Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеню магістра

студентки *Кулікової Катерини Ігорівни*

академічної групи *125м-19-2*

спеціальності *125 Кібербезпека*

спеціалізації[1]

за освітньо-професійною програмою *Кібербезпека*

на тему *Метод виявлення зовнішніх комп'ютерних атак за допомогою моніторингу мережевих об'єктів*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Сафаров О.О.			
розділів:				
спеціальний	ас. Ковальова Ю.В.			
економічний	к.е.н., доц. Пілова Д.П.			

Рецензент	к.т.н., доц. Шидловський І.А.			

Нормоконтролер	ст. викл. Мєшков В.І.			

Дніпро
2020

ЗАТВЕРДЖЕНО:
завідувач кафедри
безпеки інформації та телекомунікацій
_____ д.т.н., проф. Корнієнко В.І.

«_____»_____20___року

ЗАВДАННЯ
на кваліфікаційну роботу ступеня бакалавра

студенту **Куліковій Катерині Ігорівні** **академічної групи** **125м-19-2**
(прізвище та ініціали) (шифр)

спеціальності ***125 Кібербезпека***

спеціалізації _____

за освітньо-професійною програмою ***Кібербезпека***

на тему «Метод виявлення зовнішніх комп'ютерних атак за допомогою моніторингу мережевих об'єктів»

Затверджену наказом ректора НТУ «Дніпровська політехніка» від 22.10.20 № 888-С

Розділ	**Зміст**	**Термін виконання**
Розділ 1	Аналітичний огляд СВА та порівняльний аналіз різних СВА.	
Розділ 2	Описана модель виявлення атак, архітектура СВА та алгоритм її роботи.	
Розділ 3	Досліджена ефективність запропонованого методу для експериментальної СВА.	
Розділ 4	Економічно обґрунтована розробка та впровадження методу виявлення атак.	

Завдання видано _____ _____
(підпис керівника) (прізвище, ініціали)
Дата видачі завдання: 02.09.2020 р.

Дата подання до екзаменаційної комісії: 11.12.2020 р.

Прийнято до виконання _____ _____
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 95 с., 5 рис., 11 табл., 7 додатки, 32 джерела.

Об’єкт досліджень: метод виявлення зовнішніх комп’ютерних атак за допомогою моніторингу мережевих об’єктів.

Метою кваліфікаційної роботи є: розробка методу і експериментальної системи виявлення атак на РІС на основі спостереження за поведінкою об'єктів в РІС, що дозволяє об'єднати переваги двох підходів - виявлення аномалій і виявлення зловживань - при поліпшенні показників ефективності та складності методів виявлення зловживань.

Предмет досліджень: розробка методу виявлення атак на розподілені інформаційні системи, на основі аналізу поведінки мережевих об'єктів в інформаційній системі.

В першому розділі роботи проаналізовано різновиди мережевих атак, принципи їх роботи, вразливості які вони використовують. Також проведено порівняльний аналіз різних СВА.

В другому розділі описана модель виявлення атак, її архітектура та алгоритми роботи.

В третьому розділі досліджена ефективність методу для експериментальної системи виявлення атак.

В економічному розділі була доведена економічна доцільність розробки та впровадження модуля для системи виявлення атак для компанії.

ІНФОРМАЦІЙНА БЕЗПЕКА, КОМПЛЕКС ВИЯВЛЕННЯ АТАК, КОМП’ЮТЕРНІ МЕРЕЖІ, ВИЯВЛЕННЯ МЕРЕЖЕВИХ АТАК.

РЕФЕРАТ

Пояснительная записка: 95 с., 5 рис., 11 табл., 7 приложения, 32 источника.

Объект исследований: Метод выявления внешних компьютерных атак с помощью мониторинга сетевых объектов.

Целью работы является Целью данной работы является разработка метода и экспериментальной системы обнаружения атак на РИС на основе наблюдения за поведением объектов в РИС, что позволяет объединить преимущества двух подходов - выявление аномалий и выявления злоупотреблений - при улучшении показателей эффективности и сложности методов выявления злоупотреблений.

