Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: **Методи та засоби аналізу мережевого трафіку для виявлення комп'ютерних атак**

Виконав(ла): студент(ка) 6 курсу, групи СТм-61
спеціальності 126 «Інформаційні системи та технології»
(шифр і назва спеціальності)

(підпис) Сташук В.М. (прізвище та ініціали)

Керівник (підпис) Матійчук Л.П. (прізвище та ініціали)

Нормоконтроль (підпис) Мацюк О. В. (прізвище та ініціали)

Завідувач кафедри (підпис) Боднарчук І. О. (прізвище та ініціали)

Рецензент (підпис) Жаровський Р.О. (прізвище та ініціали)

Тернопіль
2021

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет Комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра Комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
« » 20___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)

за спеціальністю 126 «Інформаційні системи та технології» група СТм-61
(шифр і назва спеціальності)

студенту Сташуку Вадиму Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Методи та засоби аналізу мережевого трафіку для виявлення комп'ютерних атак

Керівник роботи Матійчук Любомир Павлович, к.е.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «*28*» *жовтня* 2021 року № *4/7-911*.
2. Термін подання студентом завершеної роботи
3. Вихідні дані до роботи Наукові публікації, електронні ресурси, підручники , посібники з тематики дослідження

4. Зміст роботи (перелік питань, які потрібно розробити) Вступ. 1. Стан та перспективи розвитку засобів виявлення комп'ютерних атак. 2. Нейромережева система аналізу мережевого трафіку для виявлення комп'ютерних атак. 3. Реалізація нейромережевої системи аналізу мережевого трафіку для виявлення комп'ютерних атак. 4. Охорона праці та безпека в надзвичайних ситуаціях. Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Актуальність. 2. Типовий приклад атаки на мережу. 3. Об'єкти атак. 4.Рейтинг виявлених зразків шкідливих программ. 5. Схема виявлення атак. 6.Основні елементи локальної архітектури систем виявлення вторгнень. 7. Нейромережева штучна імунна системи для виявлення комп'ютерних атак. 8.Структура і алгоритм навчання нейромережевого імунного детектора. 9.Програмна реалізація. 10.Експерименти. 11. Висновки. 12. Завершальний слайд

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Приймак М.В., проф.		
Безпека в надзвичайних ситуаціях	Клепчик В.М., ст..викл.		

7. Дата видачі завдання 27 вересня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	27.09.2020-29.09.2020	*Виконано*
2.	Підбір наукових джерел про методи та засоби аналізу	30.09.2020-03.10.2020	*Виконано*
	мережевого трафіку для виявлення комп'ютерних атак		
3.	Переклад та опрацювання наукових джерел про	04.10.2020-10.10.2020	*Виконано*
	методи та засоби аналізу мережевого трафіку для		
	виявлення комп'ютерних атак		
4.	Виконання дослідження щодо аналізу мережевого	11.10.2020-17.10.2021	*Виконано*
	трафіку для виявлення комп'ютерних атак		
5.	Оформлення розділу «Стан та перспективи розвитку	18.10.2021-24.10.2021	*Виконано*
	засобів виявлення комп'ютерних атак»		
6.	Оформлення розділу «Нейромережева система аналізу	25.10.2021-31.10.2021	*Виконано*
	мережевого трафіку для виявлення комп'ютерних		
	атак»		
7.	Оформлення розділу «Реалізація нейромережевої	01.11.2021-07.11.2021	*Виконано*
	системи аналізу мережевого трафіку для виявлення		
	комп'ютерних атак»		
8.	Виконання завдання до підрозділу «Охорона праці»	08.11.2021-11.11.2021	*Виконано*
9.	Виконання завдання до підрозділу «Безпека в	12.11.201-14.11.2021	*Виконано*
	надзвичайних ситуаціях»		
10.	Оформлення кваліфікаційної роботи	15.11.201-24.11.2021	*Виконано*
11.	Нормоконтроль	25.11.2021-28.11.2021	*Виконано*
12.	Перевірка на плагіат	01.12.2021	*Виконано*
13.	Попередній захист кваліфікаційної роботи	07.12.2021	*Виконано*
14.	Захист кваліфікаційної роботи	20.12.2021	

Студент _____ Сташук В. М.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Матійчук Л.П.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Методи та засоби аналізу мережевого трафіку для виявлення комп'ютерних атак // Кваліфікаційна робота освітнього рівня «Магістр» // Сташук Вадим Миколайович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СТм-61 // Тернопіль, 2021 // с. 65, рис. – 5, табл. 8, бібліогр. – 53, додат. – 6.

Ключові слова: системи виявлення атак IDS, виявлення вторгнень, нейромережева система.

У кваліфікаційній роботі розглянуто методи та засоби аналізу мережевого трафіку для виявлення комп'ютерних атак. Розроблено нейромережеву систему аналізу мережевого трафіку для виявлення комп'ютерних атак. Подана архітектура системи аналізу мережевого трафіку для виявлення комп'ютерних атак. Запропонована структура і алгоритм навчання нейромережевого імунного детектора.

На основі проведеного дослідження реалізовано нейромережева система аналізу мережевого трафіку для виявлення комп'ютерних атак.

ANNOTATION

Methods and means of network traffic analysis for cyber-attacks detection. // Qualification thesis Master Degree // Stashuk Vadym Mykolayovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science of Yuter Sciences, STm-61 group // Ternopil, 2021 // p. 65, Fig. -5, table. -8, bibliogr. - 53, add. - 6.

Key words: IDS attack detection systems, intrusion detection, neural network system.

The qualification work discusses methods and tools for analyzing network traffic to detect computer attacks. A neural network system for analyzing network traffic to detect computer attacks has been developed. The architecture of the network traffic analysis system for detecting computer attacks is presented. The structure and algorithm of training of the neural network immune detector are offered.

Based on the study, a neural network system for analyzing network traffic to detect computer attacks was implemented.

ЗМІСТ

ВСТУП

Актуальність теми. З розвитком інформаційних технологій і впровадженням їх в повсякденне життя, суспільство зіткнулося з особливим видом неправомірної діяльності – кіберзлочинністю. На даний час гостро стоїть проблема захисту персональних комп'ютерів користувачів і комп'ютерних мереж організацій від мережевих вторгнень і атак хакерів. Дослідження показують, що комп'ютер, підключений до мережі Інтернет може бути атакований кожні 39 секунд. Сьогодні застосовуються різноманітні методи і засоби для захисту від мережевих атак, проте всі вони мають ряд істотних недоліків, і не здатні повною мірою захистити користувача від вторгнень. У зв'язку з цим, для надійного захисту комп'ютерних систем від мережевих вторгнень необхідно розробляти принципово нові методи захисту.

Комп'ютерні мережі за декілька останніх десятиліть з чисто технічного рішення перетворилися на глобальне явище, розвиток якого впливає на більшість сфер економічної діяльності. Забезпечення працездатності мережі і інформаційних систем, що функціонують в ній, залежить не тільки від надійності апаратури, але і, часто, від здатності мережі протистояти цілеспрямованим діям, які направлені на порушення її роботи.

Створення інформаційних систем, стійких до шкідливих дій і комп'ютерних атак, позв'язане з істотними витратами як часу, так і матеріальних ресурсів. Створення ефективних засобів захисту інформаційних систем стикається також з браком обчислювальної потужності.

Таким чином, зростання обчислювальної потужності вузлів мережі відстає від зростання об'ємів інформації, що передається по мережі, що з кожним роком посилює вимоги до обчислювальної складності алгоритмів систем захисту інформації.

Одним з перспективних напрямів забезпечення безпеки комп'ютерних систем є використання методів штучного інтелекту (ШІ), таких як нейронні

мережі, штучні імунні системи, еволюційне програмування і т.д., які вже довели свою ефективність у вирішенні складних задач розпізнавання, класифікації, управління і виявлення. На їх основі вже існують прототипи систем захисту комп'ютерної інформації. Застосування методів ШІ дозволить створити ефективну адаптивну самонавчальну систему виявлення мережевих вторгнень і підвищити рівень захисту комп'ютерних систем від атак хакерів.

Мета і задачі дослідження. Метою роботи є розробка системи аналізу мережевого трафіку для виявлення комп’ютерних атак на основі нейромережевих технологій.

Для вирішення поставленої мети вирішуються наступні завдання:

1) провести аналіз існуючих підходів до рішення поставлених задач;

2) розглянути стандартні методи їхнього розв’язання та запропонувати альтернативні варіанти рішень із використанням новітніх технологій, зокрема методів штучних нейронних мереж;

3) розробити алгоритми системи аналізу мережевого трафіку для виявлення комп’ютерних атак;

4) теоретично обґрунтувати та здійснити детальний опис практично проведених експериментальних досліджень;

5) розробити структуру системи та реалізувати програмні модулі;

6) отримати результати тестування програмних модулів;

7) зробити висновки про адекватність функціонування та відповідність поставленим цілям.

Об’єкт дослідження – процеси аналізу мережевого трафіку для виявлення комп’ютерних атак.

Предмет дослідження – алгоритм навчання та алгоритм функціонування нейромережевого детектора системи аналізу мережевого трафіку для виявлення комп’ютерних атак.

Методи дослідження. Дослідження проводилися на базі комплексного системного аналізу, теорії штучних нейронних мереж.

Наукова новизна одержаних результатів. Розроблений алгоритм навчання нейромережевого детектора системи аналізу мережевого трафіку для

виявлення комп'ютерних атак. Вдосконалений алгоритм функціонування нейромережевих детекторів, що дозволяє ефективно ідентифікувати комп'ютерні атаки.

Практичне значення отриманих результатів. Розроблена нейромережева система аналізу мережевого трафіку для виявлення комп'ютерних атак, яка може бути використана для побудови як нових систем захисту комп'ютерів від атак, так і в додаток до вже існуючих методів.

Апробація результатів кваліфікаційної роботи. Основні положення та результати проведених досліджень доповідались та обговорювались на IX науково-технічній конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя (м. Тернопіль, 2021 р.). Публікації. Основні результати кваліфікаційної роботи опубліковані у двох працях науково-технічної конференції (Див. додаток А).

1 СТАН І ПЕРСПЕКТИВИ РОЗВИТКУ ЗАСОБІВ ВИЯВЛЕННЯ КОМП'ЮТЕРНИХ АТАК

1.1. Системи виявлення атак IDS

Системи виявлення вторгнень (IDS - Intrusion Detection Systems) - один з найважливіших елементів систем інформаційної безпеки мереж. Ріст в останні роки числа проблем, пов'язаних з комп'ютерною безпекою, привів до того, що системи виявлення вторгнення дуже швидко стали ключовим компонентом будьякої стратегії мережного захисту. За останні кілька років їхня популярність значно зросла, оскільки продавці засобів захисту значно поліпшили якість і сумісність своїх програм [3].

Системами виявлення атак (СВА) називають безліч різних програмних і апаратних засобів, поєднуваних одною загальною властивістю - вони займаються аналізом використання довірених їм ресурсів і, у випадку виявлення яких-небудь підозрілих або просто нетипових подій, здатні вживати деякі самостійні дії по виявленню, ідентифікації й усуненню їхніх причин [9].

Але системи виявлення атак це лише один з інструментів захисного арсеналу й він не повинен розглядатися як заміна для кожного з інших захисних механізмів. Захист інформації найбільш ефективний, коли в інтрамережі підтримується багаторівневий захист. Вона складається з наступних компонентів [7]:

Політика безпеки інтрамережі організації;

Система захисту хостів у мережі;

Мережний аудит;

Захист на основі маршрутизаторів;

Міжмережеві екрани;

Системи виявлення вторгнень;

План реагування на виявлені атаки.

Отже для повного захисту цілісності мережі необхідна реалізація всіх перерахованих вище компонентів захисту. І використання багаторівневого

захисту є найбільш ефективним методом запобігання несанкціонованого використання комп'ютерних систем і мережних сервісів. Таким чином, система виявлення вторгнень - це один з компонентів забезпечення безпеки мережі в багаторівневій стратегії її захисту.

Класифікація IDS розглянемо на рисунку 117.

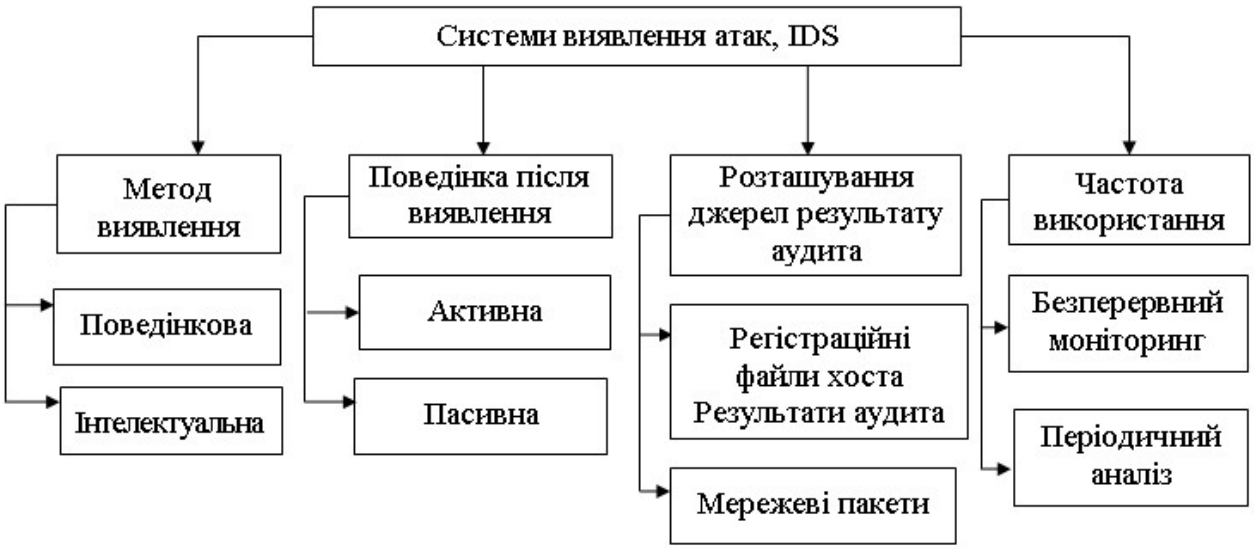


Рисунок 1.1. Характеристики систем виявлення атак [7]

Метод виявлення описує характеристики аналізатора. Коли IDS використовує інформацію про нормальне поводження контрольованої системи, вона називається поведінковою. Коли IDS працює з інформацією про атаки, вона називається інтелектуальної.

Поводження після виявлення вказує на реакцію IDS на атаки. Реакція може бути активна - IDS уживає коригувальні (усуває лазівки) або дійсно активні (закриває доступ для можливих порушників, роблячи недоступними сервіси) дії.

Якщо IDS тільки видає попередження, її називають пасивною.

Розташування джерел результату аудита підрозділяє IDS залежно від виду вихідної інформації, що вони аналізують. Вхідними даними для них можуть бути результати аудита, системні реєстраційні файли або мережні пакети.

Частота використання відбиває або безперервний моніторинг контрольованої системи з боку IDS, або відповідним періодичним запускам IDS для проведення аналізу.

Класифікувати IDS можна також по наступних параметрах (рисунок 1.2) [7].

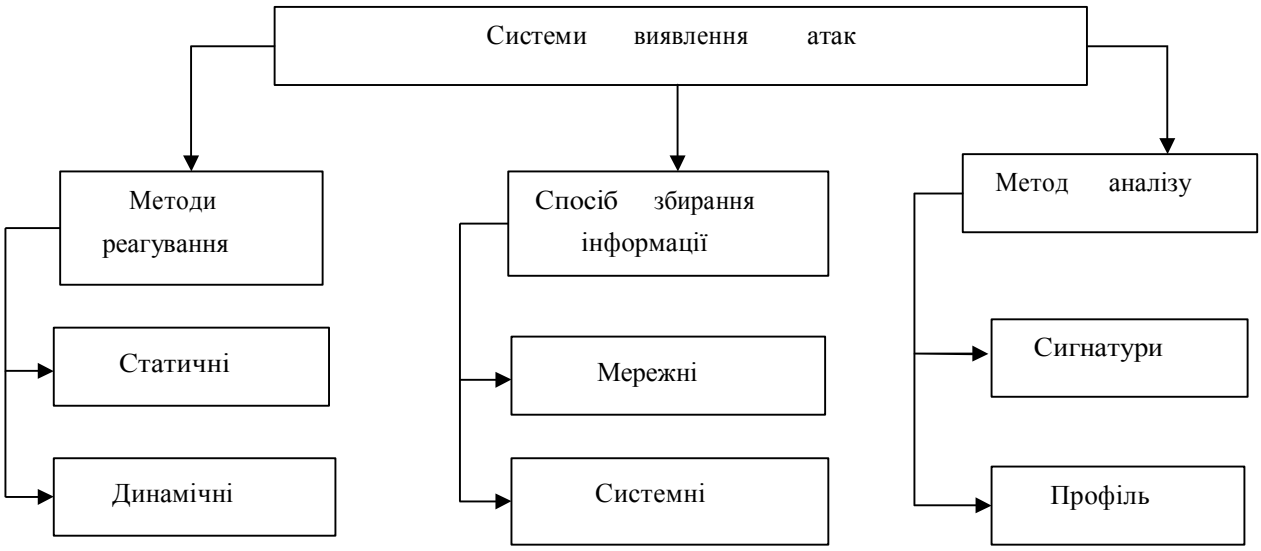


Рисунок 1.2. Класифікація систем виявлення атак [7]

По способах реагування розрізняють статичні й динамічні IDS. Статичні засоби роблять «знімки» (snapshot) середовища й здійснюють їхній аналіз, розшукуючи уразливе програмне забезпечення, помилки в конфігураціях. Статичні IDS перевіряють версії працюючих у системі додатків на наявність відомих слабких паролів, перевіряють уміст спеціальних файлів у директоріях користувачів або перевіряють конфігурацію відкритих мережних сервісів. Статичні IDS виявляють сліди вторгнення. Динамічні IDS здійснюють моніторинг у реальному часі всіх дій, що відбуваються в системі, переглядаючи файли аудита або мережні пакети, передані за певний проміжок часу. Динамічні IDS реалізують аналіз у реальному часі й дозволяють постійно стежити за безпекою системи.

