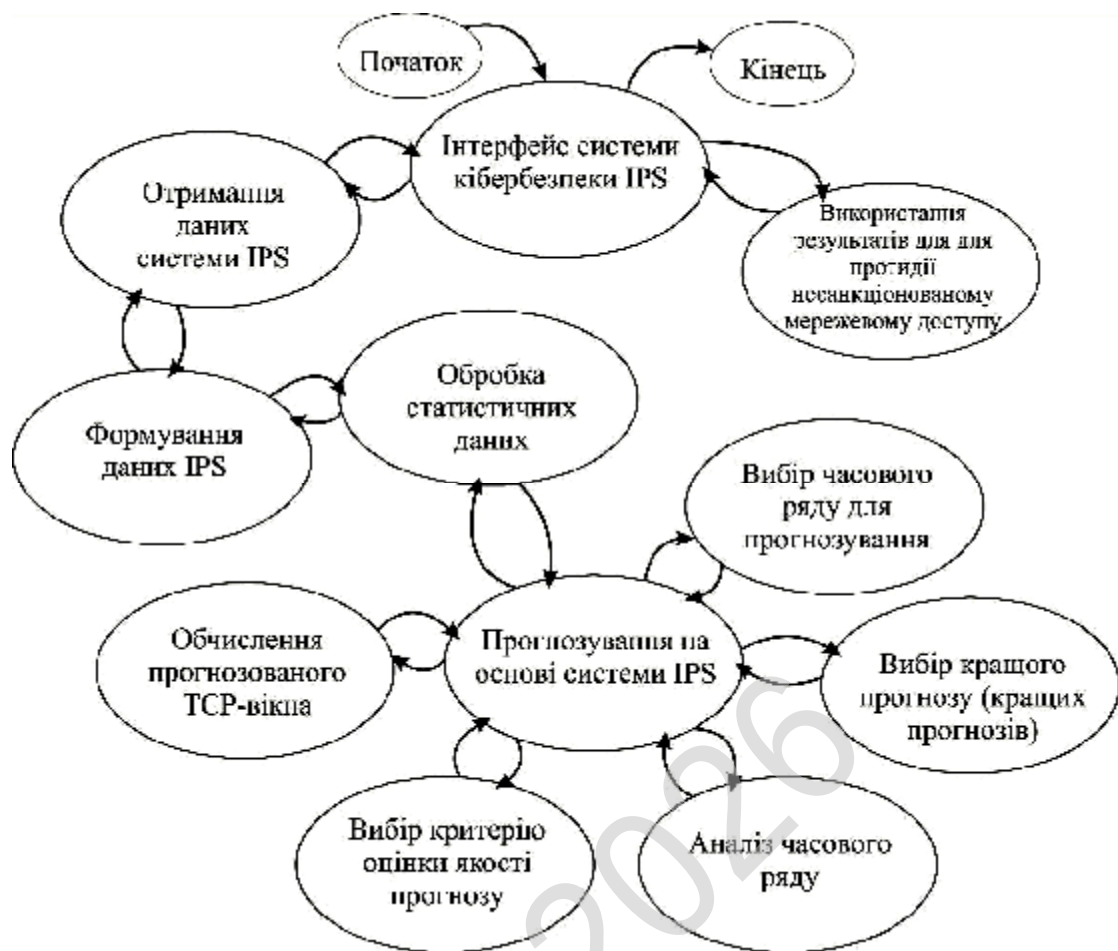


Рисунок 3.3 – Діаграма взаємодії процесів

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

Розглянемо реалізацію бакалаврської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень. Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо.

Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. 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 схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.

Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елемента.

Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елемента. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується.

Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи.

Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.

Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи). З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми.

Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

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 підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