Предмет исследований: разработка метода обнаружения атак на распределенные информационные системы на основе анализа поведения сетевых объектов в информационной системе.

В первом разделе работы проанализированы разновидности сетевых атак, принципы их работы, уязвимости, которые они используют. Также проведен сравнительный анализ различных СВА.

Во втором разделе описана модель обнаружения атак, ее архитектура и алгоритмы работы.

В третьем разделе исследована эффективность метода для экспериментальной системы обнаружения атак.

В экономическом разделе была доказана экономическая целесообразность разработки и внедрения модуля для системы обнаружения атак для компании.

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ, КОМПЛЕКС ОБНАРУЖЕНЯ АТАК, КОМПЬЮТЕРНЫЕ СЕТИ, ОБНАРУЖЕНИЕ СЕТЕВІХ АТАК.

ABSTRACT

Explanatory note: 95 pages, 5 figures, 11 tables, 7 appendices, 32 sources.

Object of research: a method of detecting external computer attacks by monitoring network objects.

The purpose of the thesis is: to develop a method and experimental system for detecting attacks on DIS based on monitoring the behavior of objects in DIS, which combines the advantages of two approaches - detecting anomalies and detecting abuse - while improving efficiency and complexity methods of detecting abuse.

Subject of research: development of a method for detecting attacks on distributed information systems, based on the analysis of the behavior of network objects in the information system.

The first section of the work analyzes the types of network attacks, the principles of their work, the vulnerabilities they use. A comparative analysis of different CBAs was also performed.

The second section describes the attack detection model, its architecture and algorithms.

The third section investigates the effectiveness of the method for an experimental system for detecting attacks.

The economic section proved the economic feasibility of developing and implementing a module for an attack detection system for the company.

INFORMATION SECURITY, ATTACK DETECTION COMPLEX, COMPUTER NETWORKS, NETWORK ATTACK DETECTION.

СПИСОК УМОВНИХ СКОРОЧЕНЬ

АС – автоматизована система;

ІД – інформаційна діяльність;

ІКС – інформаційно-комунікаційні системи;

ІТС – інформаційно-телекомунікаційна система;

КЗ – контрольована зона;

КЗЗ – комплекс засобів захисту;

КСЗІ – комплексна система захисту інформації;

ОІД – об’єкт інформаційної діяльності;

ОС – операційна система;

ПЗ – програмне забезпечення;

РІС – розподілена інформаційна система;

СВА – система виявлення атак

СЗІ – служба захисту інформації;

СУБД – система управління базами даних;

ТЗ – технічне завдання;

ТЗІ – технічний захист інформації;

DoS – denial of service;

FTP – file transfer protocol;

HTTP – hypertext transfer protocol;

IP – internet protocol;

OSI – open system interconnection;

SQL – structured query language.

TCP – transmission control protocol;

ЗМІСТ с.

ВСТУП

Інтенсивний розвиток Інтернету, повсюдний перехід на електронні форми зберігання і передачі інформації, активне впровадження в повсякденне життя електронних форм платежів і багато інших чинників сьогоднішньої реальності вплинули на те, що безпека мереж і мережевих сервісів стала дійсно нагальною проблемою практично всіх організацій. У невеликих організаціях зі слабо розвиненою інформаційною структурою ця проблема поки що малопомітна, і вирішується вона установкою антивірусного пакета, який, як правило, рідко оновлюється. Але, як показує практика, цього явно недостатньо. Важлива інформація для компанії може бути втрачена, викрадена зловмисниками, якщо її керівництво абияк ставиться до безпеки своєї корпоративної мережі.

Дані можуть бути модифіковані або знищені в наступних випадках:

- всередині локальної мережі співробітниками навмисно або ненавмисно;
- стороння особа проникне в локальну мережу ззовні;
- стороння особа перехопить інформацію в глобальній мережі на її шляху від одного підрозділу до іншого.

Всі перераховані вище випадки можуть завдати значної шкоди компанії. Тому продумана і добре організована система безпеки дозволяє уникнути або звести до мінімуму втрату важливих даних організації, тим самим виключивши додаткові витрати.