По способі збору інформації розрізняють мережні й системні IDS. Мережні (NIDS) контролюють пакети в мережному оточенні й виявляють спроби зловмисника проникнути усередину захисту системи, щоб, або

реалізувати атаку «відмова в обслуговуванні». Ці IDS працюють із мережними потоками даних. Типовий приклад NIDS - система, що контролює велике число TPC-запитів на з'єднання (SYN) з багатьма портами на обраному комп'ютері, виявляючи, таким чином, що хтось намагається здійснити сканування TCP-портів. Мережна IDS може запускатися або на окремому комп'ютері, що контролює свій власний трафік, або на виділеному комп'ютері, що прозоро переглядає весь трафік у мережі (концентратор, маршрутизатор). Мережні IDS контролюють багато комп'ютерів, тоді як інші IDS контролюють тільки один. IDS, які встановлюються на хосте й виявляють зловмисні дії на ньому називаються хостовими або системними IDS. Прикладами хостових IDS можуть бути системи контролю цілісності файлів (СКЦФ), які перевіряють системні файли з метою визначення, коли в них були внесені зміни. Монітори реєстраційних файлів (Log-file monitors, LFM), контролюють реєстраційні файли, створювані мережними сервісами й службами. Облудні системи, що працюють із псевдосервисами, ціль яким полягає у відтворенні добре відомих уязвимостей для обману зловмисників.

По методах аналізу IDS ділять на дві групи [8]: IDS, які порівнюють інформацію із бази сигнатур атак і IDS, що контролюють частоту подій або виявлення статистичних аномалій. Аналіз сигнатур був першим методом, застосованим для виявлення вторгнень. Він базується на простому понятті збігу послідовності зі зразком. У вхідному пакеті проглядається байт за байтом і рівняється із сигнатурою (підписом) - характерним рядком програми, що вказує на характеристику шкідливого трафіка. Такий підпис може містити ключову фразу або команду, що пов'язана з нападом. Якщо збіг знайдений, оголошується тривога.

Другий метод аналізу складається в розгляді строго форматованих даних трафіка мережі, відомих як протоколи. Кожний пакет супроводжується різними протоколами. Автори IDS, знаючи це, впровадили інструменти, які розвертають і оглядають ці протоколи, відповідно до стандартів. Кожний протокол має кілька полів з очікуваними або нормальними значеннями. Якщо що-небудь порушує ці стандарти, то ймовірно зловмисність. IDS переглядає кожне поле всіх

протоколів вхідних пакетів: IP, TCP, і UDP. Якщо є порушення протоколу, наприклад, якщо він містить несподіване значення в одному з полів, оголошується тривога

Системи аналізу сигнатури мають кілька важливих сильних сторін. Поперше, вони дуже швидкі, тому що повний аналіз пакета - відносно важке завдання. Правила легко написати, зрозуміти й настроїти. Крім того, є просто фантастична підтримка комп'ютерного співтовариства у швидкому виробництві сигнатур для нових небезпек. Ці системи перевершують всі інші при вилові хакерів на первинному етапі: прості атаки мають звичку використовувати якісь попередні дії, які легко розпізнати. Нарешті, аналіз, заснований на сигнатурі, точно й швидко повідомляє, що в системі все нормально (якщо це дійсно так), оскільки повинні відбутися якісь особливі події для оголошення тривоги.

З іншої сторони IDS, що ґрунтується тільки на аналізі сигнатур, має певні слабості. Будучи спочатку дуже швидкої, згодом швидкість її роботи буде вповільнюватися, оскільки зростає число сигнатур, що перевіряються. Це - істотна проблема, оскільки число сигнатур, що перевіряються, може рости дуже швидко. Фактично, кожна нова атака або дія, придумана атакуючої, збільшує список сигнатур, що перевіряються. Не допоможуть навіть ефективні методи роботи з даними й пакетами: величезна кількість злегка змінених атак можуть прослизнути через таку систему.

Є й інша сторона проблеми: тому що система працює, порівнюючи список наявних сигнатур з даними пакета, така IDS може виявити тільки вже відомі атаки, сигнатури яких є занесені у базу сигнатур.

Але необхідно відзначити, що відповідно до статистики 80% атак відбувається по давно відомих сценаріях. Наявність у системі виявлення сигнатур відомих атак дає високий відсоток виявлення атак.

У випадку аналізу протоколів теж є свої переваги й недоліки. Через предпроцесів, що вимагають ретельної експертизи протоколів, аналіз протоколу може бути досить повільним. Крім того, правила перевірки для системи протоколу важко написати й зрозуміти. Можна навіть сказати, що в цьому

випадку доводиться уповати на сумлінність виробника програми, тому що правила відносно складні й важкі для самостійного настроювання.

На перший погляд, IDS на основі аналізу протоколу працюють повільніше, ніж системи на основі сигнатури, вони, більше «ґрунтовні» у змісті масштабності й результатів. Крім того, ці системи шукають «генетичні порушення» і часто можуть виявляти нові атаки. Розглянемо архітектуру IDS (рисунок 1.3).

У систем виявлення вторгнень доцільно розрізняти локальну й глобальну архітектуру. У рамках локальної архітектури реалізуються елементарні складові, які потім можуть бути об'єднані для обслуговування корпоративних систем [2].

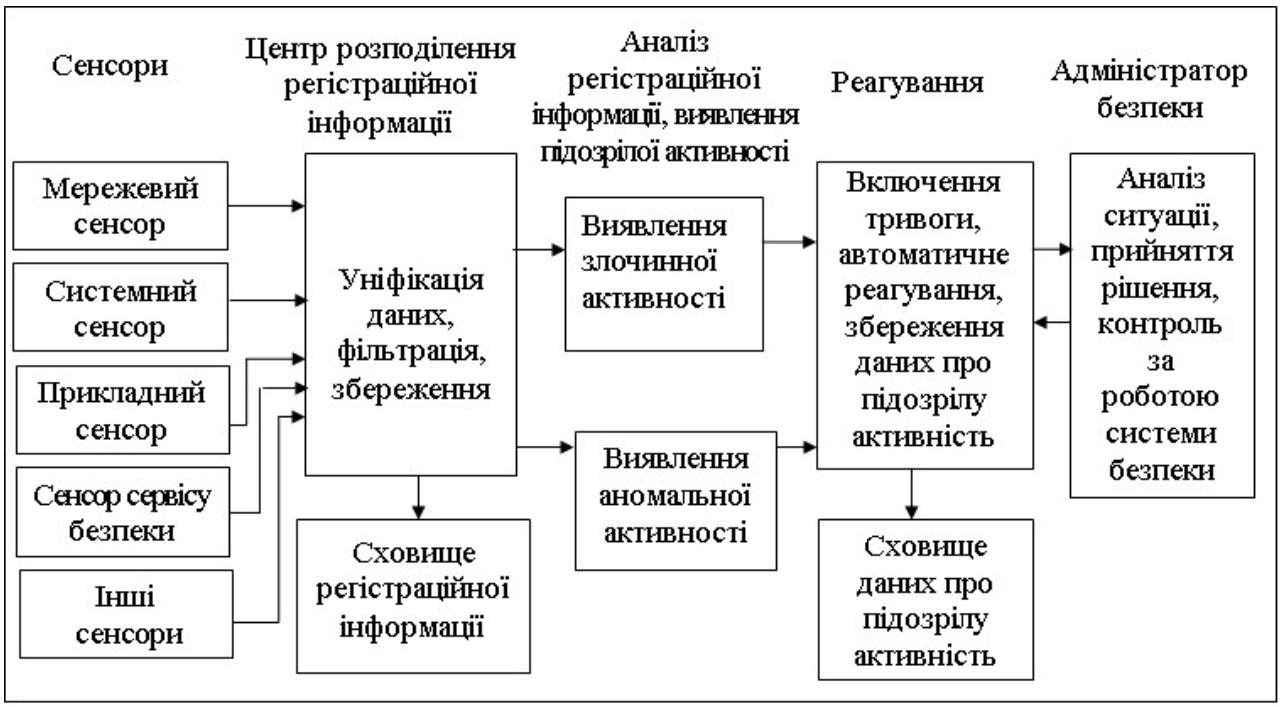


Рисунок 1.3. Основні елементи локальної архітектури систем виявлення вторгнень [2]

Основні елементи локальної архітектури й зв'язку між ними показані на малюнку 3. Первинний збір даних здійснюють агенти, які називаються ще сенсорами. Реєстраційна інформація може витягати із системних або прикладних журналів (технічно нескладно одержувати її й прямо від ядра ОС), або добуватися з мережі за допомогою відповідних механізмів активного мережного встаткування або шляхом перехоплення пакетів за допомогою встановленої в режим моніторингу мережної карти.

На рівні агентів (сенсорів) може виконуватися фільтрація даних з метою зменшення їхнього обсягу. Це жадає від агентів деякого інтелекту, але зате розвантажує інші компоненти системи.

Агенти передають інформацію в центр розподілу, що приводить її до єдиного формату, можливо, здійснює подальшу фільтрацію, зберігає в базі даних і направляє для аналізу статистичному й експертному компонентам. Один центр розподілу може обслуговувати кілька сенсорів.

Змістовний активний аудит починається зі статистичного й експертного компонентів. Якщо в процесі статистичного або експертного аналізу виявляється підозріла активність, то відповідне повідомлення направляється решателю, він визначає, чи є тривога виправданою і вибирає спосіб реагування.

Система виявлення вторгнень повинна вміти виразно пояснити, чому вона зняла тривогу, наскільки серйозні ситуація і які рекомендуються дії. Якщо вибір повинен залишатися за людиною, то нехай він зводиться до декількох елементів меню, а не до рішення концептуальних проблем.

Глобальна архітектура має на увазі організацію однорангових і різнорангових зв'язків між локальними системами виявлення вторгнень (рисунок 1.4).

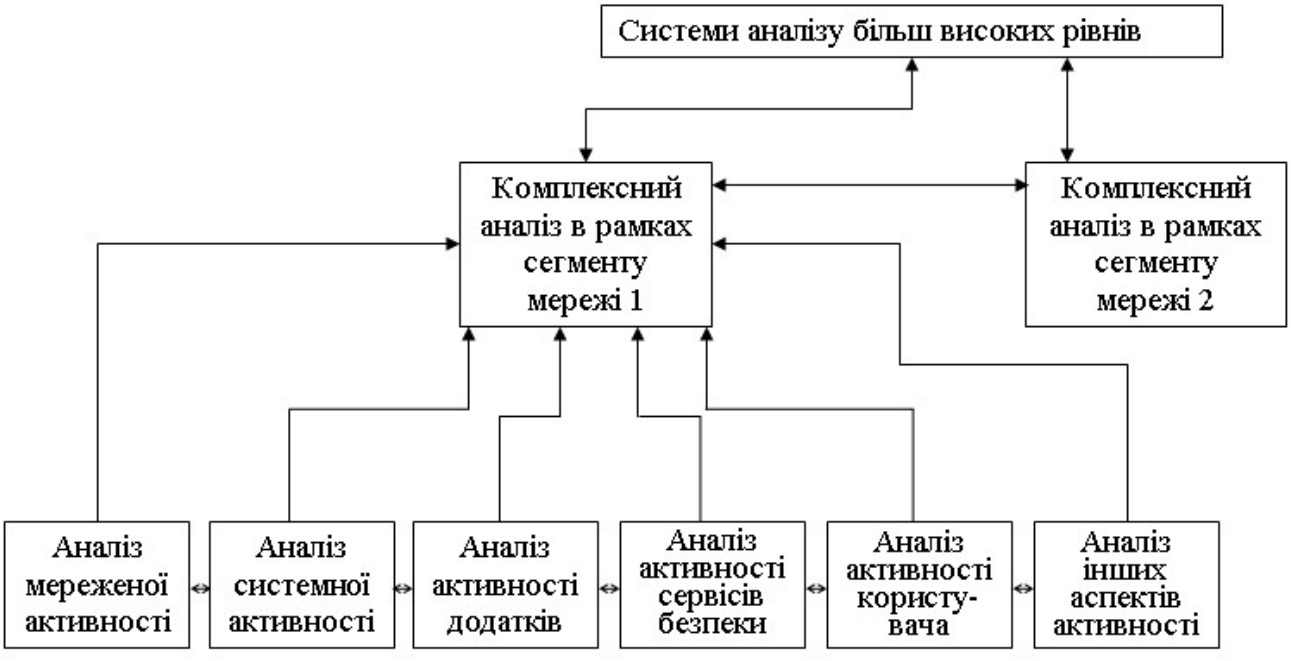


Рисунок 1.4. Глобальна архітектура систем виявлення вторгнень [2]

На одному рівні ієрархії розташовуються компоненти, що аналізують підозрілу активність із різних точок зору.

Наприклад, на хості можуть розташовуватися підсистеми аналізу поводження користувачів і додатків. Їх може доповнювати підсистема аналізу мережної активності. Коли один компонент виявляє щось підозріле, то в багатьох випадках доцільно сповістити про це сусідам для вживання заходів, або для посилення уваги до певних аспектів поводження системи.

Різнорангові зв'язки використовуються для узагальнення результатів аналізу й одержання цілісної картини про те що відбувається. Іноді в локального компонента недостатньо підстав для збудження тривоги, але "по сукупності" підозрілі ситуації можуть бути об'єднані й спільно проаналізовані, після чого поріг підозрілості виявиться перевищеним. Цілісна картина, можливо, дозволить виявити скоординовані атаки на різні ділянки інформаційної системи й оцінити збиток у масштабі організації.

Розглянемо стандарти в області систем виявлення вторгнень.

Обмін даними про підозрілу активність. Багато атак на інформаційні системи носять розподілений характер. При цьому різні засоби активного аудита бачать той самий інцидент із різних точок зору.

Поділ інформації про підозрілу активність є головним напрямком робіт створеної в рамках Тематичної групи за технологією Інтернет (Internet Engineering Task Force, IETF) Робочої групи по виявленню вторгнень (Intrusion Detection

Working Group, IDWG).

В групі IDWG має бути спеціальний формат IDMEF (Intrusion Detection Message Exchange Format) — формат обміну даними між компонентами IDS. Він використовується для передачі попереджуючих повідомлень про підозрілі події між системами виявлення атак. Даний формат повинен забезпечити сумісність між комерційними й вільно розповсюджуваними IDS і можливість їхнього спільного використання для забезпечення найвищого рівня захищеності.

IDMEF повинен підтримувати всі механізми виявлення підозрілої активності. Він повинен бути розрахований на IPv6, містити все необхідне для

інтернаціоналізації, підтримувати фільтрацію повідомлень компонентом реагування, їхню надійну доставку (у тому числі через межсетевой екран без внесення в конфігурацію останнього змін, здатних послабити периметр безпеки).

Зрозуміло, формат IDMEF повинен підтримувати взаємну аутентифікацію сторін, що спілкуються, невідмовність від факту передачі, а також цілісність і конфіденційність потоку повідомлень.

У повідомленнях формату IDMEF повинні втримуватися дата й час підозрілих подій і, якщо можливо, дата й час атаки.

Якщо аналізатор сам вжив відповідних заходів, в IDMEF-повідомленнях повинна бути інформація про це. Якщо аналізатор може оцінити наслідки зафіксованої атаки, він також зобов'язаний сповістити про це.

Формат IDMEF повинен підтримувати інформацію про виробника системи виявлення атак, з генеровані повідомлення, а також розширення, специфічні для конкретної системи.

Передбачається, що буде затверджений список стандартних атак і методів їхнього проведення. Якщо аналізатор може ідентифікувати атаку й використовуваний метод, він повинен включити відповідну інформацію в IDMEFповідомлення. Якщо атака є нестандартної, її ім'я може бути специфічним для виробника системи активного аудита.

Загальний каркас систем виявлення вторгнень (Common Intrusion Detection Framework, CIDF) розробляється групою дослідницьких організацій, фінансованих агентством DARPA і працюючих в області виявлення підозрілої активності.

У рамках CIDF розроблений язик опису підозрілої активності й спосіб кодування інформації про підозрілі події. Язик пристосований для опису, принаймні, трьох видів повідомлень:

"сирої" інформації про події (наприклад, записів реєстраційного журналу або мережних пакетів);

результатів аналізу (таких як виявлені аномалії або атаки);

рекомендованих реакцій (перервати яку-небудь активність або змінити конфігурацію захисних засобів).

Крім того, мовою можуть бути описані наступні сутності:

- зв'язку між подіями (наприклад, причинно-наслідкові);
- ролі об'єктів у подіях (наприклад, об'єкт ініціював подію);
- властивості об'єктів;
- зв'язку між об'єктами.[2]

1.2. Постановка задачі дослідження

Технології виявлення атак розвиваються швидше блискавки. Напрямок розвитку - здешевлення інфраструктури. Ніхто не хоче багато витрачати на захист. Змагання в технологіях виявлення атак буде вигране тими, хто запропонує замовникам найбільш дешеве рішення.