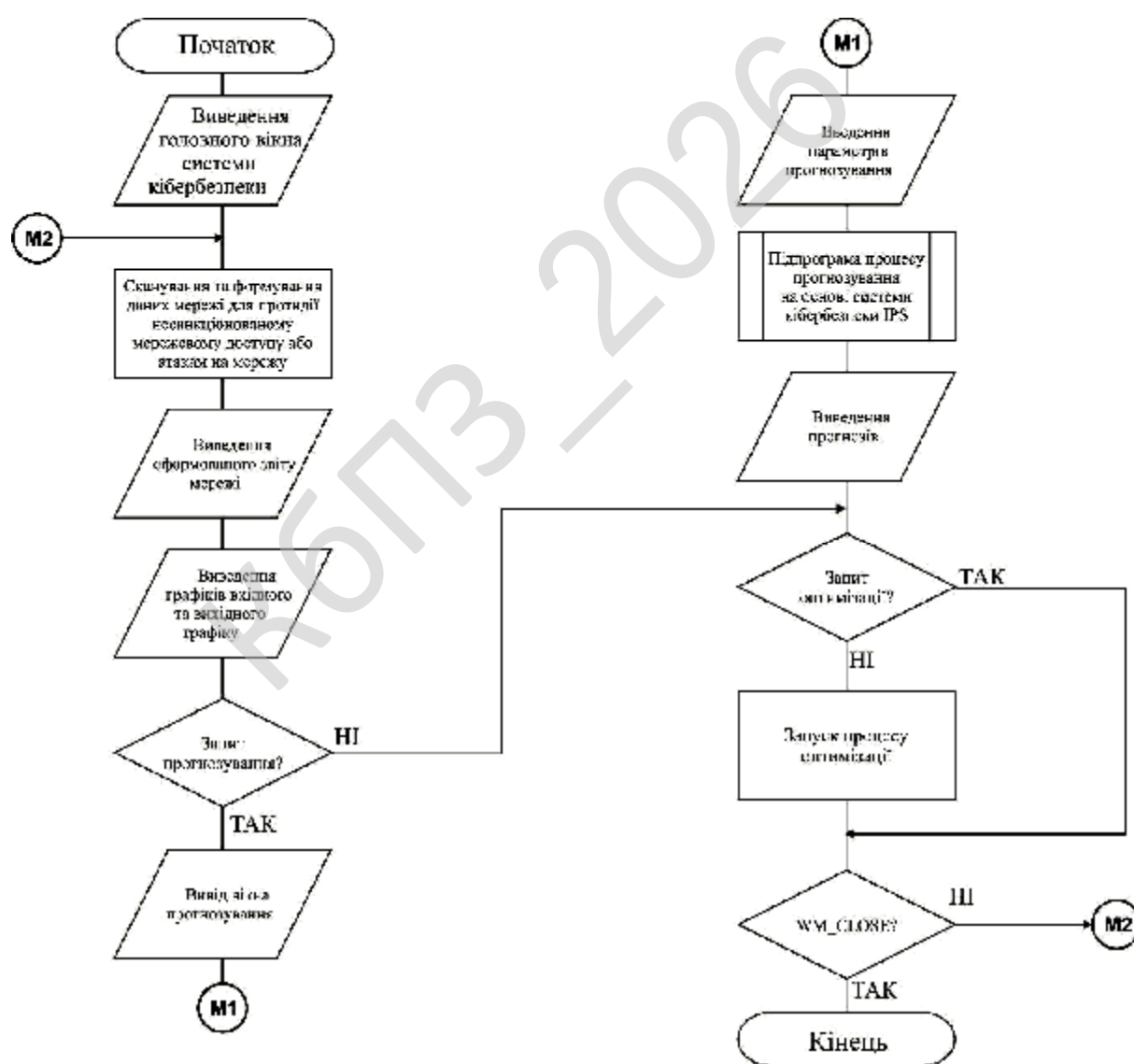


Рисунок 4.1 – Блок-схема основної програми

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.

Архітектура системи

Програмне забезпечення IPS на Python виконує виявлення та блокування підозрілої мережевої активності. Система аналізує події мережевого доступу, перевіряє IP адреси, порти, типи запитів, частоту звернень і наявність ознак атак. Основна мета системи полягає у протидії несанкціонованому доступу, скануванню портів, brute force спробам, підозрілим HTTP запитам та атакам на мережеві сервіси.

Система має модульну архітектуру. Модуль збору подій отримує дані про мережеву активність. Модуль аналізу перевіряє події за правилами безпеки. Модуль поведінкового контролю рахує частоту підключень з однієї IP адреси. Модуль реагування додає небезпечні адреси до списку блокування. Модуль журналювання зберігає події, рішення системи та причини блокування.

Система приймає мережеві події у вигляді структурованих записів. Вона визначає підозрілі IP адреси, фіксує перевищення кількості запитів, виявляє спроби підбору паролів, реагує на сканування портів, перевіряє HTTP запити на небезпечні шаблони та формує журнал інцидентів. У межах бакалаврської роботи така система демонструє принцип роботи IPS без втручання у реальну мережеву інфраструктуру. Розглянемо реалізацію у вихідному коді ПЗ.

```
import sqlite3
from datetime import datetime, timedelta
from collections import defaultdict
```

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

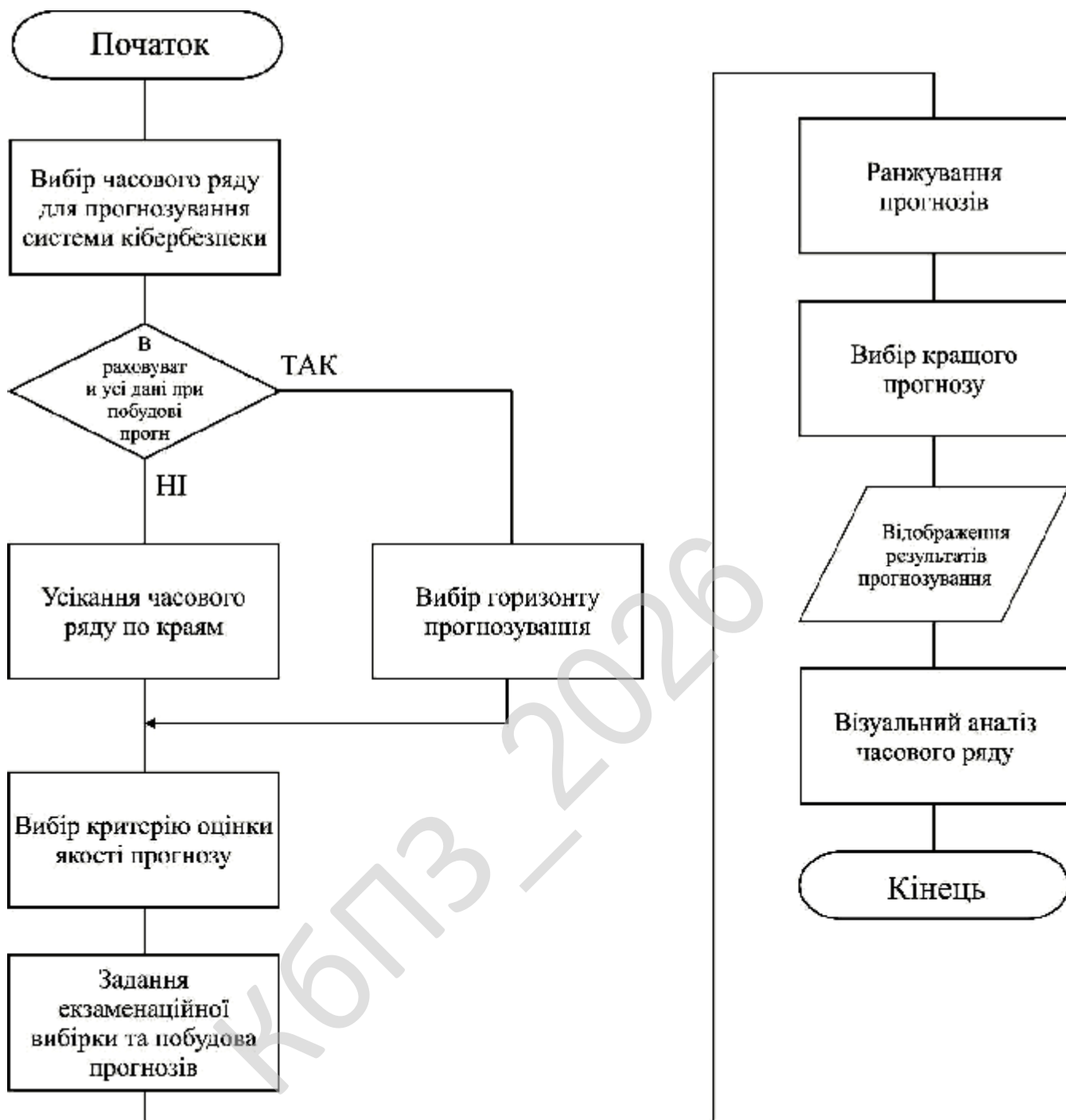


Рисунок 4.2 – Блок-схема роботи підпрограми

```

class IPSDatabase:
    def __init__(self, db_name="ips_events.db"):
        self.db_name = db_name
        self.create_tables()
    def create_tables(self):
        connection = sqlite3.connect(self.db_name)
        cursor = connection.cursor()
  