Тема кваліфікаційної роботи є актуальною, так як відбувається постійне збільшення інформаційних ресурсів, тому необхідно використовувати засоби захисту для виявлення загроз.

РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД СИСТЕМ ВАЯВЛЕННЯ АТАК

1.1. Завдання виявлення комп'ютерних атак

Дана робота присвячена розробці методу виявлення атак на розподілені інформаційні системи (РІС) на основі аналізу поведінки мережевих об'єктів в інформаційній системі. Суть завдання полягає в створенні моделі комп'ютерної атаки і методу автоматичного виявлення атаки на основі даної моделі. Метод дозволяє виявляти комп'ютерні атаки спостерігаючи за поведінкою об'єктів в РІС і їх взаємодією.

Потрібно розробити:

1. модель функціонування РІС, придатну для виявлення атак на РІС;
2. метод виявлення атак на основі запропонованої моделі.

1.2. Актуальність теми

Комп'ютерні мережі за кілька останніх десятиліть з чисто технічного рішення перетворилися на глобальне явище, розвиток якого впливає на більшість сфер економічної діяльності. Одним з перших кількісну оцінку значущості мереж дав Роберт Меткалф, який брав участь у створенні протоколу Ethernet: за його оцінкою «значимість» мережі у всіх сенсах пропорційна квадрату числа вузлів в ній.[1] Тобто, залежність від нормальної роботи мереж зростає швидше, ніж самі мережі. Забезпечення працездатності мережі і інформаційних систем які в ній функціонують залежить не тільки від надійності апаратури, а й, найчастіше, від здатності мережі протистояти цілеспрямованим діям, які спрямовані на порушення її роботи.

Створення інформаційних систем, стійких до шкідливих впливів і комп'ютерних атак, пов'язане з істотними витратами як часу, так і матеріальних ресурсів. Крім того, існує відома зворотна залежність між зручністю користування системою і її захищеністю: чим досконаліша система захисту, тим складніше користуватися основним функціоналом інформаційної системи. У

2000-і роки, в рамках оборонних проектів США, робилися спроби створення розподілених інформаційних систем спеціального призначення (MMS - Military Messaging System), для яких доводилася реальність реалізації основної теореми безпеки - невиведення системи із стану захищеності для будь-яких послідовних дій інформаційних об'єктів в системі. У цих системах використовувалося спеціалізоване програмне забезпечення на всіх рівнях, включаючи системний. Однак, на сьогоднішній день подібні системи не отримали розвитку, і для організації інформаційних систем використовуються операційні системи загального призначення, такі як ОС сімейства Microsoft Windows, GNU / Linux, *BSD і різні клони UNIX (Solaris, HP-UX, etc).

За минулі роки в рамках академічних розробок були створені сотні систем виявлення атак для різних платформ: від систем класу mainframe до сучасних операційних систем загального призначення, СУБД і додатків.

При створення ефективних систем захисту інформаційних систем імплементори стикаються також з нестачею обчислювальної потужності. З самого початку розвитку комп'ютерів і комп'ютерних мереж спостерігаються дві тенденції:

1. Закон Мура;
2. Закон Гілдера.

Закон Мура говорить про щорічне подвоєння продуктивності обчислювальних елементів, доступних за одну і ту ж вартість, а закон Гілдера - про потроєння пропускної здатності каналів зв'язку за такий же період. Таким чином, зростання обчислювальної потужності вузлів мережі відстає від зростання обсягів переданої в мережі інформації, що з кожним роком посилює вимоги до обчислювальної складності алгоритмів систем захисту інформації.[2]

Методи виявлення атак в сучасних системах виявлення атак (СВА) недостатньо опрацьовані в частині формальної моделі атаки, і, отже, для них досить складно чітко оцінити такі властивості як обчислювальна складність, коректність і т.д. Прийнято розділяти методи виявлення атак на методи виявлення аномалій і методи виявлення зловживань. До другого типу методів відносяться

більшість сучасних комерційних систем (Cisco IPS, ISS RealSecure, NFR) - вони використовують сигнатурні (експертні) методи виявлення.[3] Існує безліч академічних розробок в області виявлення аномалій, але в промислових системах вони використовуються рідко і з великою обережністю, так як такі системи породжують велику кількість помилкових спрацьовувань. Для експертних систем основною проблемою є низька, близька до нуля, ефективність виявлення невідомих атак (адаптивність). Низька адаптивність до сих пір залишається проблемою, хоча такі переваги як низька обчислювальна складність і низька ціна розгортання визначають домінування таких систем в даній області.