Буде спостерігатися розвиток трьох областей: розгортання, технологічність і якість виявлення атак. В області розгортання ми будемо бачити розширення числа місць виявлення атак: на мережному рівні (на міжмережних екранах, на комутаторах, на маршрутизаторах), на рівні операційної системи (на серверах, на робочих станціях) і на прикладному рівні (у СУБД або на сервері SAP, наприклад). Для технологічності, ми будемо бачити, що системи стануть більше простими у функціонуванні й більше “приборо-подібними", щоб вмонтувати їх у мережну інфраструктуру без внесення в останню серйозних змін. В області якості виявлення атак ми побачимо, що логіка розпізнавання атак почне включати моделі побудови поведінкових профілів і відхилень від цього профілю. Буде набагато більше "інтелекту" у визначенні того, що є неправильним використанням ресурсу або атакою. Загальне число атак з ростом мережних технологій неминуче збільшиться.

Системи виявлення атак досить вчасно виявляють відомі атаки. Не варто чекати від таких систем виявлення невідомих на сьогоднішній день атак. Проблема виявлення чогось, невідомого до справжнього моменту, є дуже важкою й межує із областю штучного інтелекту й експертних систем (однак у цих областях уже досягнуті чималі успіхи; особливо з розвитком теорій нейронних мереж і нечіткої логіки). Також не слід очікувати, що системи виявлення атак здатні реагувати на атаки шляхом нападу. Це дуже небезпечна можливість, тому що вона означає, що фіктивна тривога або помилкове спрацьовування може викликати реакцію, що забороняє ту або іншу послугу або мережу, що блокує доступ в мережу. Проблема із системами виявлення атак полягає в тому, що, багато з людей, думають, що системи виявлення атак діють подібно інтелектуальному "ICE" (щось начебто штучного розуму, що забезпечує захист інформаційної системи) і можуть захистити мережі набагато ефективніше, ніж це може бути насправді.

В майбутньому можна очікувати від систем виявлення атак ідентифікації в практично реальному режимі часу будь-яких спроб використання відомих слабких місць або несанкціонованого дослідження вашої внутрішньої мережі. Вони також повинні стежити за спробами перевантаження критичних ресурсів. Поряд із цим, вони повинні видавати звукові попередження про атаку, виконувати певні дії й створювати журнал реєстрації подій для наступного аналізу.

Наступним великий крок - виявлення розподілених атак шляхом прийому даних від безлічі датчиків, рознесених по мережі, наприклад, підприємства, і групування цих даних у єдине зображення, що відбиває загальну картину нападів на мережу (перші такі рішення вже стали з'являтися, наприклад, системи RealSecure або NetRanger). Єдиний спосіб робити це сьогодні - вручну, і можна припустити, що необхідність у людському контролі буде ще потрібна протягом деякого часу.

В даний час відбувається безперервне зростання кількості атак і зловживань у сфері високих технологій. Тому забезпеченню безпеки комп'ютерних систем приділяється все більше і більше уваги.

Традиційні методи виявлення атак, такі, як сигнатурний метод або метод виявлення аномалій, не дозволяють досягти оптимальних характеристик виявлення атак.

Основним недоліком існуючих систем, заснованих на традиційних методах є їх нездатність виявляти нові або невідомі атаки, які характеризуються відсутністю про них записів в системі. Сучасні системи виявлення вторгнень також погано пристосовані до роботи в реальному режимі часу, що знижує їх ефективність використання в системах захисту.

Одним з перспективних напрямів забезпечення безпеки комп'ютерних систем є використання технологій штучних нейронних мереж.

Отже, метою даної роботи є розробка системи аналізу мережевого трафіку для виявлення комп’ютерних атак на основі штучних нейронних мереж.

Висновки до першого розділу

В даному розділі введено і означено поняття IDS, як один із семи компонентів багаторівневого захисту. Наведено класифікацію IDS, що дало змогу оцінити основні напрямки спрямування виявлення мережевих атак. Наведена архітектура IDS. Проаналізовано стандарти в області систем виявлення атак.

2 НЕЙРОМЕРЕЖЕВА СИСТЕМА АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ ДЛЯ ВИЯВЛЕННЯ КОМП'ЮТЕРНИХ АТАК

2.1. Архітектура системи аналізу мережевого трафіку для виявлення комп'ютерних атак

Розглянемо основні принципи побудови нейромережевої штучної імунної системи для виявлення комп'ютерних атак. Схема узагальненого алгоритму функціонування штучної імунної системи представлено в додатку А.

Нейромережевий модуль штучної імунної системи для виявлення і класифікації комп'ютерних атак зображений в додатку Б і складається з популяції нейромережевих імунних детекторів, які застосовуються для виявлення комп'ютерних атак і нейромережевого класифікатора, призначеного для класифікації виявлених комп'ютерних атак.

Розглянемо процес побудови штучної імунної системи на основі нейронних мереж. Спочатку генерується початкова популяція імунних детекторів, кожний з яких є штучною нейронною мережею. Представимо нейромережевий імунний детектор у вигляді чорного ящика, який має *n*-входів і два виходи (рисунок 2.1).

Рисунок 2.1. Нейромережевий імунний детектор

Вихідні значення детектора формуються після подачі всіх образів на нього відповідно до наступних виразів

$$Z_1 = \begin{cases} 1, \text{якщо не атака} \\ 0, \text{інакше} \end{cases} \quad (2.1)$$

$$Z_2 = \begin{cases} 1, \text{якщо атака} \\ 0, \text{інакше} \end{cases} \quad (2.2)$$

Набір з чистих записів і атак утворюють навчальну вибірку для нейромережевих детекторів. Присутність чистих записів і атак дозволяє навченим імунним детекторам знаходити різницю між атаками і не атаками. Бажано також мати представників всіх типів комп'ютерних атак.

Нейронна мережа навчається шляхом навчання з вчителем, тобто ми вказуємо штучній нейронній мережі, де дані з чистих записів, а де з атаками.

Нехай T – множина чистих записів, а F – множина атак. З них випадковим чином формується множина вхідних образів для навчання i-го детектора.

$$X_i = \begin{bmatrix} X_i^1 \\ X_i^2 \\ \dots \\ X_i^L \end{bmatrix} = \begin{bmatrix} X_i^1 X_{i2}^1 \dots X_{in}^1 \\ X_i^2 X_{i2}^2 \dots X_{in}^2 \\ \dots \\ X_i^L X_{i2}^L \dots X_{in}^L \end{bmatrix} \quad (2.3)$$

Відповідно**,** множина еталонних образів

$$L_i = \begin{bmatrix} L_i^1 \\ L_i^2 \\ \dots \\ L_i^L \end{bmatrix} = \begin{bmatrix} L_i^1 L_{i2}^1 \dots L_{in}^1 \\ L_i^2 L_{i2}^2 \dots L_{in}^2 \\ \dots \\ L_i^L L_{i2}^L \dots L_{in}^L \end{bmatrix} \quad (2.4)$$

Тут L – розмірність навчальної вибірки.

Еталонні вихідні значення для i-го детектора формуються таким чином:

$$l_{i1}^k = \begin{cases} 1, \text{якщо } X_{i1}^k \in T \\ 0, \text{інакше} \end{cases} \quad (2.5)$$

$$l_{i2}^k = \begin{cases} 1, \text{якщо } X_{i1}^k \in F \\ 0, \text{інакше} \end{cases} \quad (2.6)$$

Навчання кожного детектора здійснюється з метою мінімізації сумарної квадратичної помилки детектора. Сумарна квадратична помилка *i*-го детектора визначається таким чином:

$$E_i = \frac{1}{2}\sum_{k=1}^{L}\sum_{j=1}^{2}\left(Z_{ij}^k - L_{ij}^k\right)^2 \qquad (2.7)$$

де Z_{ij}^k – значення *j*-го виходу *i*-го детектора при подачі на вхід його *k*-го образу.

Величина сумарної квадратичної помилки характеризує пристосованість детектора до виявлення шкідливих записів. Чим менше її значення, тим більша пристосованість детектора. Тому величину сумарної квадратичної помилки можна використовувати для відбору кращих детекторів.

Загальний алгоритм побудови і функціонування нейромережевої імунної системи можна представити у вигляді наступної послідовності:

1. Генерація початкової популяції імунних детекторів, кожний з яких є штучною нейронною мережею з випадковими синаптичними зв'язками:

$$D = \left\{D_i, i = \overline{1,r}\right\} \qquad (2.8)$$

де D_i – *i*-й нейромережевий імунний детектор; r – загальна кількість детекторів.

2. Навчання сформованих імунних нейромережевих детекторів. Навчальна вибірка формується випадковим чином з сукупності чистих записів і з сукупності записів атак. Еталонні вихідні значення нейронної мережі формуються відповідно формулі 2.4.

3. Відбір (селекція) нейромережевих імунних детекторів на тестовій вибірці. На даній ітерації знищуються ті детектори, які стали нездатні до навчання, і детектори, в роботі яких, спостерігаються різні недоліки (наприклад, помилкові спрацьовування). Для цього кожен детектор перевіряється на тестовій вибірці. В результаті для кожного детектора визначається значення квадратичної помилки E_i (формула 2.7).

Селекція детектора проводиться таким чином:

$$D_i = \begin{cases} 0, \text{якщо } E_i \neq 0 \\ D_i, \text{інакше} \end{cases} \quad (2.9)$$

де 0 - позначає операцію знищення детектора.

4. Кожен детектор наділяється часом життя і випадковим чином вибирає запис для сканування з сукупності записів, які він не перевіряв.

5. Сканування кожним детектором вибраного запису, в результаті якого визначаються вихідні значення детекторів $Z_{i1}, Z_{i2}\ i = \overline{1,r}$.

6. Якщо *i*-й детектор не виявив атаку, тобто $Z_{i1} = 0, Z_{i2} = 0$, то він вибирає наступний запис для сканування. Якщо час життя *i*-го детектора закінчився, то він знищується і замість нього генерується новий детектор.

7. Якщо *i*-й детектор виявив атаку, тобто $Z_{i1} = 0, Z_{i2} = 1$, то подається сигнал про виявлення атаки і здійснюються операції клонування і мутації відповідного детектора. Операція мутації полягає в додатковому навчанні детекторів-клонів на виявленій атаці. В результаті створюється сукупність детекторів, налаштованих на виявлену атаку

$$D_i = \left(D_{i1}, D_{i2} \ldots D_{in}\right) \quad (2.10)$$

8. Відбір клонованих детекторів, які є найбільш пристосованими до виявлення атаки. Якщо $E_{ij} > E_{i1}$, то детектор пройшов відбір. Тут E_{ij} – сумарна квадратична помилка *j*-го клона *i*-го детектора, яка обчислюється на атаці.

9. Формування детекторів імунної пам'яті. На цій ітерації визначаються нейромережеві імунні детектори, що показали якнайкращі результати при виявленні комп'ютерних атак. Детектори імунної пам'яті знаходяться в комп'ютерній системі достатньо тривалий час, і забезпечують захист комп'ютерної системи від повторного нападу. Додаток В демонструє роботу сукупності імунних детекторів, побудованих при застосуванні нейронних мереж.

2.2. Структура і алгоритм навчання нейромережевого імунного детектора

Розглянемо структуру і навчання імунного детектора, в основі якого лежить нейронна мережа. Основним завданням нейромережевого імунного детектора є розділення простору вхідних образів на два класи: не шкідливий (чистий) клас і шкідливий (атака) клас. Пропонується використовувати нейромережевий підхід до побудови імунного детектора, який володіє здатністю до навчання і узагальнення результатів навчання при подачі на вхід детектора невідомих образів.

Розглянемо вибір класу нейронної мережі, яка лежить в основі нейромережевого імунного детектора. В процесі циркуляції НІД відбувається їх безперервна еволюція, шляхом знищення старих і формування нових детекторів. Після генерації нових детекторів відбувається процес їх навчання, трудомісткість якого пропорційна розмірності навчальної вибірки. Тому, для збільшення швидкодії нейромережевої штучної імунної системи необхідно вибрати такий клас нейронної мережі, який характеризується мінімальним розміром навчальної вибірки. Розглянемо багатошаровий персептрон, який складається з n нейронів розподільного шару, m нейронів прихованого шару і 2 нейронів вихідного шару. Загальна кількість параметрів (вагових коефіцієнтів і порогових значень), що настроюються, в такій мережі визначається таким чином:

$$V = m\cdot(n+3)+2 \tag{2.11}$$

Для хорошої класифікації розмір навчальної вибірки повинен визначатися відповідно до наступного виразу:

$$L \approx \frac{V}{\varepsilon} \tag{2.12}$$

де ε - допустима точність класифікації.

Хай $n = 128, m = 10 i = 0{,}1. Тоді L = 13120$.

Аналогічний результат можна отримає для мультирекурентних нейронних мереж.

Розглянемо аналогічну мережу зустрічного розповсюдження з ідентичною кількістю нейронних елементів в шарах. У прихованому шарі використовуватимемо нейронні елементи Кохонена. В цьому випадку немає жорстких вимог до розмірності навчальної вибірки. Достатньо, щоб розмір навчальної вибірки був наступним:

$$L \geq 2m \qquad (2.13)$$

Тому виберемо як основу нейромережевого імунного детектора нейронну мережу зустрічного розповсюдження.

В додатку Г зображена архітектура нейромережевого імунного детектора, який складається з трьох шарів нейронних елементів і арбітра. На вхід такого детектора в режимі функціонування подається масив параметрів з'єднання. Перший шар нейронних елементів є розподільним. Він розподіляє вхідні сигнали на нейронні елементи другого (прихованого) шару. Кількість нейронних елементів розподільного шару дорівнює розмірності ковзаючого вікна.

Другий шар складається з нейронів Кохонена, які використовують конкурентний принцип навчання і функціонування відповідно до правила «переможець бере все».

Третій шар складається з двох лінійних нейронних елементів, які використовують лінійну функцію активації. Арбітр здійснює процедуру остаточного рішення про приналежність сканованого запису до одного з двох класів.

Розглянемо вибір кількості нейронів в шарі Кохонена. Нейронний шар Кохонена здійснює кластеризацію вхідного простору образів, внаслідок чого утворюються кластери різних образів, кожному з яких відповідає свій нейронний $m = p + r$, елемент. Кількість нейронів шару Кохонена дорівнює m. Причому, p – кількість перших нейронів шару Кохонена, які відповідають класу чистих записів; r – кількість останніх нейронів шару Кохонена, активність яких характеризує клас атак.

При навчанні нейромережвих імунних детекторів використовується навчальна вибірка, що складається з 80% образів чистого класу, і з 20% образів шкідливого класу. Таким чином, співвідношення записів в навчальній вибірці

дорівнює чотири до одного. Дане співвідношення було отримане експериментальним шляхом і показало якнайкращі результати. У проведених експериментах генерувалося 5 сукупностей нейромережевих імунних детекторів, що складаються з 100 детекторів, які потім проходили стадію навчання і відбору. Для першої популяції нейромережевих імунних детекторів використовувалася навчальна вибірка, що складається із співвідношення образів чистого класу до образів шкідливого класу 5/1, для другої популяції – 4/1, для третьої популяції – 3/1, для четвертої – 2/1, для п'ятої популяції – 1/1. Після навчання і відбору детектори сканували тестові записи.

Якнайкращий результат показали детектори, для навчання яких використовувалася вибірка, що складається з 80% образів чистого класу і 20% образів шкідливого класу.

Нехай А – кількість фрагментів вибираних з кожного запису для навчання, і є 4 чистих записи і один шкідливий. Тоді навчальна вибірка складається з *4A* фрагментів тих, що відносяться до чистого класу і *A* фрагментів тих, що характеризують шкідливий клас. Тому співвідношення між кількістю нейронів в шарі Кохонена, які характеризують різні класи, повинно бути кратним співвідношенню чотири до одного

$$\frac{p}{r} = \frac{i}{i} \cdot \frac{4}{1} \qquad (2.14)$$

де $i = 1,2$.

Так, наприклад, при $\mathrm{i} = 1$ $\mathrm{p} = 4, \mathrm{r} = 1$, а при $i = 2$ $p = 8$, $r = 2$

Звідси витікає, що алгоритм формування навчальної вибірки складається з наступних кроків:

1) Формується сукупність чистих записів і записів з атаками.

2) З сформованої вибірки випадковим чином вибираються чотири чистих і один шкідливий запис.

3) З кожного запису випадковим чином вибираються *A* фрагментів довжиною *n*, внаслідок чого утворюється навчальна вибірка розмірністю $L = 5A$.

Для навчання нейронів шару Кохонена використовується контрольоване конкурентне навчання. При такому навчанні вагові коефіцієнти нейрона переможця модифікуються тільки тоді, коли відбувається коректна класифікація вхідного образу, тобто вхідний образ відповідає заданій множині нейронів в шарі Кохонена. Оскільки в шарі Кохонена використовується p нейронів для чистих вхідних образів і r нейронів для шкідливих вхідних образів, то коректна класифікація відбувається, якщо при подачі на вхід мережі чистого фрагмента переможцем є один з перших p нейронів шару Кохонена. Аналогічним чином коректна класифікація відбувається, якщо при подачі на вхід мережі запису з атакою, переможцем є один з r останніх нейронів шару Кохонена. У решта випадків відбувається некоректна класифікація.