```

```

cursor.execute("""
CREATE TABLE IF NOT EXISTS events (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    event_time TEXT,
    source_ip TEXT,
    destination_port INTEGER,
    protocol TEXT,
    event_type TEXT,
    action TEXT,
    reason TEXT
)
""")
cursor.execute("""
CREATE TABLE IF NOT EXISTS blocked_ips (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    source_ip TEXT UNIQUE,
    blocked_at TEXT,
    reason TEXT
)
""")
connection.commit()
connection.close()
def save_event(self, event, action, reason):
    connection = sqlite3.connect(self.db_name)
    cursor = connection.cursor()
    cursor.execute("""
INSERT INTO events (
    event_time,
    source_ip,
    destination_port,
    protocol,
    event_type,
    action,
    reason
)
VALUES (?, ?, ?, ?, ?, ?, ?)
""", (
    datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
    event.get("source_ip"),
    event.get("destination_port"),
    event.get("protocol"),
    event.get("event_type"),

```

```

        action,
        reason
    ))
    connection.commit()
    connection.close()

def block_ip(self, source_ip, reason):
    connection = sqlite3.connect(self.db_name)
    cursor = connection.cursor()
    cursor.execute("""
INSERT OR IGNORE INTO blocked_ips (
    source_ip,
    blocked_at,
    reason
)
VALUES (?, ?, ?)
""", (
    source_ip,
    datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
    reason
))
    connection.commit()
    connection.close()

class IntrusionPreventionSystem:
    def __init__(self):
        self.database = IPSTDatabase()
        self.blocked_ips = set()
        self.request_history = defaultdict(list)
        self.port_history = defaultdict(set)

        self.dangerous_patterns = [
            "union select",
            "drop table",
            "../",
            "<script>",
            "admin' or '1'='1",
            "cmd.exe",
            "/etc/passwd"
        ]

    def analyze_event(self, event):
        source_ip = event.get("source_ip")

```

```

destination_port = event.get("destination_port")
payload = event.get("payload", "").lower()
event_type = event.get("event_type", "unknown")
if source_ip in self.blocked_ips:
    return "BLOCK", "IP адреса вже знаходиться у списку блокування"
if self.detect_port_scan(source_ip, destination_port):
    return "BLOCK", "Виявлено ознаки сканування портів"
if self.detect_brute_force(source_ip, event_type):
    return "BLOCK", "Виявлено багаторазові спроби авторизації"
if self.detect_dangerous_payload(payload):
    return "BLOCK", "Виявлено небезпечний вміст запиту"
if self.detect_request_flood(source_ip):
    return "BLOCK", "Виявлено надмірну кількість запитів"
return "ALLOW", "Подія не має критичних ознак атаки"
def detect_port_scan(self, source_ip, destination_port):
    self.port_history[source_ip].add(destination_port)
    if len(self.port_history[source_ip]) > 5:
        return True
    return False
def detect_brute_force(self, source_ip, event_type):
    if event_type != "login_failed":
        return False
    current_time = datetime.now()
    self.request_history[source_ip].append(current_time)
    recent_events = [
        item for item in self.request_history[source_ip]
        if current_time - item <= timedelta(minutes=1)
    ]
    self.request_history[source_ip] = recent_events
    if len(recent_events) >= 4:
        return True
    return False
def detect_dangerous_payload(self, payload):
    for pattern in self.dangerous_patterns:
        if pattern in payload:
            return True
    return False
def detect_request_flood(self, source_ip):
    current_time = datetime.now()
    self.request_history[source_ip].append(current_time)
    recent_events = [
        item for item in self.request_history[source_ip]

```

```

        if current_time - item <= timedelta(seconds=10)
    ]
    self.request_history[source_ip] = recent_events
    if len(recent_events) > 10:
        return True
    return False

def process_event(self, event):
    action, reason = self.analyze_event(event)
    source_ip = event.get("source_ip")
    if action == "BLOCK":
        self.blocked_ips.add(source_ip)
        self.database.block_ip(source_ip, reason)
    self.database.save_event(event, action, reason)
    return {
        "source_ip": source_ip,
        "action": action,
        "reason": reason
    }

def main():
    ips = IntrusionPreventionSystem()
    #Імітація мережевих подій для перевірки роботи IPS
    test_events = [
        {
            "source_ip": "192.168.1.45",
            "destination_port": 80,
            "protocol": "TCP",
            "event_type": "http_request",
            "payload": "GET /index.html"
        },
        {
            "source_ip": "192.168.1.50",
            "destination_port": 22,
            "protocol": "TCP",
            "event_type": "login_failed",
            "payload": "wrong password"
        },
        {
            "source_ip": "192.168.1.60",
            "destination_port": 8080,
            "protocol": "TCP",
            "event_type": "http_request",

```

```

        "payload": "UNION SELECT password FROM users"
    }
]
for event in test_events:
    result = ips.process_event(event)
    print(
        f"IP {result['source_ip']} "
        f"Дія {result['action']} "
        f"Причина {result['reason']}"
    )
if __name__ == "__main__":
    main()

```

Розглянемо поетапний алгоритм роботи розробленого ПЗ:

– Вибір часового ряду завантаженості трафіку для прогнозування. На початку роботи експертові необхідно визначитися, на підставі якого часового ряду завантаженості трафіку необхідно провести прогнозування. Після вибору часового ряду він передається в систему прогнозування.

– Візуалізація часового ряду. Переданий у систему прогнозування часовий ряд завантаженості трафіку, відображається у вигляді графіка, на якому по осі абсцис відкладаються часові такти, а по осі ординат – значення часового ряду завантаженості трафіку.

– Візуальний аналіз часового ряду завантаженості трафіку. У ході візуального аналізу часового ряду завантаженості трафіку експерт ухвалює рішення щодо того, чи варто враховувати всі дані ряду при побудові прогнозу й, при необхідності, виключає необхідну кількість послідовних тактів ліворуч. Таким чином, ігноруються «старі» дані. У випадку, якщо експертові хочеться зіставити прогноз із реальними даними, він може виключити з розгляду кілька послідовних тактів праворуч (ігноруються «нові» дані).

– «Усікання» часового ряду завантаженості трафіку по краях. Експерт вказує, яку кількість тактів праворуч і ліворуч варто виключити з розгляду, після чого ці такти ігноруються системою при побудові прогнозу завантаженості трафіку.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

– Вибір об'єкту прогнозування. На цьому етапі експерт приймає рішення, на яку кількість тактів уперед необхідно побудувати прогноз, з огляду при цьому на кількість спостережених значень у вихідному часовому ряді. Чим більше це число, тим, як правило, більш достовірним виходить прогноз.