1.3.Мета роботи

Метою даної роботи є розробка методу і експериментальної системи виявлення атак на PIC на основі спостереження за поведінкою об'єктів в PIC, що дозволяє об'єднати переваги двох підходів - виявлення аномалій і виявлення зловживань - при непогіршенні показників ефективності та складності методів виявлення зловживань.

При виконанні роботи були поставлені такі завдання:

1. Побудувати модель функціонування PIC, в рамках якої визначити таке явище як атака;

2. Виділити класи реальних атак, які представлені в рамках запропонованої моделі функціонування PIC;

3. Розробити метод виявлення атак на PIC на основі інформації про поведінку об'єктів PIC;

4. Розробити архітектуру системи виявлення атак для PIC на базі ОС GNU/Linux, Windows 10, Windows Server і мережевого стека TCP / IP.

1.4. Методи вирішення поставленої задачі

У роботі запропонована модель функціонування PIC в умовах впливу комп'ютерних атак в формі системи переходів. Ця модель має такі особливості:

1. Функціонування PIC визначається через поняття стану об'єкта PIC і переходи між станами, всі об'єкти типізовані;

2. Сукупність станів розділяється на безпечні та небезпечні стану;

3. Вводиться поняття траєкторії і поведінки об'єкта;

4. Поняття атаки вводиться як траєкторія з безпечного стану деякого об'єкта в небезпечне;

5. Для кожного класу атак вводиться поняття автомата першого роду, який приймає будь-яку траєкторію даного класу;

6. Для кожного класу об'єктів вводиться поняття автомата другого роду, який приймає будь-яку траєкторію даного об'єкта і дозволяє розділити безліч траєкторій на два класи - нормальних і аномальних. Запропоновано мову опису поведінки об'єктів PIC, що дозволяє описувати стан об'єктів PIC і переходи між ними. В основу мови покладено формалізм кінцевих автоматів, у яких переходи між станами типізовані, а стан визначається логічним предикатом.

1.5 Мета огляду

Метою огляду є дослідити ефективність доступних в даний час СВА і визначити основні недоліки використовуваних в них методів виявлення атак. Основною проблемою складання подібного огляду є те, що безліч доступних реалізацій СВА представлено, в основному, комерційними системами (такими як Cisco IPS, Juniper NetScreen, ISS RealSecure, NFR і т.д.), для яких відсутня відкрита інформація про програмну архітектуру і методи виявлення атак які використовуються в них. Доступна інформація по подібним системам носить маркетинговий характер, що ускладнює проведення порівняльного аналізу з публікацій у літературі. З цієї причини безліч розглянутих систем буде обмежено СВА з відкритим вихідним кодом, доступним публічно.

В результаті огляду буде показано, що:

1. Більшість сучасних СВА використовують на базовому рівні ту чи іншу реалізацію сигнатурного методу виявлення (pattern matching, порівняння

шаблонів). Реалізації відрізняються один від одного рівнем розгляду системи, алфавітом сигнатур і використовуваним «рушієм» - від простого пошуку підстроки до повноцінної реалізації регулярних виразів над заданим алфавітом.

2. Сукупність існуючих методів виявлення атак набагато ширше, але їх використання в системах має принципові обмеження, пов'язані з вимогами верифікованості, стійкості і відтворюваності результату, а також великим числом помилок другого роду (помилкових спрацьовувань). Використання таких методів обмежене експериментальними академічними розробками.

3. Доступні реалізації СВА нестійкі до модифікацій атак і не можуть автоматично адаптуватися до появи нових атак. При цьому використання методів виявлення аномалій обмежене з причин, перерахованих в п.2.