Нехай P і J характеризують відповідно чистий і шкідливий запис. Тоді правило коректної класифікації можна представити у вигляді наступної імплікації:

$$P \wedge k = 1,2,...p \rightarrow T \qquad (2.15)$$

де T позначає коректну класифікацію.

При коректній класифікації вагові коефіцієнти нейрона-переможця посилюються

$$\omega_{сл}(t+1) = \omega_{ck}(t) + \gamma\left(X_c - \omega_{ck}(t)\right) \qquad (2.16)$$

а при некоректній класифікації ослабляються:

$$\omega_{сл}(t+1) = \omega_{ck}(t) - \gamma\left(X_c - \omega_{ck}(t)\right) \qquad (2.16)$$

де t – крок навчання.

Алгоритм навчання шару Кохонена складається з наступних кроків:

1. Випадкова ініціалізація вагових коефіцієнтів нейронів шару Кохонена.

2. Подається вхідний образ з навчальної вибірки на нейронну мережу і проводяться наступні обчислення:

а) обчислюється Евклідово відстань між вхідним і ваговими векторами нейронних елементів шару Кохонена

$$D_j = \left|X - \omega_j\right| = \sqrt{\left(X_1 - \omega_{1j}\right)^2 + \left(X_2 - \omega_{2j}\right)^2 + \left(X_n - \omega_{nj}\right)^2} \qquad (2.17)$$

де $j = 1...m$

b) визначається нейронний елемент переможець з номером до

$$D_k = \min_j D_j$$

c) проводиться модифікація вагових коефіцієнтів нейрона-переможця відповідно до наступних виразів:

$$\omega_{ck}(t+1) = \omega_{ck}(\ )t + \gamma(X_c - \omega_{ck}(t)), \quad (2.18)$$

якщо при подачі на вхід мережі чистого фрагмента переможцем є один з перших 8 нейронів або при подачі на вхід мережі шкідливого фрагмента переможцем є один з двох останніх нейронів мережі Кохонена. І

$$\omega_{ck}(t+1) = \omega_{ck}(\ )t - \gamma(X_c - \omega_{ck}(t)), \quad (2.19)$$

інакше.

3. Процес повторюється, починаючи з пункту 2 для всіх вхідних образів.

4. Навчання проводиться до бажаного ступеня узгодження між вхідними і ваговими векторами, тобто до тих пір, поки значення сумарної квадратичної помилки не стане рівним нулю:

$$E_i = \frac{1}{2}\sum_{k=1}^{L}\sum_{j=1}^{2}\left(Z_{ij}^k - L_{ij}^k\right)^2 \quad (2.20)$$

Основна відмінність запропонованого алгоритму від відомих полягає в тому, що при коректній класифікації кожному вхідному образу відповідає не конкретний нейрон, а один з перших *p* нейронів, або один з останніх *r* нейронів шару Кохонена.

Третій шар, що складається з двох лінійних нейронних елементів, здійснює відображення кластерів, сформованих шаром Кохонена, в два класи, які характеризують чисті і шкідливі вхідні образи. У загальному випадку вихідне значення j-го нейрона третього шару визначається таким чином:

$$Y_i = \frac{1}{2}\sum_{i=1}^{m}\omega_{ij} \cdot Y_i \quad (2.21)$$

де ω_{ij} – ваговий коефіцієнт між *i*-м нейроном шару Кохонена і *j*-м нейроном лінійного шару.

Якщо нейрон-переможець в шарі Кохонена має номер *k* то вихідне значення *j*-го нейрона третього шару дорівнює Y_i.

Для відповідного відображення вхідних образів в два класи матриця вагових коефіцієнтів третього шару повинна формуватися таким чином:

$$\omega_{kj} = \begin{cases} 1, \text{якщо } k = 1,2...p \text{ и } j = 1 \text{ або } k = p+1...r \text{ и } j = 2 \\ 0, \text{якщо } k = 1,2...p \text{ и } j = 2 \text{ або } k = p+1...r \text{ и } j = 1 \end{cases} \quad (2.22)$$

Так, наприклад, для *p=8* і *r=2*, виходить наступна матриця вагових коефіцієнтів

$$W^T = \begin{bmatrix} 1111111100 \\ 0000000011 \end{bmatrix}$$

2.3. Алгоритм функціонування нейромережевого імунного детектора

Як вже наголошувалося, нейронні елементи шару Кохонена функціонують за принципом «переможець бере все». Це означає, що вихідне значення нейронапереможця дорівнює «1», а вихідні значення решти нейронних елементів дорівнюють «0». Для визначення нейрона-переможця використовується Евклідово відстань між вхідним і ваговими векторами. Так евклідово відстань між вхідним і ваговим вектором і-го нейронного елементу визначається таким чином:

$$D_j = \left| X - \omega_j \right| = \sqrt{\left(X_1 - \omega_{1j}\right)^2 + \left(X_2 - \omega_{2j}\right)^2 + \left(X_c - \omega_{cj}\right)^2} \quad (2.23)$$

де ω_{cj} – ваговий коефіцієнт між с-м нейроном розподільного шару і і-м нейроном шару Кохонена;

$X = \left[X_1, X_2...X_n\right]$ – вхідний образ.

Нейронний елемент-переможець з номером k визначається відповідно до мінімальної евклідової відстані.

Тоді вихідна активність нейронів шару Кохонена визначається

$$Y_i = \begin{cases} 1, \text{якщо } i = \text{k} \\ 0, \text{інакше} \end{cases} \quad (2.24)$$

Як вже наголошувалося, вихідне значення j-го нейрона третього шару визначається таким чином:

$$Y_i = \omega_{kj} Y_k \quad (2.25)$$

Арбітр ухвалює остаточне рішення про те, чи є сканований запис шкідливим. Для цього він обчислює кількість чистих і шкідливих фрагментів сканованого запису

$$\overline{Y_2} = L - \overline{Y_1} = \sum_{k=1}^{L} Y_2^{\ k} \quad (2.26)$$

де L – множина образів сканованого запису; Y_i – вихідне значення і-го нейрона лінійного шару при подачі на вхід мережі к-го образу.

Далі визначається ймовірність приналежності сканованого запису відповідно до чистого і шкідливого класу

Остаточне рішення про приналежність запису до чистого класу арбітр приймає таким чином:

$$Z_1 = \begin{cases} 1, \text{якщо } \text{P}_\text{T} > 80\% \\ 0, \text{інакше} \end{cases} \quad (2.27)$$

Відповідно, рішення про приналежність сканованого запису до шкідливого класу ухвалюється відповідно до наступного виразу:

$$Z_2 = \begin{cases} 1, \text{якщо } \text{P}_\text{F} > 20\% \\ 0, \text{інакше} \end{cases} \quad (2.28)$$

Таким чином, простір вихідних значень арбітра можна представити в табличному вигляді (таблиця 2.1).

Таблиця 2.1.

Простір вихідних значень арбітра

Z_1	Z_2	клас
1	0	Чистий
0	1	Атака
0	0	Не визначено

Якщо вихідні значення арбітра мають нульові значення, то сканований запис відправляється на додаткову перевірку іншому нейромережевому імунному детектору.

В процесі сканування запису, що перевіряється, на нейромережевий детектор послідовно подаються фрагменти запису по методу ковзаючого вікна.

Алгоритм функціонування нейромережевого імунного детектора в режимі сканування запису можна звести до наступної послідовності кроків:

Встановлюються наступні початкові значення:

$$\overline{Y_1}(k-1)=0$$

$$\overline{Y_2}(k-1)=0$$

По методу ковзаючого вікна послідовно подаються вхідні образи $(\mathrm{k}=1,\mathrm{L})$ з сканованого запису на нейронну мережу і для кожного вхідного образу проводяться наступні обчислення:

Визначається евклідова відстань між вхідним чином і ваговими векторами нейронів шару Кохонена.

Визначається нейронний елемент-переможець з номером k

$$D_k=\min_j D_j$$

Обчислюються вихідні значення лінійних нейронних елементів третього шару:

$$Y_j \omega_{ij} Y_k,\ j=\overline{1,2}$$

де ω_{ij} – ваговий коефіцієнт між к-м нейроном шару Кохонена і j-м нейроном вихідного шаруючи.

Визначається кількість чистих і шкідливих фрагментів сканованого запису:

$$\overline{Y_1}(k) = \overline{Y_1}(k-1) + Y_1^k$$

$$\overline{Y_2}(k) = \overline{Y_2}(k-1) + Y_2^k$$

Обчислюється вірогідність приналежності сканованого запису відповідно до чистого і шкідливого класу:

$$P_F = 1 - P_T = \frac{\overline{Y_2}(L)}{L} \cdot 100\%$$

На підставу обчислень вірогідності по формулах ухвалюється рішення про приналежність сканованого запису до одного з класів.

Якщо $Z_1 = 0$ то призначається інший нейромережевий імунний детектор для повторної перевірки запису.

У даному розділі розроблений алгоритм функціонування нейромережевого імунного детектора. Він дозволяє виявляти шкідливі записи, які не входили в навчальну вибірку, в той же час, залишаючись «байдужим» до чистих записів.

Висновки до друго розділу

У розділі запропонована архітектура нейромережевого імунного детектора, яка складається з трьох шарів нейронних елементів і арбітра.

Вона характеризується малим об'ємом навчальної вибірки і відрізняється від відомих тим, що відношення кількості нейронів в шарі Кохонена, що характеризують відповідно чистий і шкідливий класи дорівнює 4/1. Розроблений алгоритм навчання нейромережевих імунних детекторів.

3 РЕАЛІЗАЦІЯ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ ДЛЯ ВИЯВЛЕННЯ КОМП'ЮТЕРНИХ АТАК

3.1. Реалізація модуля виявлення атак

Розроблене програмне забезпечення системи виявлення атак має бути досить швидким і в водночас простим, оскільки опрацьовується та аналізується значний об'єм інформації, який надходитиме із комп'ютера, підключеного до Інтернет. Враховуючи, що швидкість програмного модуля залежить від мови програмування, для реалізації НМ було обрано пакет програмних засобів Matlab.

Matlab представляє собою, так би мовити надбудову над мовою програмування С з оптимізацією під математичні потреби. При цьому програмування в даному пакеті дуже схожі з програмуванням на звичайній мові С, з тією лише різницею, що завдяки широкому виборі математичних функцій та спрощенням синтаксису значно скорочується час отримання потрібного результату. Але для роботи нейронної мережі використовувались стандартні бібліотеки Matlab C Match.

Для виявлення мережевих атак була вибрана нейронна мережа Кохонена

(клас нейронних мереж, основним елементом яких є шар Кохонена. Шар Кохонена складається з адаптивних лінійних формальних нейронів. Як правило, вихідні сигнали шару Кохонена обробляються за правилом «переможець забирає все» - найбільший сигнал перетворюється на одиничний, інші перетворюються в нуль) з наступною конфігурацією: кількість вхідних нейронів = 41 (по кількості параметрів з'єднання), кількість прихованих нейронів = 10, кількість вихідних нейронів = 2 (активність першого нейрона – значення 1 характеризує атаку; активність другого нейрона – значення 1 на виході характеризує нормальне з'єднання).

Структура нейронної мережі представлена на рисунку 3.1.

Алгоритм навчання нейромережевого детектора: для навчання нейромережевого детектора формується навчальна вибірка згідно наступного алгоритму:

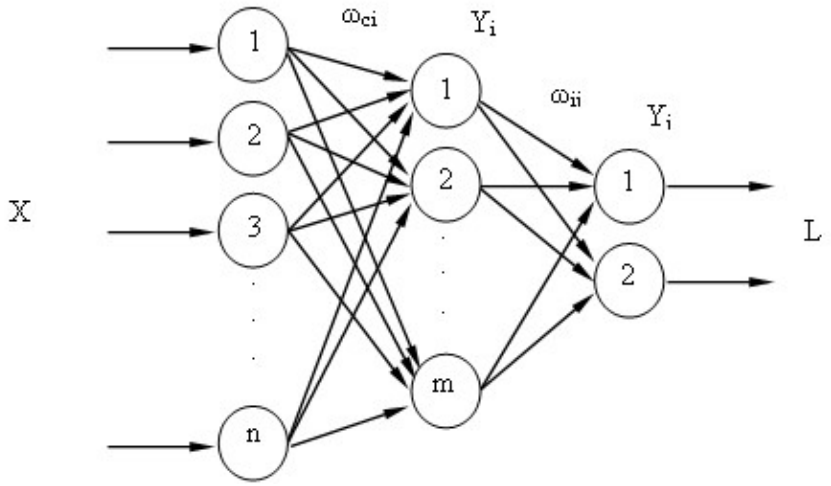


Рисунок 3.1. Нейронна мережа Кохонена

1. Для кожного типу мережевої атаки і нормального з'єднання створюється масив параметрів з'єднання, кількість яких завжди дорівнює 41.

Параметри з'єднання беруться з бази KDD'99. Опис типів атак в базі даних KDD99 представлений в таблиці 3.1.

Таблиця 3.1.

Опис типів атак в базі даних KDD-99

№	Категорія атаки	Опис	Тип атаки
1	2	3	4
1	back	ця атака здійснюється проти apache Web-сервера, який блокується великим потоком запитів, що містять велике число символів (/) в описі URL. Намагаючись обробити ці запити, сервер виявляється не здатним обслужити інші нормальні запити.	DOS

1	2	3	4
2	buffer_overflow	принцип даної атаки побудований на використанні програмних помилок, що дозволяють викликати порушення меж пам'яті і аварійно завершити застосування або виконати довільний бинарний код від імені користувача, під яким працювала вразлива програма. Якщо програма працює під обліковим записом адміністратора системи, то дана атака дозволить отримати повний контроль над комп'ютером жертви.	U2R
3	ftp_write	це атака, коли хакер створює rhost файл для того, щоб зробити анонімну ftp директорію доступну на запис і у результаті дістати доступ до системи. ftp_write ` write to a file on an FTP server.	R2L
4	guess_passwd	злом файлу паролів	R2L
5	imap	IMAP дозволяє користувачам одержувати їх електронні листи з поштового сервера. В останній рік було випущено програмне забезпечення сервера IMAP, в якому містилися помилки, що дозволяють що віддаленому хакеру отримувати повний контроль над машиною. Ця уразливість дуже небезпечна, оскільки велику кількість поштових серверів використовують уразливе програмне забезпечення IMAP.	R2L

6	ipsweep	функція IPSweep, яка проглядає тільки активні хости/об'єкти в заданому діапазоні. IPSweep визначить хосты/об'єкти, навіть в тому випадку, якщо вони не відповідають на пінг (блокуються запити icmp). Користувачі можуть уручну додати IP-адреси, які вони хочуть просканувати або використовувати IPSweep, щоб їх визначити.	Probe
7	land	при атаках Land, хакер посилає жертві пакет, TCP SYN, що містить, де IP-адреси відправника і одержувача ідентичні. Такий пакет повністю блокує роботу системи жертви.	DOS
8	loadmodule	loadmodule використовується серверною програмою для завантаження двох динамічно підвантажуваних драйверів ядра в поточну завантажену систему і створення спеціальних пристроїв в /dev директорії. Із-за наявності помилки в програмі неавторизовані користувачі можуть отримати права доступу до локальної машини	U2R
9	multihop	деякі системи виявлення вторгнень проводять моніторинг трафіку безпосередньо за межами роутера і дивляться за вхідним і витікаючим трафіком тільки мережі в цілому. Сценарій multihop розроблявся спеціально для того, щоб перевірити, чи зможе така система виявити атаку коли хакер спочатку зламає внутрішню машину, і потім, використовуючи цю машину для атаки на решту частини мережі. Це дуже ефективний спосіб атаки, оскільки системи виявлення вторгнень проводять	R2L

		моніторинг трафіку тільки зовні, але не всередині.	
10	neptune	хакер посилає потік SYN пакетів на певний порт комп'ютеражертви. У перебігу деякого короткого проміжку часу після того, як ці пакети були послані, інші користувачі не мають можливості діставання доступу до служб, що забезпечуються цим портом.	DOS
11	nmap	для перевірки вашої системи на уразливість використовується програма NMAP. NMAP - дуже популярна програма, яка використовується хакерами для сканування Internet. Вона працює під Unix, і має багато конфігураційних опцій і використовує декілька трюків, щоб уникнути детектування системами, що детектують вторгнення. NMAP не дозволяє хакерові вторгнутися в систему, але допускає отримання їм корисній інформації про конфігурацію системи і доступні послуги. Програма часто використовується як прелюдія серйознішої атаки.	Probe
12	perl	атаки, що використовують помилки виконання Perl. В результаті, будь-хто може отримати права доступу адміністратора.	U2R
13	phf	використання погано написаних CGI скриптів, що дозволяють виконувати команди на http сервері з привілейованим рівнем. Будь-які CGI програми з використанням таких CGI функцій як escape_shell_cmd() можуть бути уразливими для атак.	R2L

14	pod	Ping of Death. У атаках Ping of Death атакер формує пакет, який містить більше 65,536 байт, що більше межі, визначеної в IP-протоколі. Цей пакет викликає різного роду руйнування в машині, яка його одержує, іноді це викликає rebooting.	DOS
15	portsweep	сканування доступних портів для прослушки портів. Надалі зазвичай використовується для пошуку специфічних сервісів.	Probe
16	rootkit	даний сценарій може бути розглянутий як продовження стандартного сценарію злому. Руткити – це набір програм, які призначені для допомоги хакеру в отримані доступу до віддаленої машини. Типовий руткит містить сніфер, su і інші програми з backdoors, які допомагають в доступі, а також нові версії ps, netstat, і ls, які приховують факт «прослушки» і запускають приховані файли. Одного разу встановлений руткит дозволяє хакеру багато раз отримувати вкрадену інформацію.	U2R
17	satan	збір різної інформації про віддалений комп'ютер або віддалені мережі для вивчення наявності і настройок таких мережевих сервісів як finger, NFS, NIS, ftp and tftp, rexd, statd, і ін. На підставі отриманої інформації виявляються можливі вразливості. Вони можуть полягати в некоректно встановлених або настроєних мережевих сервісів, добре відомих «дірок».	Probe

18	smurf	при атаці "smurf" жертва перевантажується великим числом пакетів-відгуків ICMP. Хакер посилає величезне число ICMP "echo-запитів" за широкомовними адресами багатьох субмереж. Ці пакети містять IP-адресу жертви як адресу відправника. Кожна машина субмережі пошле ICMP відгук машині-жертві. Атаки smurf достатньо небезпечні, оскільки вони є розподіленими. Для протидії цим атакам слід заборонити вхід зовнішніх пакетів, IP-адреса призначення яких є широкомовною. Слід також не пропускати через шлюз зовнішні пакети з адресою відправника з LAN. У вищій мірі ефективним є блокування передачі в Інтернет пакетів з локальної мережі з адресами відправника, невідповідними локальним адресам.	DOS
19	spy	інформаційні «колекціонери», періодично «сканують» компьютер-жертву для збору інформації. Можуть шукати конфіденційну інформацію, або, наприклад, читати особисті листи користувача. Роблять певні кроки для мінімізації можливості виявлення присутності.	R2L
20	teardrop	поки пакет подорожує від відправника до машини одержувача, він може бути роздільний на невеликі фрагменти. Атака Teardrop створює потік IP-фрагментів з надмірно великими значеннями поля зсув (offset). Машина місця призначення, яка намагається відновити ці фальсифіковані фрагменти може блокуватися або навіть здійснити операцію перезавантаження.	DOS

21	warezclient	безпосередньо пов'язаний з warezmaster. Після того, як зловмисник розмістив на FTP сервері нелегальне ПО, користувачі можу викачувати його із зламаного сервера.	R2L
22	warezmaster	ці експлойти асоціюються з FTP протоколом. Стандартно, користувач «гість» не має прав запису на FTP сервер. Атака зв'язана з використанням «дірок» в ПО FTP сервера. Хакер отримує доступ до сервера і завантажує нелегальне ПО.	R2L

Згідно таблиці 3.1 всі атаки діляться на 4 типи: DOS, U2R, R2L, Probe.