– Настроювання набору прогнозних моделей. Експерт визначає, які прогнозні моделі й з яким діапазоном параметрів варто використовувати при побудові прогнозу завантаженості трафіку.

– Вибір критерію оцінки якості прогнозу. Експерт вибирає формальну постановку задачі прогнозування із запропонованих системою прогнозування завантаженості трафіку. На підставі обраного критерію система прогнозування буде визначати, які прогнози «краще», а які «гірше».

– Завдання екзаменаційної вибірки. Залежно від критерію оцінки якості прогнозу й обраних прогнозних моделей, експерт вибирає крапки вихідного часового ряду, які необхідно включити в екзаменаційну вибірку, по крапках якої будуть перевірятися побудовані прогнози. За замовчуванням екзаменаційними вважаються τ останніх тактів часового ряду.

Розглянемо визначення API. Це прикладний програмний інтерфейс (Application Programming Interface, API) – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

Спрощено – це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного забезпечення. API може бути для веб-базованих систем, операційних систем, баз даних, апаратного забезпечення, програмних бібліотек.

Одним з найпоширеніших призначень API є надання набору широко використовуваних функцій, наприклад для малювання вікна чи іконок на екрані. API є абстрактним поняттям – програмне забезпечення, що пропонує деякий API, часто називають реалізацією (implementation) даного API. У багатьох випадках API є частиною набору розробки програмного забезпечення, водночас, набір розробки може включати як API, так і інші інструменти/апаратне забезпечення,

отже ці два терміни не є взаємозамінювані. Високорівневі API часто програють у гнучкості. Виконання деяких функцій нижчого рівня стає набагато складнішим, або навіть неможливим.

В об'єктно-орієнтованих мовах, прикладний програмний інтерфейс зазвичай включає в себе опис набору визначень класу, з набором форм поведінки, пов'язаних з цими класами. Це абстрактне поняття пов'язане з реальними функціями, які надані або надаватимуться, класами, які реалізуються в методах класу.

Прикладний програмний інтерфейс в даному випадку можна розглядати як сукупність всіх методів, які публічно доступні в класах (зазвичай званий інтерфейс класу). Це означає, що прикладний програмний інтерфейс вказує методи, за допомогою яких взаємодіє з об'єктами, отриманими з визначень класів і обробляє їх.

У більш загальному плані можна визначити Прикладний Програмний Інтерфейс як сукупність усіх видів об'єктів, які можна вивести з визначення класу, і пов'язаних з ними можливих варіантів поведінки.

Наприклад: клас, що представляє Stack, може просто виставити публічно два методи Push() (для додавання нового елемента в стек) і Pop() (для вилучення останнього пункту, ідеально розташований на вершині стека).

У цьому випадку Прикладний Програмний Інтерфейс може бути інтерпретованим як два методи pop() і push(), або, більш широко, використовується варіант, коли можна використовувати елемент типу Stack, який реалізує поведінку стека, надаючи йому можливість для додавання / видалення елементів з вершини. Друга інтерпретація видається більш доречною в дусі об'єктно-орієнтованого підходу. Якість документації, пов'язаної з Прикладним Програмним Інтерфейсом, є часто ключовим фактором, що визначає його успішність з точки зору простоти використання.

ППІ, як правило, пов'язаний із бібліотеками програмного забезпечення: ППІ описує і вказує очікувану поведінку в той час, як бібліотека є фактичною

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

реалізацією даного набору правил. Один ППІ може мати декілька реалізацій (або жодної, будучи абстрактним) у вигляді різних бібліотек, які мають такий же інтерфейс.

Прикладний програмний інтерфейс також може бути пов'язаним з платформами програмування: платформа може бути заснована на кількох бібліотеках реалізує декілька інтерфейсів ППІ, але на відміну від звичайного використання ППІ, доступ до поведінки вбудований в платформу опосередкований шляхом розширення його змісту новими класами і вставлений в саму платформу. Крім того, загальний потік управління програми може бути під контролем абонента.

Прикладний програмний інтерфейс може бути також реалізацією протоколу.

Коли ППІ реалізує протокол, він може бути заснованим на проксі-методах віддалених викликів, що засновані на протоколі зв'язку. Роль ППІ може заключатися саме в тому, щоб приховати деталі транспортного протоколу. Наприклад: RMI є ППІ, який реалізує протокол або JRMP IIOP як RMI-IIOP.

Протоколи, як правило, розподіляються між різними технологіями і зазвичай дозволяють різним технологіям обмінюватися інформацією, діючи як абстракція між двома світами. ППІ, як правило, є специфічним для конкретної технології: звідси, інтерфейси даної мови не можуть бути використані на інших мовах, якщо виклики функції не будуть перетворені з конкретної адаптації бібліотеки.

Деякі мови, серед яких такі, що працюють на віртуальних машинах (наприклад: мови, сумісні з NET CLI середовища CLR і JVM сумісних мов у віртуальній машині Java) можуть ділитися програмними інтерфейсами.

У цьому випадку віртуальна машина дозволяє мові взаємодії завдяки спільному знаменнику віртуальної машини, що абстрагується від конкретної мови, використовувати проміжний байт-код і його мову. При використанні прикладного програмного інтерфейсу в контексті веб-розробки, як правило, ППІ

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

визначається набором повідомлень запиту HTTP, також визначається структура повідомлень-відповідей, зазвичай у розширенні мови розмітки XML або в форматі об'єктного запису JavaScript (JSON). У той час як прикладний програмний інтерфейс у Web історично був практично синонімом для веб-служби, останнім часом тенденція змінилась (так званий Web 2.0) на відхід від Simple Object Access Protocol (SOAP) на основі веб-сервісів і сервіс-орієнтованої архітектури (SOA) на більш прямі передачі репрезентативного стану (REST) стилів веб-ресурсів та ресурсів-орієнтованої архітектури (ROA).

Частина цієї тенденції пов'язана з рухом Семантичного веб-ресурсу до Опису Платформ (RDF), Концепції розвитку веб-технологій інженерних онтологій. Прикладні програмні інтерфейси у Web, що дозволяють комбінувати декількома прикладними програмними інтерфейсами в нові додатки називають гібридними.

Розглянемо формат що використовується – JSON (JavaScript Object Notation, укр. запис об'єктів JavaScript, вимовляється джейсон) – це текстовий формат обміну даними між комп'ютерами.

JSON базується на тексті, може бути прочитаним людиною. Формат дозволяє описувати об'єкти та інші структури даних. Цей формат головним чином використовується для передачі структурованої інформації через мережу (завдяки процесу, що називають серіалізацією). Розробив і популяризував формат Дуглас Крокфорд.