1.6 Критерії порівняння

При розгляданні методів і систем виявлення комп'ютерних атак використовуються дві групи критеріїв:

1. Перша група характеризує власне методи виявлення атак і специфічні для них якісні і кількісні показники ефективності;

2. Друга група критеріїв характеризує реалізації цих методів в системах виявлення атак.

Для порівняльного аналізу методів виявлення атак обрано такі критерії:

Рівень спостереження за системою:

Цей критерій визначає рівень абстракції аналізованих подій в системі, що захищається і визначає межі застосування методу для виявлення атак в мережах. В рамках даного огляду розглядаються такі рівні:

1. HIDS - спостереження на рівні операційної системи окремого вузла мережі;

2. NIDS - спостереження на рівні мережевої взаємодії об'єктів на вузлах мережі;

3. AIDS - спостереження на рівні окремих додатків вузла мережі;

4. Hybrid - комбінація спостерігачів різних рівнів.

Верифікованість методу:

Цей критерій дозволяє оцінити, чи може людина (наприклад, кваліфікований оператор СВА або експерт) відтворити послідовність кроків щодо прийняття рішення про наявність атаки, зіставляючи вхідні і вихідні дані СВА. Наприклад, сигнатурні методи вважатимемо верифіковані, а кластерні - ні. Верифікованість дозволяє провести експертну оцінку коректності методу і його реалізації в довільний момент часу, в тому числі в процесі експлуатації системи виявлення. Властивість верифікованості методу важлива при експлуатації системи виявлення атак в реальній обстановці в якості засобу збору доказової бази про атаки. Можливі значення: висока (+), низька (-).[4]

Адаптивність методу:

Оцінка стійкості методу до малих змін реалізації атаки, які не змінюють результат атаки. Адаптивність є єдиною суттєвою перевагою «альтернативних» методів виявлення атак перед «сигнатурними». Відсутність адаптивності не дозволяє системі захисту оперативно реагувати на невідомі атаки і вимагає організації системи регулярного оновлення баз відомих атак, за аналогією з антивірусними системами. Можливі значення: висока (+), низька (-).

Стійкість:

Цей критерій характеризує незалежність виходу методу від системи що захищається - для одного і того ж входу метод повинен давати один і той же вихід, незалежно від системи що захищається. Проблема стійкості особливо гостро стоїть для статистичних методів, які аналізують абсолютні значення параметрів продуктивності і завантаженості ресурсів мережі і вузлів, які можуть істотно відрізнятися на різних вузлах і в різних мережах. Навчений в одній мережі розпізнавач може бути стійким в межах даної мережі і нестійким у всіх інших мережах. Таку стійкість будемо називати локальною. Так як процедура навчання зазвичай є «дорогою» - вимагає використання великої кількості ресурсів і часу -

число процедур навчання бажано мінімізувати. Методи виявлення атак, що аналізують семантику введення, більш стійкі, ніж статистичні. Можливі значення: глобальна (+), локальна (-).

Обчислювальна складність:

Теоретична оцінка складності методу на основі інформації з публікацій. В огляді розглядається тільки складність методу в режимі виявлення, без урахування можливих попередніх етапів налаштування та навчання. Цей критерій є ключовим для завдання виявлення атак в мережах і має набагато більше значення, ніж складність з пам'яттю через випереджальне зростання пропускної здатності каналів передачі даних і здешевлення машинної пам'яті.[5]

1.6.1. Порівняння систем виявлення атак

Для порівняльного аналізу СВА було обрано такі критерії:

Клас атак, що виявляються.

Цей критерій визначає, які класи атак здатна виявити розглянута система. Це один з ключових критеріїв. У зв'язку з тим, що на сьогоднішній день жодна система не здатна виявити атаки всіх класів, для більш повного покриття всього спектра атак необхідно комбінувати різні СВА. Тут ми використовуємо класифікацію атак, засновану на поділі ресурсів системи що захищається за типами.

Клас атаки - це четвірка <L,R,A,D>, де

L - розташування атакуючого об'єкта,

R - ресурс що атакується,

A - цільовий вплив на ресурс,

D - ознака розподіленого характеру атаки.