DOS - відмова в обслуговуванні, характеризується генерацією великого об'єму трафіку, що приводить до перевантаження і блокування сервера.

U2R - припускає отримання зареєстрованим користувачем привілеїв локального суперкористувача (адміністратора).

R2L - характеризується отриманням доступу незареєстрованим користувачем до комп'ютера з боку віддаленої машини.

Probe - полягає в скануванні портів з метою отримання конфіденційної інформації.

Кожний з цих типів ділиться на декілька категорій.

Параметри облікового запису в базі даних KDD-99 представлені в таблиці 3.2.

Таблиця 3.2.

Параметри облікового запису в базі даних KDD-99

№	Параметр	Опис	Тип
1	2	3	4
	Вбудовані атрибути (intrinsic attributes) Ці атрибути витягуються із зони заголовка мережевих пакетів		

1	duration	час роботи з'єднання (сік)	Continuous, integer
2	protocol_type	тип протоколу (TCP, UDP, ICMP)	symbolic
3	service	служба на стороні приймача (http, smtp, telnet... and other)	symbolic
4	flag	прапор помилки з'єднання (нормальне/помилкове) можливі прапори: SF, S0, S1, S2, S3, OTH, REJ, RSTO, RSTOS0, SH, RSTRH, SHR	symbolic
5	src_bytes	кількість байт від джерела до приймача одного з'єднання	Continuous, integer
6	dst_bytes	кількість байт від приймача до джерела одного з'єднання	Continuous, integer
1	2	3	4
7	land	1 якщо з'єднання з (в) той же самий порт (або IP адрес), 0 інакше	Symbolic, binary
8	wrong_fragment	кількість пакетів з неправильною контрольною сумою	Continuous, integer
9	urgent	кількість термінових пакетів. Терміновий пакет – це пакет з активованим бітом терміновості	Continuous, integer
Атрибути контенту (content attributes) Ці атрибути витягуються експертами із зони контенту мережевих пакетів			

10	hot	кількість "hot" індикаторів, таких як доступ до системної директорії, запуск і завершення програм	Continuous, integer
11	num_failed_logins	кількість невдалих спроб реєстрації в системі за з'єднання	Continuous, integer
12	logged_in	1 якщо реєстрація в системі пройшла вдало, інакше - 0	Symbolic, binary
13	num_compromised	кількість виникнень помилки «не знайдена» за з'єднання	Continuous, integer
14	root_shell	1 якщо отримана root-оболонка, 0 інакше	Continuous, binary
15	su_attempted	1 якщо була спроба виконати команду "Su root", інакше - 0	Continuous, binary
16	num_root	кількість операцій класифікованих як root за з'єднання	Continuous, integer
17	num_file_creations	число операцій створення файлів за з'єднання	Continuous, integer
18	num_shells	кількість входів (login) користувачів-normal	Continuous, integer
19	num_access_files	кількість операцій контролю над файлами за з'єднання	Continuous, integer
20	num_outbound_cmds	кількість команд відправлення в ftp-сесії	Continuous
21	is_host_login	1 якщо користувач реєструється як root або adm, інакше - 0	Symbolic, binary

22	is_guest_login	1 якщо користувач реєструється як guest, anonymous або visitor, інакше - 0	Symbolic, binary
Атрибути трафіку (Traffic attributes) Ці атрибути обчислюються виходячи з попередніх з'єднань. 9 + 10 атрибути поділені на дві групи: 1- часовий трафік (time traffic). 2 – машинний трафік (machine traffic). Різниця між групами полягає в режимі вибору попереднього пакету.			
Time traffic attributes Даний час з'єднання – 2 секунди			
23	count	кількість з'єднань до цього ж IP адресі	Continuous, integer
24	srv_count	кількість з'єднань до цього ж номера порту	Continuous, integer
25	serror_rate	відсоток з'єднань з активованим прапором flag (4) s0, s1, s2 і s3 впродовж з'єднань відображених в count (23)	Continuous, real
26	srv_serror_rate	відсоток з'єднань з активованим прапором flag (4) s0, s1, s2 і s3 впродовж з'єднань відображених в srv_count (24)	Continuous, real
27	rerror_rate	відсоток з'єднань з активованим прапором flag (4) REJ впродовж з'єднань відображених в count (23)	Continuous, real

28	srv_rerror_rate	відсоток з'єднань з активованим прапором flag (4) REJ впродовж з'єднань відображених в srv_count (24)	Continuous, real
29	same_srv_rate	відсоток з'єднань до такої ж служби впродовж з'єднань відображених в count (23)	Continuous, real
30	diff_srv_rate	відсоток з'єднань до інших служб впродовж з'єднань відображених в count (23)	Continuous, real
31	srv_diff_host_rate	відсоток з'єднань до різних видалених машин впродовж з'єднань, вказаних в srv_count (24)	Continuous, real
Machine traffic attributes Дана кількість з'єднань – 100 з'єднань			
32	dst_host_count	кількість з'єднань до такої ж IP адреси	Continuous integer
33	dst_host_srv_count	кількість з'єднань до такого ж номера порту	Continuous, integer
34	dst_host_same_srv_rate	відсоток з'єднань до такої ж служби впродовж з'єднань, відображених в dst_host_count (32)	Continuous, real
35	dst_host_diff_srv_rate	відсоток з'єднань до різних служб впродовж з'єднань, відображених в dst_host_count (32)	Continuous, real

36	dst_host_same_src_port_rate	відсоток з'єднань до такого ж джерела порту впродовж з'єднань, відображених в dst_host_srv_count (33)	Continuous, real
37	dst_host_srv_diff_host_rate	відсоток з'єднань до різних видалених машин впродовж з'єднань, відображених в dst_host_srv_count (33)	Continuous, real
38	dst_host_serror_rate	відсоток з'єднань з активованим прапором flag (4) s0, s1, s2 і s3 впродовж з'єднань відображених в dst_host_count (32)	Continuous, real
39	dst_host_srv_serror_rate	відсоток з'єднань з активованим прапором flag (4) s0, s1, s2 і s3 впродовж з'єднань відображених в dst_host_srv_count (33)	Continuous, real
40	dst_host_rerror_rate	відсоток з'єднань з активованим прапором flag (4) REJ впродовж з'єднань відображених в dst_host_count (32)	Continuous, real
41	dst_host_srv_rerror_rate	відсоток з'єднань з активованим прапором flag (4) REJ впродовж з'єднань відображених в dst_host_srv_count (33)	Continuous, real

Адаптація параметрів з'єднання (варіанти адаптації параметрів представлені в таблицях 3.3-3.6).

Таблиця 3.3.

Варіанти для параметра protocol_type

Символьний тип	tcp	udp	icmp
Цілочисельний тип	1	2	3

Таблиця 3.4.

Варіанти для параметра service

Символьний тип	http	smtp	finger	domain_u	telnet	eco_i	ntp_u	auth	ecr_i	other	ftp
Цілочисельний тип	1	2	3	4	5	6	7	8	9	10	11
Символьний тип	login	private	imap	time	rje	link	ctf	uucp	hostnames	gopher	pm_dump
Цілочисельний тип	12	13	14	15	16	17	18	19	20	21	22

Таблиця 3.5.

Варіанти для параметра flag

Символьний тип	SF	S0	S1	S2	S3	OTH	REJ	RSTO	RSTOS0	SH	RSTRH	SHR
Цілочисельний тип	1	2	3	4	5	6	7	8	9	10	11	12

Таблиця 3.6.

Адаптація параметрів з'єднання

№	Параметр	Тип	Приклад (до)	Перетворення	Приклад (після)
1	2	3	4	5	6
1.	duration (час роботи з'єднання (сік))	integer	0	немає	0
2.	protocol_type (тип протоколу (tcp, udp, icmp))	symbolic	tcp	a > 1	1
3.	service (служба на стороні приймача (http, smtp, telnet... and other))	symbolic	http	a > 1	1
4.	flag (прапор помилки з'єднання (нормальне/помилкове) можливі прапори: sf, s0, s1, s2, s3, oth, rej, rsto, rstos0, sh, rstrh, shr)	symbolic	sf	a > 1	1
5.	src_bytes (кількість байт від джерела до приймача одного з'єднання)	integer	181	немає	181
6.	dst_bytes (кількість байт від приймача до джерела одного з'єднання)	integer	5450	немає	5450
7.	land (1 якщо з'єднання з (в) той же самий порт (або ip адресу), 0 інакше)	binary	0	немає	0

8.	wrong_fragment (кількість пакетів з неправильною контрольною сумою)	integer	0	немає	0
9.	urgent (кількість термінових пакетів. терміновий пакет – це пакет з активованим бітом терміновості)	integer	0	немає	0
10.	hot (кількість "hot" індикаторів, таких як доступ до системного директорія, запуск і завершення програм)	integer	0	немає	0
11.	num_failed_logins (кількість невдалих спроб реєстрації в системі за з'єднання)	integer	0	немає	0
12.	logged_in (1 якщо реєстрація в системі пройшла вдало, інакше - 0)	binary	1	немає	1
13.	num_compromised (кількість виникнень помилки «не знайдена» за з'єднання)	integer	0	немає	0
14.	root_shell (1 якщо отримана root-оболонка, 0 інакше)	binary	0	немає	0
15.	su_attempted (1 якщо була спроба виконати команду "Su root", інакше - 0)	binary	0	немає	0

16.	num_root (кількість операцій класифікованих як root за з'єднання)	integer	0	немає	0
17.	num_file_creations (число операцій створення файлів за з'єднання)	integer	0	немає	0
18.	num_shells (кількість входів (login) користувачів-normal)	integer	0	немає	0
19.	num_access_files (кількість операцій контролю над файлами за з'єднання)	integer	0	немає	0
20.	num_outbound_cmds (кількість команд відправлення в ftp-сесії)	integer	0	немає	0
21.	is_host_login (1 якщо користувач реєструється як root або adm, інакше - 0)	binary	0	немає	0
22.	is_guest_login (1 якщо користувач реєструється як guest, anonymous або visitor, інакше - 0)	binary	0	немає	0
23.	count (кількість з'єднань до цієї ж ip адреси)	integer	8	немає	8
24.	srv_count (кількість з'єднань до цього ж номера порту)	integer	8	немає	8

25.	serror_rate (відсоток з'єднань з активованим прапором flag (4) s0, s1, s2 і s3 впродовж з'єднань відображених в count (23))	real	0,00	0,1 > 1	0
26.	srv_serror_rate (відсоток з'єднань з активованим прапором flag (4) s0, s1, s2 і s3 впродовж з'єднань відображених в srv_count (24))	real	0,00	0,1 > 1	0
27.	rerror_rate (відсоток з'єднань з активованим прапором flag (4) rej впродовж з'єднань відображених в count (23))	real	0,00	0,1 > 1	0
28.	srv_rerror_rate (відсоток з'єднань з активованим прапором flag (4) rej впродовж з'єднань відображених в srv_count (24))	real	0,00	0,1 > 1	0
29.	same_srv_rate (відсоток з'єднань до такої ж служби впродовж з'єднань відображених в count (23))	real	1,00	0,1 > 1	100
30.	diff_srv_rate (відсоток з'єднань до інших служб впродовж з'єднань	real	0,00	0,1 > 1	0

	відображених в count (23))				
31.	srv_diff_host_rate (відсоток з'єднань до різних видалених машин впродовж з'єднань, вказаних в srv_count (24))	real	0,00	0,1 > 1	0
32.	dst_host_count (кількість з'єднань до такої ж ip адреси)	integer	9	немає	9
33.	dst_host_srv_count (кількість з'єднань до такого ж номера порту)	integer	9	немає	9
34.	dst_host_same_srv_rate (відсоток з'єднань до такої ж служби впродовж з'єднань, відображених в dst_host_count (32))	real	1,00	0,1 > 1	100
35.	dst_host_diff_srv_rate (відсоток з'єднань до різних служб впродовж з'єднань, відображених в dst_host_count (32))	real	0,00	0,1 > 1	0
36.	dst_host_same_src_port_rate (відсоток з'єднань до такого ж джерела порту впродовж з'єднань, відображених в dst_host_srv_count (33))	real	0,11	0,1 > 1	11

37.	dst_host_srv_diff_host_rate (відсоток з'єднань до різних видалених машин впродовж з'єднань, відображених в dst_host_srv_count (33))	real	0,00	0,1 > 1	0
38.	dst_host_serror_rate (відсоток з'єднань з активованим прапором flag (4) s0, s1, s2 і s3 впродовж з'єднань відображених в dst_host_count (32))	real	0,00	0,1 > 1	0
39.	dst_host_srv_serror_rate (відсоток з'єднань з активованим прапором flag (4) s0, s1, s2 і s3 впродовж з'єднань відображених в dst_host_srv_count (33))	real	0,00	0,1 > 1	0
40.	dst_host_rerror_rate (відсоток з'єднань з активованим прапором flag (4) rej впродовж з'єднань відображених в dst_host_count (32))	real	0,00	0,1 > 1	0
41.	dst_host_srv_rerror_rate (відсоток з'єднань з активованим прапором flag (4) rej впродовж з'єднань відображених в dst_host_srv_count	real	0,00	0,1 > 1	0

Формування навчальної вибірки для кожного окремого детектора.

Кожен окремий детектор навчається розпізнавати певний тип атак, наприклад, DoS-атаки. Для цього в навчальну вибірку входить 80% трафіку цього типу атаки і 20% трафіку нормального з'єднання. Оскільки НМ

навчається за принципом «переможець забирає все» (winner take all) на виході завжди буде або 1/0, або 0/1. Активність першого нейрона характеризує атаку, активність другого нейрона характеризує нормальне з'єднання.

3.2. Тестування системи

Верифікація програмного модуля проводилось із використанням засобів Matlab.

Для цього використовувалась навчена НМ, програмний код якої наведено у додатку Д.

Результати експериментів представлені в таблиці 3.7.

Таблиця 3.7.

Результати експериментів

Навчання	normal	back	buffer_overflow	ftp_write	guess_passwd	imap	ipsweep	land	loadmodule	multihop	neptune	nmap	perl
back	100%	100%	0%	0%	0%	8%	0%	0%	0%	29%	0%	0%	0%
ipsweep	93%	0%	0%	25%	96%	75%	100%	100%	0%	0%	80%	100%	0%
neptune	93%	0%	0%	38%	80%	75%	100%	100%	0%	14%	100%	100%	0%
nmap	93%	0%	0%	50%	87%	75%	100%	100%	0%	14%	100%	100%	0%
portsweep	93%	0%	0%	13%	97%	75%	97%	100%	0%	0%	100%	100%	0%
satan	97%	0%	27%	38%	93%	17%	97%	86%	44%	57%	100%	100%	0%
smurf	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

Навчання	phf	pod	portsweep	rootkit	satan	smurf	spy	teardrop	warezclient	warezmaster
back	0%	0%	0%	0%	0%	0%	0%	0%	3%	80%
ipsweep	0%	0%	72%	20%	80%	0%	0%	100%	90%	0%
neptune	0%	0%	100%	30%	90%	33%	0%	100%	93%	0%
nmap	0%	0%	100%	30%	90%	0%	0%	100%	93%	0%
portsweep	0%	0%	100%	40%	97%	100%	0%	100%	3%	5%
satan	100%	0%	100%	70%	100%	100%	0%	100%	7%	90%
smurf	0%	0%	0%	0%	0%	100%	0%	0%	3%	0%

Аналізуючи результати експериментів можна виділити наступні закономірності:

Детектор, навчений на одному типі атаки здатний виявляти атаки іншого типу, що говорить про те, що атаки можуть мати схожу сигнатуру.