JSON знайшов своє головне призначення у написанні веб-програм, а саме при використанні технології AJAX. JSON, що використовується в AJAX, виступає як заміна XML (використовується в AJAX) під час асинхронної передачі структурованої інформації між клієнтом та сервером. При цьому перевагою JSON перед XML є те, що він дозволяє складні структури в атрибутах, займає менше місця і прямо інтерпретується за допомогою JavaScript в об'єкти.

JSON з'явився через необхідність обміну даними з сервером у реальному часі без використання плагінів для браузерів, flash-додатків або Java-апплетів, які

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

використовувались скрізь на початку 2000-х років. Дуглас Крокфорд був тим, хто активно просував новий на той час формат. Він з колегами хотів створити технологію, яка використовувала б можливості звичайного браузера давала б веб-розробникам можливість створювати веб-додатки із постійним двостороннім зв'язком із веб-сервером.

JSON вперше був використаний в проєкті в Communities.com для Cartoon Network, він дозволяв обмінюватись повідомленнями і одночасно маніпулювати DHTML-елементами.

Веб-сайт JSON.org було запущено 2002 року. У грудні 2005-го року Yahoo! почав переводити деякі зі своїх веб-сервісів на роботу з JSON. Google взялася до роботи з технологією у своєму веб-протоколі GData у грудні 2006-го року.

Використання

За рахунок своєї лаконічності в порівнянні з XML, формат JSON може бути більш придатним для серіалізації складних структур.

Якщо говорити про веб-застосунки, в такому ключі він доречний в задачах обміну даними як між браузером і сервером (AJAX), так і між самими серверами (програмні HTTP-інтерфейси).

Формат JSON так само добре підходить для зберігання складних динамічних структур в реляційних базах даних або файлового кеші.

Приклад використання JSON (JavaScript)

```
const ajaxData = '{"name": "wiki", "fname": "pedia", "rates": [1, 4, 5, 6]}'  
  
const ajaxObj = JSON.parse(ajaxData)  
console.log(`${ajaxObj.name}_${ajaxObj.rates[2]}`) /* Виведе "wiki_5"*/
```

Синтаксис

JSON будується на двох структурах:

1. Набір пар назва/значення. У різних мовах це реалізовано як об'єкт, запис, структура, словник, хеш-таблиця, список з ключем або асоціативним масивом.
2. Впорядкований список значень. У багатьох мовах це реалізовано як масив, вектор, список, або послідовність.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Це універсальні структури даних. Теоретично всі сучасні мови програмування підтримують їх у тій чи іншій формі. Оскільки JSON використовується для обміну даними між різними мовами програмування, то є сенс будувати його на цих структурах.

У JSON використовуються такі їхні форми:

1. Об'єкт – це послідовність пар назва/значення. Об'єкт починається з символу { і закінчується символом }. Кожне значення слідує за : і пари назва/значення відділяються комами.

2. Масив – це послідовність значень. Масив починається символом [і закінчується символом]. Значення відділяються комами.

3. Значення може бути рядком в подвійних лапках, або числом, або логічними true чи false, або null, або об'єктом, або масивом. Ці структури можуть бути вкладені одна в одну.

4. Рядок – це послідовність з нуля або більше символів юнікода, обмежена подвійними лапками, з використанням escape-послідовностей, що починаються зі зворотної косої риски \. Символи представляються простим рядком. Тип Рядок (String) дуже схожий на String в мовах C і Java. Число теж дуже схоже на C- або Java-число, за винятком того, що вісімкові та шістнадцяткові формати не використовуються. Пропуски можуть бути вставлені між будь-якими двома лексемами.

Наступний приклад показує JSON представлення об'єкта, що описує людину. У об'єкті є рядкові поля імені і прізвища, об'єкт, що описує адресу, і масив, що містить список телефонів.

```
{
  "firstName": "Іван",
  "phoneNumbers": [
    "044 123-1234",
    "050 123-4567"
  ]
}
```

Використання JSON в AJAX. Наступний фрагмент коду JavaScript показує, як клієнт може використати XMLHttpRequest для запиту об'єктів в форматі JSON

з серверу. (Серверна частина коду пропущена, вона просто повертає на запит URL рядок у JSON форматі).

Треба відзначити, що тут використання XMLHttpRequest не є кросс-браузерним (за деталями звертайтеся на сторінку XMLHttpRequest), і код зазнаватиме незначних модифікацій в різних версіях оглядачів Internet Explorer, Opera, Safari або Mozilla. Використання запиту XMLHttpRequest обмежено правилом одного джерела (same origin policy): URL, що відповідає на запит, має посилатися на той же сайт, що обслуговує сторінку, що ініціювала запит.

Оглядачі можуть також використовувати тег <iframe> для асинхронного запиту JSON-даних в кросс-браузерному варіанті, або використати просте перенаправлення `<form action="url_to_cgi_script" target="name_of_hidden_iframe">`. Такий підхід був поширений до приходу популярного нині запиту XMLHttpRequest.

Динамічний тег <script> також можна використати для підвантаження JSON-даних. Ця техніка можлива, щоб обійти надто суворе правило одного джерела, але вона не є безпечною. Запит JSONRequest пропонується, як безпечніша альтернатива.

Питання безпеки. Хоча JSON призначений для передачі даних в серіалізованому вигляді, його синтаксис відповідає синтаксису JavaScript і це створює низку проблем безпеки. Часто для обробки даних, отриманих від зовнішнього джерела у форматі JSON, до них застосовується функція eval() без якої-небудь попередньої перевірки.

JavaScript eval(). Оскільки JSON представляється синтаксично правильним фрагментом коду JavaScript, природним способом розбору JSON-даних в JavaScript-програмі є використання вбудованої в JavaScript функції eval(), яка призначена для обчислення JavaScript-виразів. При цьому підході відпадає необхідність у використанні додаткових парсерів.

Техніка використання eval() робить систему вразливою, якщо джерело JSON-даних, що використовуються, не відноситься до надійних. Такими даними

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

може виступати шкідливий JavaScript-код для атак за допомогою ін'єкції коду. За допомогою цієї вразливості можливо здійснювати крадіжку даних, підробку автентифікації. Проте, вразливість можна усунути за рахунок використання додаткових засобів перевірки даних на коректність. Наприклад, до виконання eval() отримані від зовнішнього джерела дані можуть перевірятися за допомогою регулярних виразів. У RFC, що визначає JSON пропонується використовувати такий код для перевірки його відповідності формату JSON

```
const my_JSON_object = !(/^[^,:{}\[\]\0-9.\-+Eaeflnr-u \n\r\t]/.test(
text.replace(/"(\\"|\\.|[^\\"\\])*"/g, ''))) &&
eval('(' + testedData + ')');
```

Як безпечніша альтернатива eval() була запропонована нова функція parseJSON(), здатна обробляти тільки JSON-дані.