*L:*розташування атакуючого об'єкта. Воно може бути або внутрішнім по відношенню до інформації, що захищається системі (li), або зовнішнім(le).

R:ресурс, що атакується. Ресурси поділяються по розташуванню і по типу.

За розташуванням:

1. вузлові (rl),
2. мережеві (rn).

За типом:

1. призначені для користувача ресурси (ru),
2. системні ресурси (rs),
3. ресурси СУБД (rd),
4. обчислювальні ресурси (rc),
5. ресурси захисту (rp).

Цільовий вплив на ресурс(A):

1. збір інформації (as),
2. отримання прав користувача ресурсу (au),
3. отримання прав адміністратора ресурсу (ar),
4. порушення цілісності ресурсу (ai),
5. порушення працездатності ресурсу (ad).

Ознака розподіленого характеру атаки(D):

1. розподілені (dd),
2. нерозподілені (dn).

Наступний критерій характеризує джерела та способи збору інформації про поведінку об'єктів і стан ресурсів:

Рівень спостереження за системою

Визначає, на якому рівні системи збирають дані для виявлення атаки. Розрізняються вузлові і мережеві джерела. В межах вузлових джерел поділяються рівні ядра і додатки. Від рівня спостереження за системою залежить швидкість збору інформації, вплив системи на інформацію що збирається, ймовірність отримання спотвореної інформації. Слід зазначити, що використання методу виявлення, що дозволяє аналізувати поведінку на всіх рівнях абстракції, не означає, що ця можливість реалізована у конкретній системі. Найчастіше

реалізація володіє меншими можливостями, ніж теоретичні можливості використовуваного нею методу.[6]

- HIDS - спостереження на рівні операційної системи окремого вузла мережі;
- NIDS - спостереження на рівні мережного взаємодії об'єктів на вузлах мережі;
- AIDS - спостереження на рівні окремих додатків вузла мережі;
- Hybrid - комбінація спостерігачів різних рівнів.

Наступний критерій визначає ефективність виявлення атаки на основі аналізу отриманої інформації.

Використовуваний метод виявлення

Метод виявлення також є ключовим критерієм порівняння. Існує два класи методів:

1. методи виявлення аномалій,
2. методи виявлення зловживань.

У наведеному нижче списку перераховані не окремі методи, а, в основному, сімейства методів, об'єднаних деякими єдиним підходом або теоретичною моделлю.

1. Виявлення зловживань
    a. Аналіз систем станів
    b. Графи атак
    c. Нейронні мережі
    d. Імунні мережі
    e. SVM
    f. Експертні системи
    g. Методи, засновані на специфікаціях
    h. MARS - Multivariate Adaptive Regression Splines
    i. Сигнатурні методи
2. Виявлення аномалій

a. Статистичний аналіз
b. Кластерний аналіз (data mining)
c. Нейронні мережі
d. Імунні мережі
e. Експертні системи
f. Поведінкова біометрія
g. SVM
h. Аналіз систем станів

Адаптивність до невідомим атак

Визначає здатність використовуваного методу виявляти раніше невідомі атаки. Наступні три критерії визначають такі архітектурні особливості СВА як управління і розподіленість.

Масштабованість

Визначає можливість додавання нових аналізованих ресурсів мережі, нових вузлів і каналів передачі даних, в тому числі можливість управління єдиної розподіленої системою виявлення атак. Управління може бути централізоване та/або розподілене. Додатково може бути присутня можливість віддаленого управління СВА. Сюди включаються завдання установки, налаштування і адміністрування системи. При повністю розподіленому управлінні необхідно управляти всіма компонентами СВА окремо. При повністю централізованому управлінні всі компоненти СВА можуть управлятися з одного вузла. Оптимальною є організація управління з централізованою схемою, в якій може бути кілька центрів, і вони можуть динамічно змінюватися.[7]

Відкритість

Визначає наскільки система є відкритою для інтеграції в неї інших методів виявлення атак, компонентів сторонніх розробників і сполучення її з іншими системами захисту інформації. Це можуть бути програмні інтерфейси для

вбудовування додаткових модулів і/або реалізація стандартів взаємодії мережевих компонентів.