Для точнішої роботи окремого детектора необхідно навчальна вибірка більшої розмірності (в даному випадку у вибірці використовувалися 35 з'єднань, 28 з яких складали з'єднання відповідного типу атаки).

Деякі атаки так і не були виявлені. Причини криються у вибірці KDD'99, що містить 10% всіх з'єднань. Кількість з'єднань для цих атак було недостатньо для формування навчальної вибірки для відповідного детектора. Проблема вирішується використанням повної бази KDD'99.

Відсоток помилкових спрацьовувань, коли детектор класифікує нормальний трафік як атаку, вдасться знизити збільшенням розмірності навчальної вибірки і за рахунок використання принципів штучних імунних систем.

Запропонований підхід продемонстрував здатність виявляти мережеві атаки, і показав добрі результати. Виявлені недоліки передбачається усувати введенням принципів штучних імунних систем для функціонування детекторів і збільшенням розмірності навчальної вибірки.

3.3. Застосування систем виявлення атак

Системи виявлення атак представляють собою важливий сегмент ринку технологій захисту. Продукти з виявлення атак розглядаються багатьма як логічне доповнення до міжмережевим екранів, розширюючи таким чином можливості системних адміністраторів в управлінні безпекою, які включають аудит захисту, моніторинг, розпізнавання атак та реагування на них.

Системи виявлення атак допомагають комп'ютерних систем підготуватися до прийому і «відбиванню» атак. Вони виконують цю роль шляхом збору інформації про низку системних і мережевих ресурсів, потім вони аналізують зібрану інформацію на предмет ідентифікації проблем захисту. У деяких випадках системи виявлення атак дозволяють користувачеві встановлювати відповідні реакції на зловживання у реальному масштабі часу.

Системи виявлення атак виконують цілий ряд функцій:

1. Моніторинг та аналіз діяльності користувачів і обчислювальних систем;
2. Аудит конфігурацій системи і вразливостей;
3. Оцінка цілісності найбільш важливих системних файлів і файлів даних;
4. Розпізнавання шаблонів активності, що відображають відомі атаки;
5. Статистичний аналіз шаблонів аномальної активності;
6. Управління журналами реєстрації операційної системи з розпізнаванням діяльності користувача, що відбиває порушення політики безпеки.

Деякі системи мають додаткові характеристики, що включають:

- автоматичну інсталяцію патчів ПЗ, що поставляються продавцем;
- інсталяцію і роботу серверів-пасток для запису інформації про порушників.

Комбінація цих характеристик дозволяє системним адміністраторам більш ефективно здійснювати моніторинг, аудит та аналіз систем і мереж. Ця безперервна діяльність з аналізу та аудиту є необхідною частиною стандартної практики з управління захистом.

Системи пошуку вразливостей (також відомі як сканери безпеки або аналізу захищеності) проводять всебічні дослідження систем з метою визначення вразливостей, які можуть призвести до порушень захисту. При проведенні цих досліджень ці системи реалізують дві стратегії. Перша - пасивна, - механізми, що діють на рівні операційної системи і додатків, інспектують конфігураційні файли системи на предмет наявності неправильних параметрів; файли з системними паролями на предмет наявності легко вгадувані паролів, а також інші системні об'єкти на предмет порушень політики безпеки. Наступні перевірки в більшості випадків є активними, вони здійснюють аналіз на мережевому рівні, відтворюючи найбільш поширені сценарії атак, і аналізують реакції системи на ці сценарії.

Результати, отримані від засобів аналізу вразливостей, становлять "миттєвий знімок" стану захисту системи в даний момент часу. Незважаючи на те, що ці системи не можуть надійно виявляти атаку в процесі її розвитку, вони можуть визначити те, що атака є можливою і, більш того, іноді вони можуть визначити, що атака мала місце. Оскільки вони пропонують можливості, аналогічні можливостям, які надаються системами виявлення атак, ми включили їх у сферу технологій та продуктів виявлення атак.

Головне завдання засобів пошуку вразливостей і виявлення атак полягає в тому, щоб автоматизувати рутинні та утомливі функції управління захистом системи, і зробити їх зрозумілими для тих, хто не є експертом в галузі захисту інформації. Тому продукти розробляються з графічним інтерфейсом, зручним і зрозумілим для користувача, що допомагає адміністраторам безпеки легко і швидко здійснювати інсталяцію, конфігурацію і використання відповідних продуктів.

Більшість систем включають інформацію про виявляються проблеми, включаючи вказівки на те, як усунути ці проблеми. Ця інформація служить в якості цінного керівництва для тих, кому необхідно підвищити свої професійні навички в області захисту інформації. Багато продавців пропонують консультаційні послуги та послуги з інтеграції своїх продуктів в технологію обробки інформації організації для того, щоб допомогти покупцям успішно використовувати дані системи з метою досягнення своїх конкретних цілей щодо захисту інформації.

Висновки до третього розділу

У третьому розділі наведено опис розробленого програмного комплексу, його особливостей побудови та структуру написання програмного коду. Опис програми проводиться з використанням екранних форм.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Вимоги щодо охорони праці при роботі з комп'ютерами. Інструкція для програміста.

Сучасний розвиток технічного та технологічного стану виробництва передбачає постійну автоматизацію та оптимізацію виробничих процесів. Сьогодні, напевно, важко уявити компанію, господарська діяльність в якій здійснювалась би без використання комп'ютерної техніки. Через масовий характер робіт, що виконуються працівниками за допомогою комп'ютера, законодавством України чітко врегульовано норми та вимоги до використання комп'ютерної техніки на підприємстві, безпосередньо й охорона праці при роботі з комп'ютером.

Вимоги до приміщення

Приміщення, в яких планується установка та подальша робота з комп'ютером, повинні відповідати проектній документації будинку, погодженій з уповноваженими державними органами. Крім того, роботодавець повинен враховувати санітарні нормативи освітлення, вимоги до параметрів мікроклімату (температура, відносна вологість), ступеня і сили вібрації, звукового шуму і вогнестійкості приміщення, а також характеристики електромагнітного, ультрафіолетового та інфрачервоного полів. Конкретні показники зазначених санітарних норм див. в Державних санітарих правилах і нормах роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98, затверджених Постановою Головного державного санітарного лікаря України №7 від 10 грудня 1998 року.

Правила поширюються на умови й організацію праці при роботі з візуальними дисплейними терміналами (ВДТ) усіх типів вітчизняного та зарубіжного виробництва на основі електронно-променевих трубок (ЕПТ), що використовуються в електронно-обчислювальних машинах (ЕОМ) колективного використання та персональних ЕОМ (ПЕОМ). Так, наприклад,

роботодавцю заборонено установлювати комп'ютери в приміщеннях, розташованих у підвалах будинків.

Для уникнення можливих аварій та замикань, поряд з приміщеннями, де вестиметься робота з комп'ютером (над чи під ними), також не дозволяється проведення робіт, що потребують здійснення надмірно вологих технологічних процесів. Відповідне приміщення повинно бути укомплектоване системами центрального або індивідуального опалення, кондиціонування чи вентиляції повітря. Але при установці зазначених систем, необхідно переконатись, що батареї опалення, водопровідні труби, вентиляційні кабелі тощо, надійно сховані під захисними щитками, які перешкоджатимуть можливому потраплянню робітника під напругу.

У кожній кімнаті, де обладнуватимуться робочі місця співробітників, що працюватимуть на комп'ютері, повинні бути наявні елементи природного та штучного освітлення. При цьому, на вікнах слід встановити легко регульовані жалюзі чи штори, які дозволять працівникам коригувати рівень освітлення в приміщенні. Бажано розмістити комп'ютери в кімнаті таким чином, щоб світло потрапляло на екрани моніторів з півдня чи північного сходу. З метою досягнення максимального рівня безпечності і охорони праці при роботі з комп'ютером, виробничі приміщення необхідно обладнати аптечками першої медичної допомоги, системами автоматичної пожежної сигналізації і вогнегасниками. В приміщенні, в якому разом працюють 5 або більше комп'ютерів, на видимому місці установлюється службовий вимикач, який у разі потреби дозволить повністю відключити електричне живлення кімнати.

Вимоги до особистого робочого місця програміста

Роботодавець, який використовує найману працю робітників, повинен забезпечити відповідність їхніх робочих місць комфортним та безпечним умовам. Розмір одного робочого місця має становити не менше 6 квадратних метрів. При необхідності, суміжні робочі місця співробітників, що працюють з комп'ютером, слід розділити перегородками висотою до 2 метрів. При визначенні достатнього розміру приміщення і робочого місця на одну особу необхідно додатково враховувати шафи, сейфи, тумби або інші предмети

меблів чи обладнання, які знаходяться в кімнаті. На столі працівника можливо розмістити допоміжні для роботи пристрої (принтери, колонки, сканери), а також місця для зберігання документів, за умови, що це не обмежуватиме видимість екрану і не заважатиме працівнику. У разі надмірного шуму чи вібрації технічного обладнання, роботодавець повинен забезпечити працівників антивібраційними килимками. Робочий стілець співробітника має бути підйомно-поворотним, легко регульованим за висотою та забезпечувати належну підтримку та зручне положення спини і хребта особи. Щодня необхідно проводити вологе прибирання приміщення, та очищати робоче місце та безпосередньо монітор комп'ютера від запиленості.

На підприємстві забороняється: проводити ремонт та технічне обслуговування комп'ютера за робочим місцем працівника; самочинно ремонтувати або намагатись здійснити технічне налагодження комп'ютера без залучення компетентних спеціалістів; складувати на робочому місці зайві документи, деталі та предмети, що не потрібні для роботи; використовувати монітори з нечітким зображенням та монітори, у яких наявні поламки екрану; працювати з матричним принтером без антивібраційного покриття та зі знятою кришкою. Допускати до роботи осіб, які не пройшли затверджений на підприємстві курс охорони праці для роботи з комп'ютером, не дозволяється.

Законодавство:

– Наказ Державного комітету України з промислової безпеки, охорони праці та гірничого нагляду «Про затвердження Правил охорони праці під час експлуатації електронно-обчислювальних машин» від 26.03.2010 № 65;

– Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98, затверджені постановою Головного державного санітарного лікаря України від 10.12.1998 № 7;

– Примірну інструкцію з охорони праці під час експлуатації електронно-обчислювальних машин, затверджену наказом Міністерства доходів і зборів України від 05.09.2013 № 443.

4.2. Забезпечення електробезпеки користувачів ПК.

Приміщення із робочими місцями користувачів комп'ютерів для забезпечення електробезпеки обладнання, а також для захисту від ураження електричним струмом самих користувачів ПК повинні мати достатні технічні засоби захисту відповідно до ГОСТ 12.1.009-76, НПАОП 40.1-1.07-01 "Правила експлуатації електрозахисних засобів", НПАОП 40.1-1.21-98 "Правила безпечної експлуатації електроустановок споживачів", НПАОП 40.1-1.32-01 "Правила будови електроустановок. Електрообладнання спеціальних установок".

З метою запобігання ушкодженням, що можуть статися через ураження електричним струмом, загоряння, коротке замикання тощо, розроблено загальний стандарт безпеки IEC 950. Загальним стандартом електробезпечності для країн Європейської співдружності є Cemark.

Під час проектування систем електропостачання, монтажу силового електрообладнання та електричного освітлення будівель та приміщень для ПЕОМ необхідно дотримуватись вимог вищеназваних нормативно-правових актів, а також СН 357-77 "Инструкция по проектированию силового осветительного оборудования промышленных предприятий", затверджених Держбудом СРСР, ГОСТу 12.1.006, ГОСТу 12.1.030 "ССБТ. Электробезопасность. Защитное заземление, зануление", ГОСТу 12.1.019 "ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты", ГОСТу 12.1.045, ВСН 59-88 Держкомархітектури СРСР "Электрооборудование жилых и общественных зданий. Нормы проектирования", Правил пожежної безпеки в Україні, ДСанПіН 3.3.2.007-98, розділів СНиП, що стосуються штучного освітлення і електротехнічних пристроїв, та вимог нормативно-технічної і експлуатаційної документації заводу-виробника ПЕОМ.

ЕОМ, периферійні пристрої ЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ, інше устаткування (апарати управління, контрольно-вимірювальні прилади, світильники тощо), електропроводи та

кабелі за виконанням та ступенем захисту мають відповідати класу зони за ПУЕ, мати апаратуру захисту від струму короткого замикання та інших аварійних режимів.

Під час монтажу та експлуатації ліній електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежувати застосування проводів з легкозаймистою ізоляцією і, за можливості, перейти на негорючу ізоляцію.

Лінія електромережі для живлення ЕОМ, периферійних пристроїв ЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ виконується як окрема групова трипровідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів.

Використання нульового робочого провідника як нульового захисного провідника забороняється. Нульовий захисний провід прокладається від стійки групового розподільчого щита, розподільчого пункту до розеток живлення. Не допускається підключення на щиті до одного контактного затискача нульового робочого та нульового захисного провідників. Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі повинна бути не менше площі перерізу фазового провідника.

Усі провідники повинні відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту, вимогам ПУЕ.

У приміщенні, де одночасно експлуатується або обслуговується більше п'яти персональних ЕОМ, на помітному та доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

ПЕОМ, периферійні пристрої ПЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ повинні підключатися до

електромережі тільки з допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників повинні мати спеціальні контакти для підключення нульового захисного провідника. Конструкція їх має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Необхідно унеможливити з'єднання контактів фазових провідників з контактами нульового захисного провідника.

Неприпустимим є підключення ПЕОМ та периферійних пристроїв ПЕОМ до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв.

Електромережі штепсельних з'єднань та електророзеток для живлення ПЕОМ, периферійних пристроїв слід виконувати за магістральною схемою, по 3…6 з'єднань або електророзеток в одному колі. Штепсельні з'єднання та електророзетки для напруги 12 В та 36 В за своєю конструкцією повинні відрізнятися від штепсельних з'єднань для напруги 127 В та 220 В і мають бути пофарбовані в колір, який візуально значно відрізняється від кольору штепсельних з'єднань, розрахованих на напругу 127 В та 220 В.

Індивідуальні та групові штепсельні з'єднання та електророзетки необхідно монтувати на негорючих або важкогорючих пластинах з урахуванням вимог ПУЕ та Правил пожежної безпеки в Україні.

Електромережу штепсельних розеток для живлення ПЕОМ, периферійних пристроїв ПЕОМ при розташуванні їх уздовж стін приміщення прокладають по підлозі поряд зі стінами приміщення, як правило, в металевих трубах і гнучких металевих рукавах з відводами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання.

При розташуванні в приміщенні за його периметром до 5 ПЕОМ, використанні трипровідникового захищеного проводу або кабелю в оболонці з негорючого або важкогорючого матеріалу дозволяється прокладання їх без металевих труб та гнучких металевих рукавів.

ВИСНОВКИ

1. На сьогоднішній день застосовуються різноманітні методи і засоби для захисту від мережевих атак, проте всі вони мають ряд істотних недоліків, і не здатні повною мірою захистити користувача від вторгнень. Для надійного захисту комп'ютерних систем від мережевих вторгнень необхідно розробляти принципово нові методи захисту.

2. Перспективним напрямком забезпечення безпеки комп'ютерних систем є використання методів штучного інтелекту (ШІ), таких як нейронні мережі, штучні імунні системи, еволюційне програмування і т.д., які вже довели свою ефективність у вирішенні складних задач розпізнавання, класифікації, управління і виявлення.

3. Розроблений алгоритм навчання нейромережевого детектора системи аналізу мережевого трафіку для виявлення комп'ютерних атак.

4. Вдосконалений алгоритм функціонування нейромережевих детекторів, що дозволяє ефективно ідентифікувати комп'ютерні атаки.

5. Відповідно до розроблених алгоритмів побудови нейромережевої системи аналізу мережевого трафіку, а також з урахуванням найбільш важливих характеристик комп’ютерних атак, розроблена нейромережева система виявлення мережевих атак.

6. Розроблена система функціонує в режимі реального часу, що забезпечує високий рівень виявлення комп’ютерних атак.

Проведені експерименти показали, що система безпеки, побудована на основі комбінування методів нейронних мереж і методів штучних імунних систем здатна виявляти невідомі атаки на комп’ютерні системи.

ПЕРЕЛІК ДЖЕРЕЛ

1. А.В. Лукацкий. Вопросы инфомационной безопасности.- http://www.infosec.ru/press/pub/t_v_1.zip.

2. А.В. Лукацкий. Адаптивное управление защитой. – Сети, №10, 1999р.

3. Э.С. Абрамов. Разработка комбинированной архитектуры системы обнаружения и выявления сетевых атак // Материалы 3-ей международной научно-практической конференции «Информационная безопасность», Таганрог 2007р.

4. Вильям Столлинге. Криптография и защита сетей. Принципы и практика, 2е издание. - М.: “Вильямс”, 2006.

5. М. Ранум. Обнаружения атак: реальность и мифы -

6. А.В. Лукацкий. Обнаружения атак. – 2-е изд., перераб. и доп. СПб.:БХВ – Петербург, 2008.