Вбудований JSON. Останні версії веб-браузерів мають вбудовану підтримку JSON і здатні його обробляти без виклику функції eval(), що призводить до описаної проблеми. Обробка JSON у такому разі зазвичай здійснюється швидше. Так у червні 2009 року вбудовану підтримку JSON мали такі браузери: Mozilla Firefox 3.5+; SeaMonkey 2; Thunderbird 3; Microsoft Internet Explorer 8; Opera 10.5; Браузери, засновані на WebKit (наприклад, Google Chrome, Apple Safari).

Принаймні дві популярні бібліотеки JavaScript використовують вбудований JSON у разі його доступності: jQuery; Dojo.

Підробка крос-доменного запиту. Непродумане використання JSON робить сайти вразливими до підробки міжсайтових запитів (CSRF або XSRF). Оскільки тег <script> допускає використання джерела, що не належить до того ж домену, що і використовуваний ресурс, це дозволяє виконувати код даних, представлених у форматі JSON, в контексті довільної сторінки, що робить можливою компрометацію паролів або іншої конфіденційної інформації користувачів, що пройшли авторизацію на іншому сайті.

Це є проблемою тільки у разі вмісту в JSON-даних конфіденційної інформації, яка може бути компрометована третьою стороною і якщо сервер

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

розраховує на політику одного джерела, блокуючи доступ до даних при виявленні зовнішнього запиту. Це не є проблемою, якщо сервер визначає допустимість запиту, надаючи дані тільки у разі його коректності. HTTP cookie не можна використовувати для визначення цього. Виключне використання HTTP cookie використовується підрубкою міжсайтових запитів.

Розширення, JSONP. JSONP або «JSON з підкладкою» є розширенням JSON, коли назва функції зворотного виклику вказується як вхідний аргумент. Спочатку ідея була запропонована в блозі MacPython в 2005 році, і в наш час використовується багатьма Web 2.0 застосунками, такими, як Dojo Toolkit Applications, Google Toolkit Applications і zanox Web Services.

Подальші розширення цього протоколу були запропоновані з урахуванням введення додаткових аргументів, як, наприклад, у разі JSONPP за підтримки S3DB вебсервісів.

Оскільки JSONP використовує скрипт-теги, виклики по суті відкриті світові. З цієї причини JSONP може бути недоречними для зберігання конфіденційних даних.

Включення скриптових тегів від віддалених сайтів дозволяє їм передати будь-який контент на сайті. Якщо віддалений сайт має вразливості, які дозволяють виконати ін'єкції JavaScript, то початковий сайт також може бути ними зачеплений.

Розширення, BSON. BSON – це бінарна форма представлення простих структур даних і асоціативних масивів (які називають об'єктами або документами). Назва «BSON» заснована на визначенні JSON і неофіційно означає «Binary JSON» (бінарний JSON).

JSON Reference. Стандарт JSON не описує посилання на інші об'єкти або частини, але існує чернетка стандарту IETF для посилань на об'єкти на основі JSON. Посилання дозволяють здійснювати трансклюзію – вставляти одні документи в інші.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

JSON Reference – це JSON-об'єкт з ключем "\$ref" (всі інші ключі ігноруються) і значенням стрічкового типу що містить URI, наприклад:

```
{ "$ref": "http://example.com/example.json#/foo/bar" }
```

Якщо URI містить ідентифікатор фрагмента (в прикладі вище "/foo/bar"), він інтерпретується як JSON Pointer.

Модуль dojoy.json.ref в Dojo toolkit, забезпечує підтримку декількох форм JSON Reference.

Розглянемо протокол SNMP (Simple Network Management Protocol, простий протокол керування мережею) – це протокол керування мережами зв'язку на основі архітектури TCP/IP.

На основі концепції TMN в 1980–1990 р. різними органами стандартизації був вироблений ряд протоколів керування мережами передачі даних з різним спектром реалізації функцій TMN. До одного з типів таких протоколів керування належить Simple Network Management Protocol (SNMP).

SNMP – це технологія, покликана забезпечити керування й контроль за пристроями й програмами в мережі зв'язку шляхом обміну керуючою інформацією між агентами, що розташовуються на мережних пристроях, і менеджерами, розташованими на станціях керування. SNMP визначає мережу як сукупність мережних керуючих станцій й елементів мережі (головні машини, шлюзи й маршрутизатори, термінальні сервери), які спільно забезпечують адміністративні зв'язки між мережними керуючими станціями й мережними агентами. SNMP різних версій присвячений цілий ряд рекомендацій IETF (RFC).

Зазвичай при використанні SNMP присутні керовані та керуючі системи. До складу керованої системи входить компонент, який називається агентом, який відправляє звіти керуючій системі. По суті SNMP агенти передають управлінську інформацію на керуючі системи як змінні (такі як «вільна пам'ять», «ім'я системи», «кількість працюючих процесів» тощо).

Керуюча система може отримати достовірну інформацію через операції протоколу GET, GETNEXT і GETBULK. Агент може самостійно без запиту надсилати дані, використовуючи операцію протоколу TRAP або INFORM.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

Управляючі системи можуть також відправляти конфігураційні оновлення або контролюючі запити, використовуючи операцію SET для безпосереднього управління системою. Операції конфігурування та управління використовуються тільки тоді, коли потрібні зміни у мережній інфраструктурі. Операції моніторингу зазвичай виконуються на регулярній основі.

Змінні, доступні через SNMP, організовані в ієрархії. Ці ієрархії та інші метадані (такі як тип і опис змінної) описуються Базами Керуючої Інформації' (Management Information Bases (MIBs)).

SNMP не визначає, яку інформацію (які змінні) керована система повинна надавати. Навпаки, SNMP використовує розширювану модель, в якій доступна інформація визначається Базами Керуючої Інформації (MIB – Management Information Base). Бази Керуючої Інформації описують структуру керуючої інформації пристроїв. Вони використовують ієрархічний адресний простір імен, що містить унікальний ідентифікатор об'єкта (object identifier (OID)). Грубо кажучи, кожен унікальний ідентифікатор об'єкта ідентифікує змінну, яка може бути прочитана чи встановлена через SNMP. MIB'и використовують нотацію, визначену в ASN.1. Ієрархія MIB може бути зображена як дерево з безіменним коренем, рівні якого приписані різними організаціями. На найвищому рівні MIB OID'и належать різним організаціям, що займаються стандартизацією, в той час як на нижчих рівнях OID'и виділяються асоційованими організаціями. Ця модель забезпечує управління на всіх шарах мережної моделі OSI, адже MIB'и можуть бути визначені для будь-яких типів даних і операцій.