7. Edward Amoroso. Intrusion Detection. An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response. Intrusion.Net Books, 1999.

8. Б. Анин. Защита компьютерной информации - http://bugtraq.ru/library/books/attack1/chapter7/c72.html.

9. Richard Power, "2000 CSI/FBI Computer Crime and Security Survey", Computer and Security Issues and Trends, Vol. 6, No. 1, Spring 2000.

10. "UK e-business at risk from hackers, reveals report" available at http://www.ananova.com/news/story/internet_security_79176.html.

11. А.В. Лукацкий. Безопасность сети банка глазами специалистов - http://securitylab.ru/tools/22111.html.

12. В.В. Домареев. Защита информации и безопасность компьютерных систем - http://www.softline.ru/course.

13. А.В. Лукацкий. Системы обнаружения атак – СПб.: БХВ – СанктПетербург, 1999. – 58 С.

14. CERT Coordination Center. CERT Advisory CA-1998-01: Smurf IP Denialof-Service Attacks. <http://www.cert.org/advisories /CA-1998-01.html> (January 5, 1998; last revised March 13, 2000).

15. Y.Ho, D. Frinke, D. Tobin. Planning, Petri-Nets and Intrusion Detection, Taylor & Francis, 1998 324 pages.

16. Justin J. Lister. Latest Developments and New Technologies for Detecting and Preventing Computer, Communication and Financial Fraud.

17. О.Ю. Гаценко. Защита информации. М.: Сентябрь, 2001. – 228с.

18. В.В. Маснянкин. Перевод на русский язик материалов honeynet project - http://cybervlad.net/lspitz/enemy/index.html.

19. The Honeynet Project. Know Your Enemy: Honeynets. Whitepaper, 2003 — http://project.honeynet.org/papers/honeynet/index.html.

20. The Honeynet Project. Know Your Enemy: The motives and psychology of the blackhat community. Whitepaper, 2000 - http://project.honeynet.org/papers/motives/index.html.

21. В.И. Завгородний. Комплексная защита информации в компьютерных системах. - М.: Логос. 2001. – 264 С.

22. The Honeynet Project. Know Your Enemy: The Tools and Methodologies of the Script Kiddie. Whitepaper, 2000 — http://project.honeynet.org/papers/enemy/index.html.

23. Л. Спитцнер. Honeynet Project: ловушка для хакеров// Открытые системы. - № 7-8, 2003. — http://www.osp.ru/os/2003/07-08/061.htm.

24. Spitzner L. Honeypots: Tracking Hackers. — Addison-Wesley Professional, 2002. - 480 p.

25. Postel, J. (ed.), "Internet Protocol - DARPA Internet Program Protocol Specification," RFC 791, USC/Information Sciences Institute, September 2002 - http://www.citforum.ru/internet/tifamily/icmpspec.shtml.

26. Spitzner L. Honeypots: Tracking Hackers. — Addison-Wesley Professional, 2002. — 480 p.

27. Ю.В. Романец, П.А. Тимофеев, В.Ф. Шаньгин. Защита информации в компьютерных системах и сетях. М.: Радио и связь. 2001. – 376 С.

28. Е.А. Степанов, И.К. Корнеев. Информационная безопасность и защита информации. Учебное пособие. М.: Инфа-М, 2001. – 304 С.

29. А.В. Галицкий, С.Д. Рябко, В.Ф. Шаньгин. Защита информации в сети. Анализ технологий и синтез решений. М.: ДМК Пресс, 2004. – 616 С.

30. А.В. Соколов, В.Ф. Шаньгин. Защита информации в распределенных корпоративных сетях и системах. М.: ДМК Пресс, 2002. – 656 С.

31. H. Demuth., M. Beale. Neural Network Toolbox/ForUse with MATLAB - http://www.mathworks.com, ftp.mathworks.com. – 1992. – 1997 by The MathWorks, Inc.

32. R. Malayter. By MD5 and SHA. - www.secure-hash-algorithm-md5-sha-1.co.uk.

33. M.T. Hagan., H. Demuth. Neural Network Design, Boston: PWS Publishing Company, 1996.

34. Головко, В.А. Нейронные сети: обучение, организация, применение / В.А. Головко // Нейрокомпьютеры и их применение : учеб. пособие / В.А. Головко. – М., 2001 – 256 с.

35. H. Debar., M. Becke., & D. Siboni. (1992). A Neural Network Component for an Intrusion Detection System. In Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy.

36. L. Kevin., Henning, R. Rhonda., and H. Jonathan. (1990). A Neural Network Approach Towards Intrusion Detection. In Proceedings of the 13th National Computer Security Conference.

37. О.А. Савенкова. Исследование алгоритмов обучения модели нейронной сети при распознавании речевых сигналов - http://www.dgma.donetsk.ua/%7Ekiber/index.htm.

38. Houle, Kevin J.; Weaver, George M.; Long, Neil; & Thomas, Rob. Trends In Denial of Service Attack Technology, CERT Coordination Center, October 2001 - http://www.cert.org/archive/pdf/DoS_trends.pdf.

39. F.H. Vonwangelin. Context menu MD5 - http://www.vonwangelin.com/md5/index.html.

40. Robert Rothenburg. MD5 implementation for Turbo Pascal - http://www.squid/spylog.com.

41. В.М. Варакін. Структура системы и ее компоненты. - http://www.unicyb.kiev.ua/~kga/pi/lec7.doc.

42. И. Трифаленков, В. Макоев. Критерии выбора средств защиты информации. pdf CIO N5/2002.

43. Маняшин С.М., Жуков И.Ю., Свирин И.С., Юраков Ю.Д. Защита информации в автоматизированных системах военного назначения в условиях современных угроз // Безопасность сетей и средств связи. – 2006. – №1. – С. 137 – 146.

44. Свирин И.С. Методология построения и оценки адекватности модели штатного поведения системы // Методы и технические средства обеспечения безопасности информации: СПб., 2006. – С. 104.

45. Фролов А.В., Фролов Г.В. Глобальные сети компьютеров. Практическое введение в Internet, E-mail, FTP, WWW, и HTML, программирование для Windiws Sockets. - Диалог - МИФИ, 1996. - 283 с.

46. Левин В.К. Защита информации в информационно-вычислительных системах и сетях // Программирование. - 1994. - N5. - С. 5-16.

47. В.Г. Олифер, Н.А. Олифер Компьютерные сети. Принципы, технологии, протоколы. — СПб.:Питер, 2001 – 672 С.

48. Максим Кульгин. Компьютерные сети. Практика построения. П.: Питер, 2003. – 464 С.

49. Джеймс Ф., Куроуз В., Кит В. Компьютерные сети. Многоуровневая архитектура Интернета. П.: Питер, 2004. – 765 С.

50. Марк Спортаж, Френк Паппас. Компьютерные сети и сетевые технологии. М.: ТИД «ДС», 2002. – 736 С.

51. Carole A. Lane. Naked in cyberspace, Wilton, CT, 1997. – 513 pages.

52. Proceedings of the NATO Advanced Research Workshop on Cyberspace Security and Defense: Research Issues. Gdansk, Poland, 6-9 September 2008.

53. Дасгупта, Д. Искусственные иммунные системы и их применение / Д.

ДОДАТКИ

Додаток А

М А Т Е Р І А Л И

ІХ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

8–9 грудня 2021 року

ТЕРНОПІЛЬ
2021

УДК 004.031.6

Л. Матійчук, кан. екон. наук, доцент, І. Павлов, В. Сташук
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ТЕОРЕТИЧНЕ ОБҐРУНТУВАННЯ МЕТОДУ ВИЯВЛЕННЯ КОМП'ЮТЕРНИХ АТАК

UDC 004.031.6

L. Matiychuk, PhD, Assoc. Prof., I. Pavlov, V. Stashuk

THEORETICAL JUSTIFICATION OF THE METHOD OF DETECTION OF COMPUTER ATTACKS

Мета виявлення вторгнень на перший погляд дуже проста: виявити проникнення в ІС. Проте це вельми складне завдання. Насправді, системи виявлення вторгнень ніяких вторгнень взагалі не виявляють вони тільки виявляють ознаки вторгнень під час таких атак.
Системи виявлення атак призначені для виявлення і протидії мережевим атакам зловмисників. Вони є спеціалізованим програмно-апаратним забезпеченням з типовою архітектурою, що включає наступні компоненти (рис.1): модулі-датчики для збору необхідної інформації про МТ в ІС; модуль виявлення атак, що виконує обробку даних, зібраних датчиками, з метою виявлення інформаційних атак; модуль реагування на виявлені атаки; модуль зберігання конфігураційної інформації, а також інформації про виявлені атаки. Таким модулем, як правило, виступає стандартна СУБД, наприклад MS SQL Server; модуль управління компонентами системи виявлення атак.

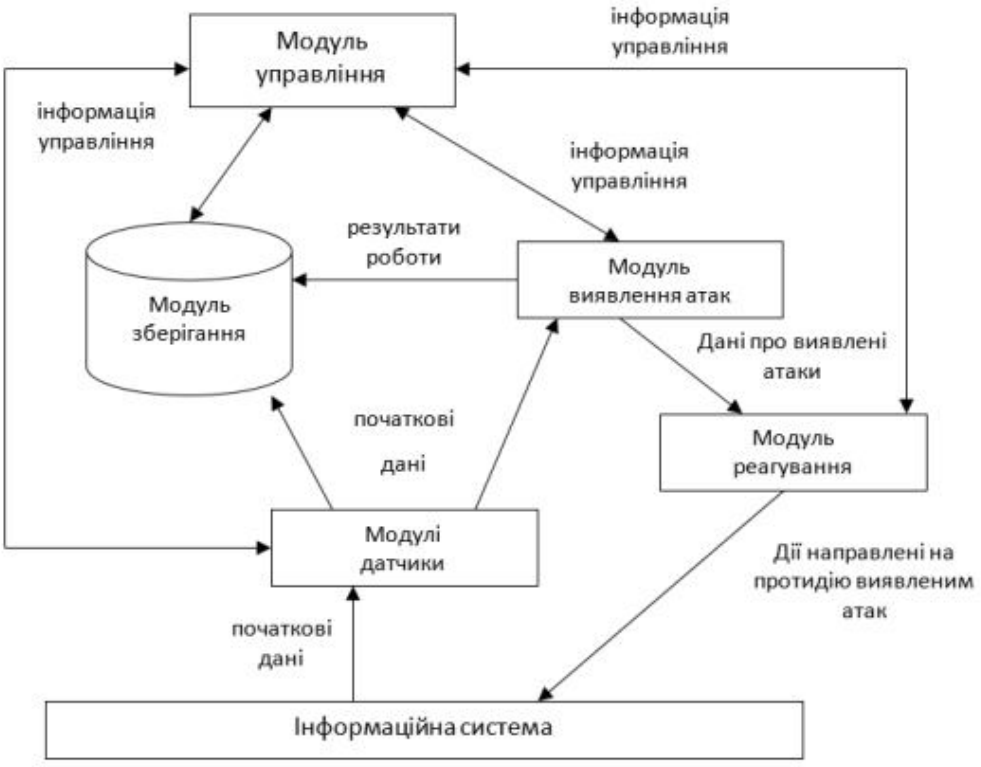


Рисунок 1. Типова архітектура виявлення атак

Для точного виявлення вторгнень необхідні надійні і вичерпні дані про те, що відбувається в системі, яка захищається. Взлом системи можливий як із сторини комп'ютера, що знаходиться в локальній мережі так і через глобальну мережу Інтернет. Проте сучасні атаки (DDOS-атаки –

distributed denial-of-service) для здійснення взлому системи можуть використовувати і проміжні комп'ютери, які прийнято називати зомбі (рис.2).

Такі системи у мережі Інтернет є незахищені або мало захищені. Зловмисник взломавши їх, бере під свій контроль і при цьому інсталює відповідне програмне забезпечення на кожному з них. Такі компютери після того стають підвладні йому.

Виходячи із відомих методів виявлення атак розглянутих у попередньому розділі, найкращим методом для вирішення задачі ідентифікації атак є застосування СМ на базі нейронних мереж. Вони описують кожну атаку у вигляді спеціальної моделі або сигнатури. Як сигнатура атаки можуть виступати: рядок символів, семантичний вираз на спеціальній мові, формальна математична модель. Алгоритм роботи СМ полягає в пошуку сигнатури атак в початкових даних, зібраних мережевими і хостовими датчиками системи. У разі виявлення шуканої сигнатури, система фіксує факт інформаційної атаки, яка відповідає знайденій сигнатурі.

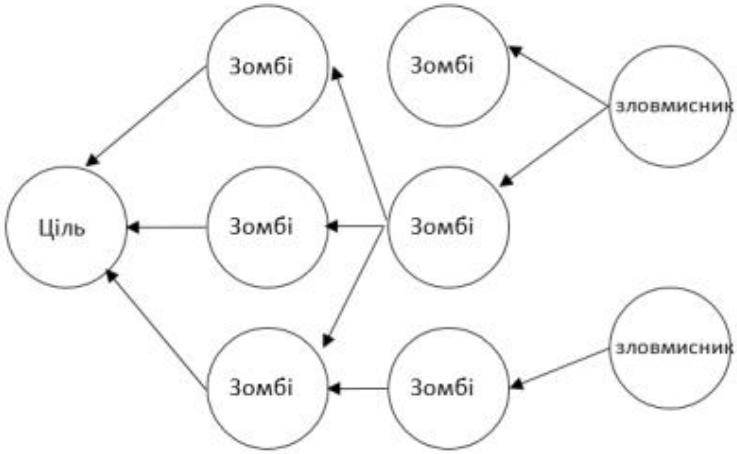


Рисунок 2. Здійснення DDOS-атаки

УДК 004.031.6

Л. Матійчук, кан. екон. наук, доцент, В. Сташук, І. Павлов
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ОЦІНКА ІСНУЮЧИХ СИСТЕМ ВИЯВЛЕННЯ АТАК

UDC 004.031.6

L. Matiychuk, PhD, Assoc. Prof., I. Pavlov, V. Stashuk

EVALUATION OF EXISTING ATTACK DETECTION SYSTEMS

На сьогоднішній день дуже інтенсивно розвиваються технології захисту корпоративних мереж, які включають в себе:

- ME (Firewall) – це програма або спеціалізована апаратна реалізація, що, ґрунтуючись на деяких правилах, дозволяє або забороняє передачу інформації, що проходить через неї, з метою обмеження деякої підмережі від зовнішнього доступу чи навпаки, для заборони виходу назовні. Міжмережеві екрани реалізують механізми контролю доступу із зовнішньої мережі до внутрішньої шляхом фільтрації всього вхідного і вихідного трафіку, пропускаючи тільки авторизовані дані. Всі міжмережеві екрани функціонують на основі інформації, яка отримується з різних рівнів еталонної моделі ISO/OSI, і чим вищий рівень OSI, на основі якого побудований ME, тим вищий рівень захисту, що ним забезпечується. Існують три основних типи міжмережевих екранів – пакетний фільтр (packet filtering), шлюз на сеансовому рівні (circuit-level gateway) і шлюз на прикладному рівні (application-level gateway). Існує дуже мало міжмережевих екранів, які можуть бути одночасно віднесені до одного з названих типів. Як правило, Firewall суміщає в собі функції двох або трьох типів.
- Найбільш очевидний недолік ME – неможливість захисту від користувачів, які знають ідентифікатор і пароль для доступу в сегмент корпоративної мережі, який захищається. ME може обмежити доступ до ресурсів, але він не може заборонити авторизованому користувачу скопіювати цінну інформацію або змінити які-небудь параметри. А по статистиці не менше ніж 70% всіх загроз безпеці надходить зі сторони співробітників організації.
- Віртуальна приватна мережа (VPN – Virtual Private Network). Технологія VPN призначена для побудови єдиного прозорого користувацького середовища поверх будь-якої транспортної мережі. Таке рішення дозволяє організувати: безпечний віддалений доступ персоналу до мережі підприємства чи організації з будь-якого робочого місця, підключеного до мережі Інтернет; достовірний підрахунок вживаних абонентом ресурсів в широкомовних транспортних мережах (наприклад, в мережах Ethernet); безпечну передачу конфіденційної інформації по мережі Інтернет без побудови додаткових фізичних каналів зв'язку. При побудові таких мереж можливо використовувати як комутовані канали зв'язку (dial-up), так і некомутовані (виділені лінії). При цьому для забезпечення конфіденційності інформації, що передається, не потребується організація додаткової виділеної лінії, а можливе використання вже існуючої, що значно знижує вартість побудови мереж VPN. Безпека інформації, що передається по мережі забезпечується шляхом шифрування з використанням будь-якого з наявних криптоалгоритмів.
- Сканер безпеки. Класичним сканером, який поставляється з усіма *nix подібними операційними системами є nmap. Програма призначена для сканування мереж з будь-якою кількістю об'єктів, визначаючи стан об'єктів мережі, що сканується а також портів і відповідних служб. Для цього nmap використовує багато різних методів сканування таких як UDP, TCP connect(), TCP SYN (напіввідкрите), FTP proxy (прорив через ftp), Reverse-ident, ICMP (ping), FIN, ACK, Xmas tree, SYN і NULL сканування.
- Одним з найновіших сканерів безпеки є Nessus. Nessus являє собою безплатний сучасний сканер безпеки локальних і віддалених систем. Початок Nessus Project було покладено в 1998 році, перший реліз вийшов в квітні. На той період найпоширенішим сканером безпеки був SATAN. Задачею Nessus являється визначення запущених служб і вразливостей, включаючи

найпопулярніші повідомленні про „дірки" wu-ftpd, наявність демонів DDOS, проблеми ipfw FreeBS і ін. Основний принцип полягає в тому, що вся інформації потребує перевірки, тобто інформація багерів основних служб не вважається основоположною.

- Систему виявлення вторгнень (IDS). Для запобігання комп'ютерним атакам, необхідно розробляти та налаштовувати системи захисту інформації та системи виявлення атак. Системи виявлення комп'ютерних атак – це один із найважливіших елементів систем інформаційної безпеки мереж. Враховуючи зростання в останні роки число проблем зв'язаних з комп'ютерною безпекою постійно зростає, як і пов'язаних з ними число хакерських атак (рис. 1). Системи виявлення вторгнень включають в себе: виявлення спроб несанкціонованого доступу та захист від атак типу „відмова в обслуговуванні" (DOS-атак).

Виявлення атак потребує виконання однієї із двох умов: розуміння очікуваної поведінки підконтрольного об'єкта системи або знання всіх можливих атак і їх модифікацій.

При створенні систем виявлення атак використовуються два основні підходи:

– виявлення аномальної поведінки, використовуючи апарат математичної статистики, який досить добре себе зарекомендував. Даний підхід використовується , як правило, при виявленні DoS-атак, які використовують посилку великої кількості трафіку за короткий інтервал часу [25];

– виявлення зловживань, використовуючи сигнатури, що описують послідовність байт і дій, які характеризують несанкціоновану діяльність. Цей підхід знайомий по антивірусних системах, які побудовані саме за цим принципом.

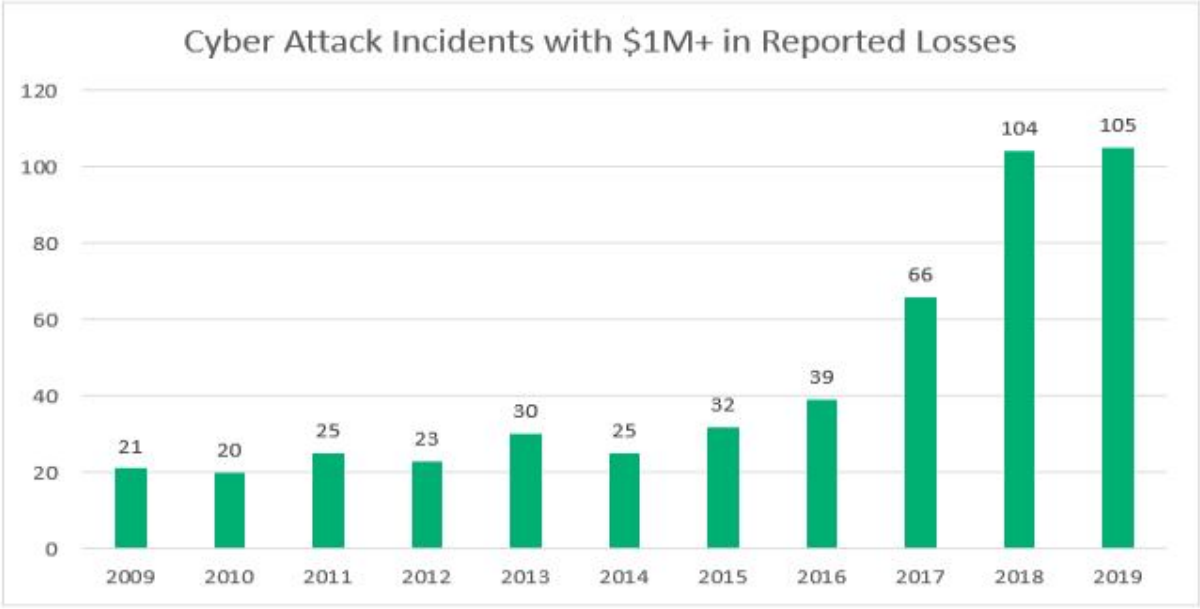
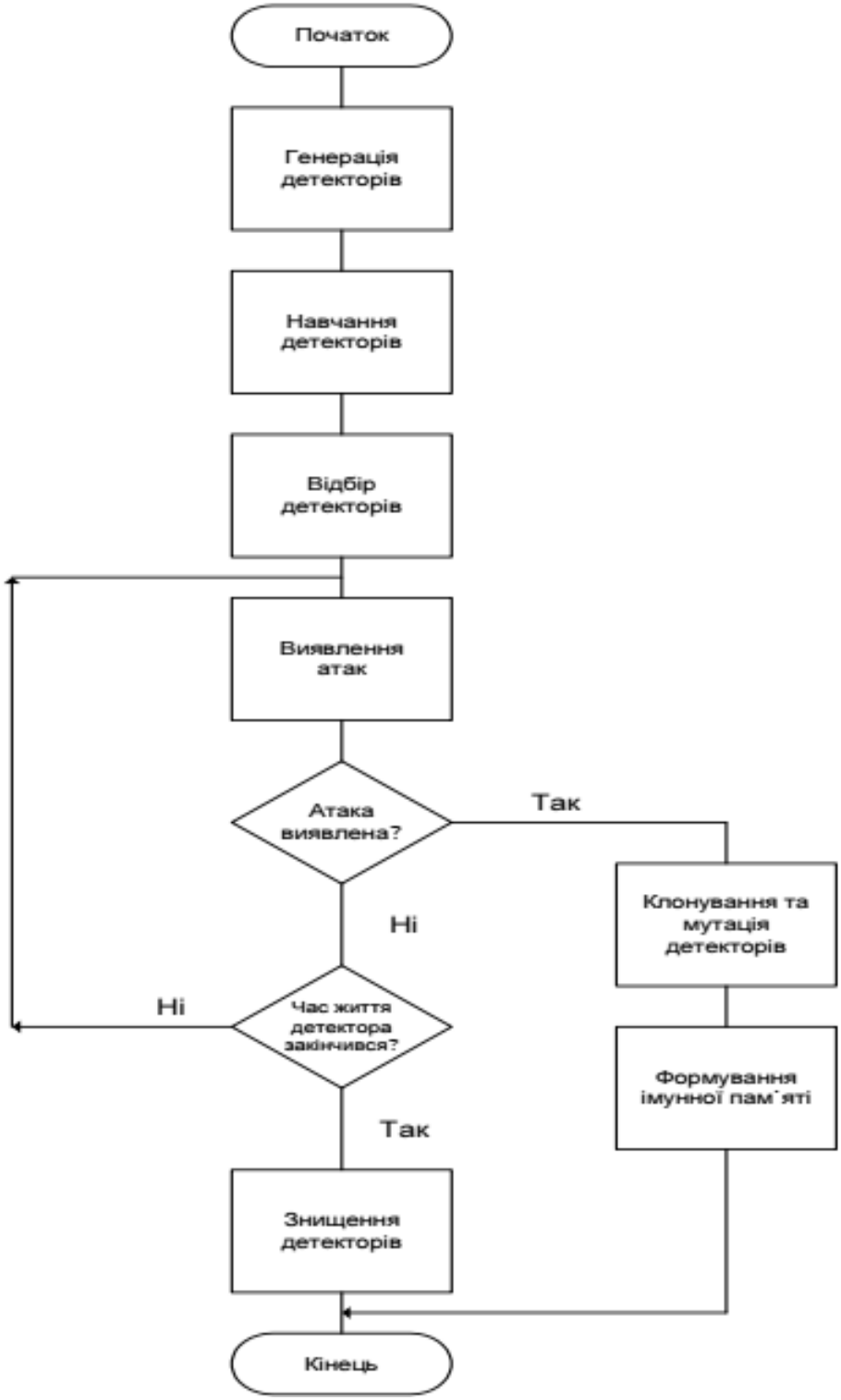


Рисунок 1. Кількість атак в мережі Інтернет

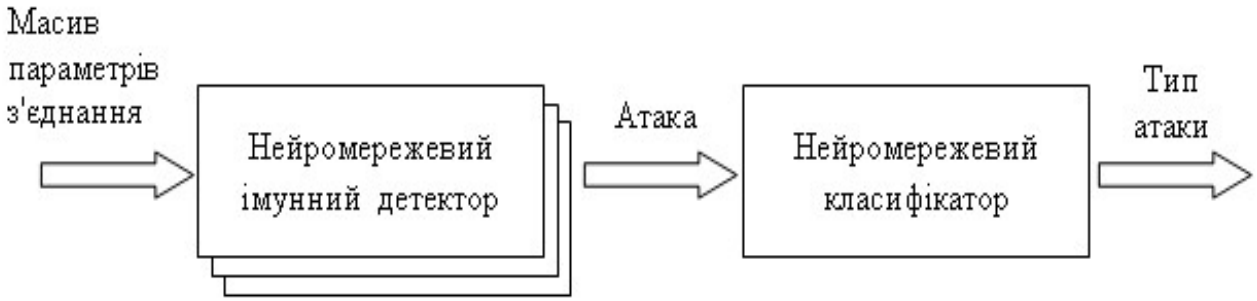
Додаток Б

Узагальнений алгоритм функціонування нейромережевої системи



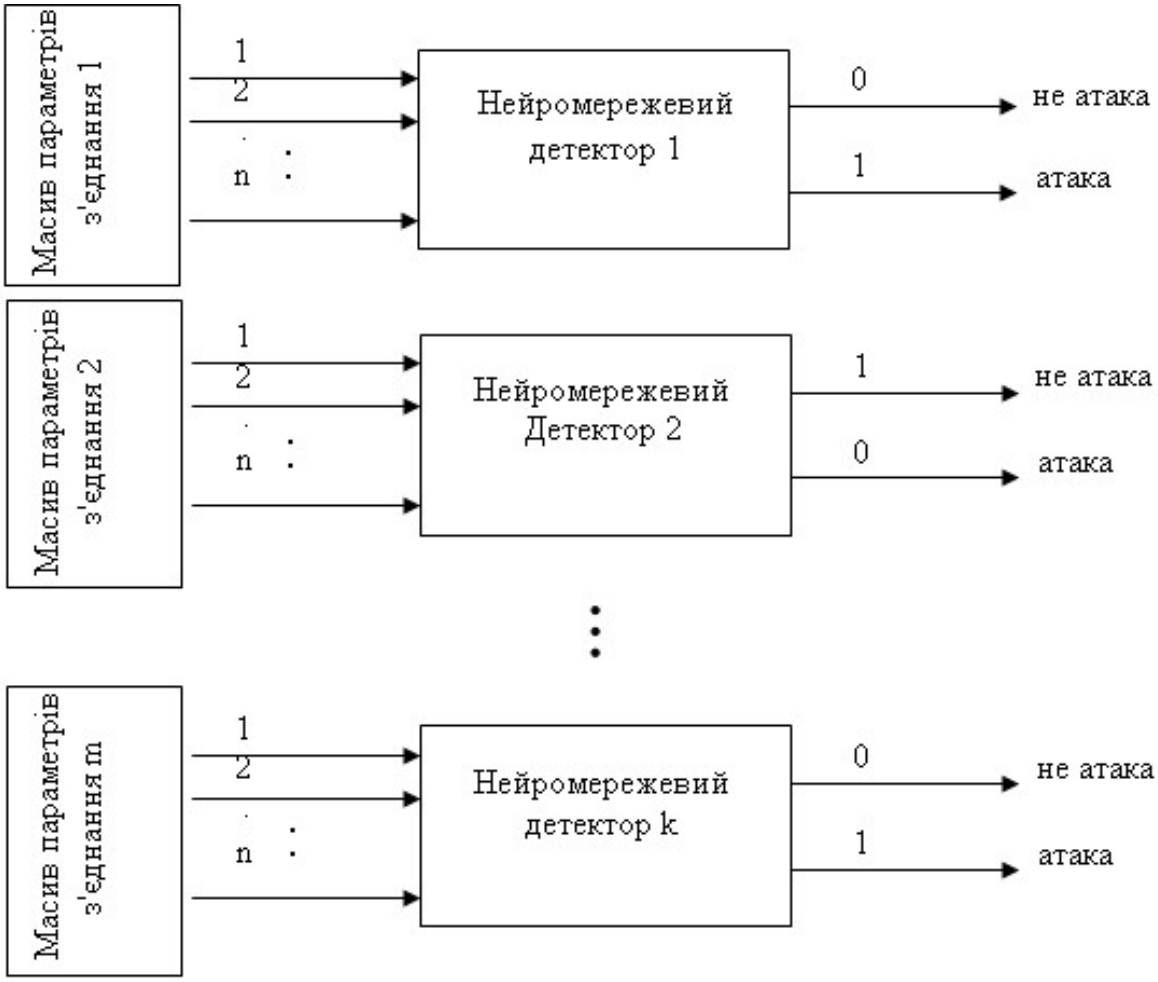
Додаток В

Нейромережевий модуль системи виявлення і класифікації комп’ютерних атак



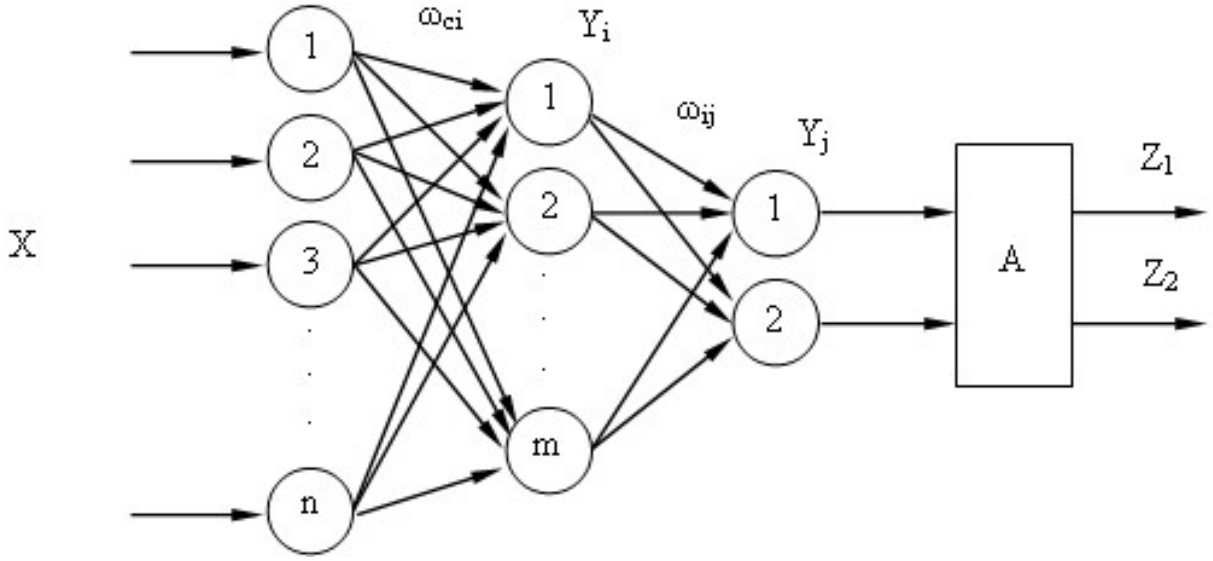
Додаток Г

Механізм функціонування імунних детекторів на основі нейронних мереж



Додаток Д

Архітектура нейромережевого імунного детектора



Додаток Е

Тестування програмного модуля

31 lines (29 sloc) | 1.18 KB

Raw Blame

```
function Z = multiLayerPerceptron(X, weights)
% multiLayerPerceptron   A multi-layer perceptron
%
%   Z = multiLayerPerceptron(X, weights) applies a multi-layer perceptron
%   to the input array X.
%
%   Inputs:
%       X           - A numFeatures-by-numInputSubwords input array.
%       weights     - The weights for the multi-layer perceptron stored in
%                     a struct. This includes:
%                       - mlp_c_fc_w_0: A weight matrix for the first fully
%                         connected layer.
%                       - mlp_c_fc_b_0: A bias vector for the first fully
%                         connected layer.
%                       - mlp_c_proj_w_0: A weight matrix for the second
%                         fully connected layer.
%                       - mlp_c_proj_b_0: A bias vector for the second
%                         fully connected layer.
%
%   Outputs:
%       Z           - A numFeatures-by-numInputSubwords output array.

Z = transformer.layer.convolution1d( X, ...
    weights.mlp_c_fc_w_0, ...
    weights.mlp_c_fc_b_0 );
Z = transformer.layer.gelu(Z);
Z = transformer.layer.convolution1d( Z, ...
    weights.mlp_c_proj_w_0, ...
    weights.mlp_c_proj_b_0 );

end
```

29 lines (25 sloc) | 785 Bytes

Raw Blame

```
function Z = normalization(X, g, b)
% normalization   Layer Normalization
%
%   Z = normalization(X, g, b) applies layer normalization to the input X.
%   Layer normzalization is described in [1].
%
%   Inputs:
%       X - A numFeatures-by-numInputSubwords-by-numObs input array.
%       g - A numFeatures-by-1 weight vector.
%       b - A numFeatures-by-1 bias vector.
%
%   Outputs:
%       Z - A numFeatures-by-numInputSubwords-by-numObs output array.
%
%   References:
%
%   [1] Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton, "Layer
%       Normalization", https://arxiv.org/abs/1607.06450

normalizationDimension = 1;

epsilon = single(1e-5);

U = mean(X, normalizationDimension);
S = mean((X-U).^2, normalizationDimension);
X = (X-U) ./ sqrt(S + epsilon);
Z = g.*X + b;

end
```