Керований об'єкт – це одна з будь-якого числа характеристик, специфічних для керованого пристрою. Керований об'єкт включає в себе один або більше екземплярів об'єкта (що ідентифікуються за OID), які насправді є змінними. Існує два типи керованих об'єктів: Скалярні об'єкти визначають єдиний екземпляр об'єкта; Табличні об'єкти визначають множинні, пов'язані екземпляри об'єктів які групуються в таблицях MIB. Прикладом керованого об'єкта може бути atInput, який є скалярним об'єктом що містить єдиний екземпляр об'єкта, ціле число, яке

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

показує загальну кількість вхідних пакетів AppleTalk на мережний інтерфейс маршрутизатора. Ідентифікатор об'єкта (OID) унікально ідентифікує керований об'єкт в ієрархії MIB. В телекомунікаціях і комп'ютерних мережах, ASN.1 є стандартною гнучкою нотацією для опису структур даних, що служать для кодування, передачі і декодування даних. ASN.1 являє собою набір правил для опису структури об'єктів, незалежних від специфічних для обладнання методик кодування, і формальну нотацію, яка дозволяє уникнути неоднозначностей.

ASN.1 це єдиний ISO та ITU-T стандарт, спочатку визначений у 1984 році як частина стандарту CCITT X.409: 1984. Пізніше, в 1988 році, завдяки його широкому застосуванню, він був виділений в окремий стандарт X.208. Значно переглянута версія 1995року описана в X.680. Адаптована підмножина ASN.1 Структура Керуючої Інформації (SMI) описана в протоколі SNMP для визначення наборів пов'язаних MIB об'єктів, також званих MIB модулями. SNMP працює на прикладному рівні TCP/IP (сьомий рівень моделі OSI). Агент SNMP отримує запити по UDP-порту 161. Менеджер може посилати запити з будь-якого доступного порту джерела на порт агента. Відповідь агента буде відправлений назад на порт джерела на менеджері. Менеджер отримує повідомлення (Traps і InformRequests) по порту 162. Агент може генерувати повідомлення з будь-якого доступного порту. При використанні TLS або DTLS запити виходять по порту 10161, а пастки відправляються на порт 10162.

У SNMPv1 зазначено п'ять основних протокольних одиниць обміну (protocol data units – PDU). Ще дві PDU, GetBulkRequest і InformRequest, були введені в SNMPv2 і перенесені в SNMPv3.

Нижче перераховані сім протокольних одиниць обміну SNMP:

1. GetRequest. Запит від менеджера до об'єкту для отримання значення змінної або списку змінних. Необхідні змінні вказуються в полі variable bindings (розділ поля values при цьому не використовується). Отримання значень зазначеної змінної повинно бути виконано агентом як Атомарна операція. Менеджеру буде повернений Response (відповідь) з поточними значеннями.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

2. SetRequest. Запит від менеджера до об'єкту для зміни змінної або списку змінних. Зв'язані змінні вказуються в тілі запиту. Зміни всіх зазначених змінних повинні бути виконані агентом як атомарна операція. Менеджеру буде повернений Response з (поточними) новими значеннями змінних.

3. GetNextRequest. Запит від менеджера до об'єкту для виявлення доступних змінних і їх значень. Менеджеру буде повернений Response зі зв'язаними змінними для змінної, яка є наступною в базі MIB в лексикографічному порядку. Обхід всієї бази MIB агента може бути проведений ітераційним використанням GetNextRequest, починаючи з OID 0. Рядки таблиці можуть бути прочитані, якщо вказати в запиті OID-и колонок в пов'язаних змінних.

4. GetBulkRequest. Покращена версія GetNextRequest. Запит від менеджера до об'єкту для численних ітерацій GetNextRequest. Менеджеру буде повернений Response з декількома пов'язаними змінними, обійденими починаючи зі пов'язаної змінної (змінних) в запиті. Специфічні для PDU поля non-repeaters і max-repetitions використовуються для контролю за поведінкою відповіді. GetBulkRequest був введений в SNMPv2.

5. Response. Повертає зв'язані змінні і значення від агента менеджеру для GetRequest, SetRequest, GetNextRequest, GetBulkRequest і InformRequest. Повідомлення про помилки забезпечуються полями статусу помилки і індексу помилки.

Ця одиниця використовується як відповідь і на Get-, і на Set-запити, в SNMPv1 називається GetResponse.

6. Trap. Асинхронне повідомлення від агента – менеджеру. Включає в себе поточне значення sysUpTime, OID, що визначає тип trap (пастки), і необов'язкові зв'язані змінні. Адресація одержувача для пасток визначається за допомогою змінних trap-конфігурації в базі MIB. Формат trap-повідомлення був змінений в SNMPv2 і PDU перейменували в SNMPv2-Trap.

7. InformRequest. Асинхронне повідомлення від менеджера менеджеру або від агента менеджеру. Повідомлення від менеджера менеджеру були можливі вже

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

в SNMPv1 (за допомогою Trap), але SNMP зазвичай працює на протоколі UDP, в якому доставка повідомлень не гарантована, і не повідомляється про втрачені пакетах. InformRequest виправляє це зворотним відправленням підтвердження про отримання. Одержувач відповідає Response-му, що повторює всю інформацію з InformRequest. Цей PDU був введений в SNMPv2.

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою Sinople – симетричний блоковий криптоалгоритм, побудований на основі незбалансованої «мережі Фейстеля». Алгоритм розроблено у 2003 році.

Основні вимоги до алгоритму при його розробці:

- Можливість програмної і апаратної реалізації.
- Висока швидкість.
- Простота.
- Низькі вимоги до пам'яті.
- Високий рівень безпеки.

Алгоритм заснований на 32-розрядних операціях і має 64 раунду, серед яких два типи – С і D. D раунди спроектовані для досягнення максимальної дифузії, С раунди – для досягнення перемішування. F-функція D раунду використовує один з елементів блоку даних (D[3]) та поточного з'єднання (K[r]) для трансформації 3-х елементів блоку даних. F-функція С раунду, навпаки, використовує перші три елемента блоку даних і поточний з'єднання (K[r]) для трансформації останнього елемента блоку даних (D[3]). Раунди D-типу виконуються до раундів С-типу. Додавання ключів з даними проводиться тільки через таблиці замін. Операції XOR (додавання за модулем 2) обов'язково поєднуються з операціями ADD (додавання за модулем 2^{32}).

Таблиці замін спочатку запозичені з алгоритму MARS і містять 512 32-розрядних елементів, проте були жорстко проаналізовано на предмет посилення.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

Ключове розклад було спроектовано з урахуванням вимог:

- Простота
- Використовується та ж процедура, що і при шифруванні та розшифрування
- Установка ключа займає менше часу, ніж зашифрування
- Виключення еквівалентних ключів
- Виключення слабких ключів

Алгоритм, згідно із заявою авторів, стійкий до лінійного і диференціального аналізу.

К6ПЗ-2026

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі система Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи: Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші; Розділу виведення результату роботи системи.

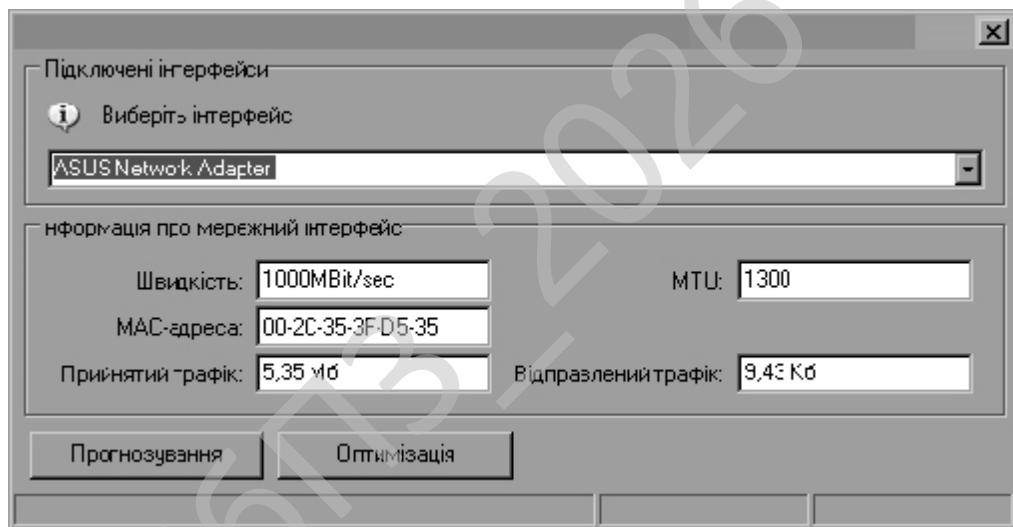


Рисунок 5.1 – Головне вікно ПЗ

Мережеві системи запобігання вторгнень (IPS) можуть стати як ефективним інструментом для людей, що займаються безпекою, так і дорогим обладнанням, що порохиться без справи. Щоб IPS-система не стала розчаруванням, вона повинна як мінімум відповідати наступним вимогам, які необхідно враховувати при її виборі.

IDS / IPS-системи найчастіше підключають там, де є критичні ресурси. Але крім того, що необхідно блокувати атаки на ці ресурси, варто вести постійний

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

контроль за ними, а саме: знати, які із цих ресурсів уразливі, як змінюється їхнє поведіння в мережі. Тому до IDS / IPS-систем необхідно додати додаткового функціонала, що дозволяє їм захищати необхідні ресурси більш надійно, знижуючи при цьому вартість володіння. Отже, захист може здійснюватися в три фази – IPS, Adaptive IPS, Enterprise Threat Management. Функціонал кожної наступної фази містить у собі всі функції з попередньої фази й наращується більш новими.

Розроблена програма має дуже простий і 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.

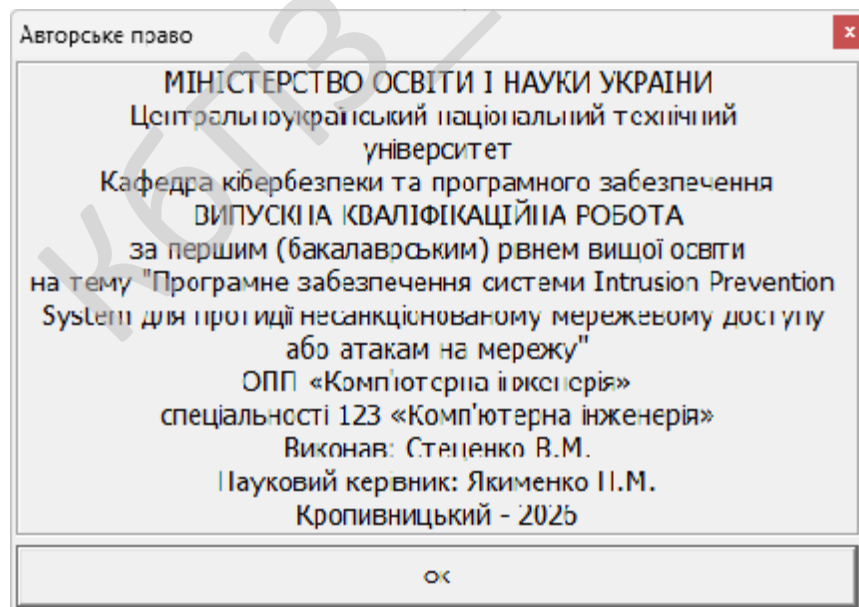


Рисунок 5.2 – Авторське право

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

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.
- Оновлення.
- 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

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

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.

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:

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

– 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 – Freeware. Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу.

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу.
- Досліджена система Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу.
- На основі отриманих результатів досліджень створена програмна реалізація системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу.

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

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

системи Intrusion Prevention System для протидії несанкціонованому мережевому доступу або атакам на мережу. 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 Sinople.

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.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Alyssa Miller. Cybersecurity Career Guide. Manning Publications. 2022. 368 p.
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. Kuznetsov O., Frontoni E., Kuznetsova Y., Chevardin V., Smirnov O. «Architectural foundations for adaptive security in edge computing systems». *Cybersecurity Defensive Walls in Edge Computing*, 2025. pp. 21-61.
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.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

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., Коноплицька-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 ІР-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., Prysiashnyy, 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

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

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., Yenchov, 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., Chubaievskiy, V., Khorolska, K., Bebesko, B. «Selection of a Rational Composition of Information Protection Means Using a Genetic Algorithm». In: *Rajakumar, G., Du, K.L., Vuppapapati, 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». *II Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи розвитку інформаційних технологій (ІІШРІТ-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.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

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

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

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.

					ВКРБ-123.26.0038.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

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.

54. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

55. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

56. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». *International Journal of Computer Network and Information Security (IJCNIS)*. Vol. 12, No. 3, 2020. PP.33-43.

57. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 646-660.

58. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

59. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.