Формування відповідної реакції на атаку

Визначає наявність в системі вбудованих механізмів відповідної реакції на атаку, крім самого факту її реєстрації. Прикладами реакції можуть бути втрати з'єднання з атакуючим об'єктом, блокування його на міжмережевому екрані, відстеження шляху проникнення атакуючого об'єкта в систему і т.д. Детально це питання розглянуто нижче.

Захищеність

Визначає ступінь захищеності СВА від атак на її компоненти, включаючи захист інформації, що передається, стійкість до часткового виходу компонентів з ладу або їх компрометації. Зачіпаються такі питання, як наявність вразливостей в компонентах СВА, захищеність каналів передачі даних між ними, а також авторизація компонентів всередині СВА[8].

Таким чином, «ідеальна» система виявлення атак має такі властивості:

1. покриває всі класи атак (система повна);

2. дозволяє аналізувати поведінку об'єктів в РІС на всіх рівнях: мережевому, вузловому і рівні окремих додатків;

3. адаптивна до невідомих атак (використовує адаптивний метод виявлення атак);

4. масштабується для РІС різних класів: від невеликих локальних мереж класу «домашній офіс» до великих багатосегментних і комутованих корпоративних мереж, забезпечуючи можливість централізованого управління всіма компонентами СВА;

5. є відкритою;

6. має вбудовані механізми реагування на атаки;

7. є захищеною від атак на компоненти СВА, в тому числі від перехоплення управління або атаки «відмова в обслуговуванні».

1.7. Методи виявлення атак

Всі методи виявлення атак можна розділити на два класи:

1. методи виявлення аномалій,

2. методи виявлення зловживань.

Методи першого класу базуються на наявності готового опису нормальної поведінки спостережуваних об'єктів в РІС, і будь-яке відхилення від нормальної поведінки вважається аномальним (порушенням).

Методи виявлення зловживань засновані на описі відомих порушень або атак: якщо поведінка деякого об'єкту в РІС збігається з описом відомої атаки, поведінка об'єкта вважається атакою.

1.7.1. Методи виявлення зловживань

Аналіз систем станів

У даній групі методів функціонування системи представляється через безліч станів і безліч переходів між ними, тобто у вигляді орієнтованого графа (як правило, нескінченного). Суть методу виявлення атак полягає в тому, що частина шляхів в такому графі позначаються як неприпустимі; кінцевий стан кожного такого шляху вважається небезпечним для системи. Процес виявлення атаки є побудова частини графа станів системи і переходів між ними, і пошук в отриманому графі відомих неприпустимих шляхів. Виявлення послідовності переходів, що приводить в небезпечний стан, означає успішне виявлення атаки. Відповідно до введених критеріїв, даний метод є гібридним з точки зору рівня спостереження за системою, верифікованим, стійким, має низьку обчислювальну складність (лінійну щодо довжини траси спостережуваних переходів і числа станів), але не є адаптивним.

Графи сценаріїв атак

В роботі запропонований підхід до виявлення атак на основі використання методів формальної специфікації на моделях. На вхід системи верифікації подається кінцева модель системи що захищається і деяке формальне свойство коректності, яке виконується тільки для дозволеної поведінки системи. Дана властивість коректності ділить всю сукупність поводжень на два класи - допустимої поведінки, для якого властивість виконується, і неприпустимого, для якого воно не виконується. Відмінність даного методу від звичайних систем верифікації полягає в тому, що їх завдання, зазвичай, знайти один контрприклад з безлічі неприпустимого поведінки, а в запропонованому методі будується повний набір таких прикладів для конкретної системи, що дає на виході опис можливих шляхів атаки. Через високу обчислювальну складність (NP) даний метод може бути використаний для пошуку вразливостей проектування систем і інших складних для виявлення вразливостей, але для завдання виявлення атак в реальному часі він непридатний[9].

За іншими критеріями метод є гібридним, верифіковані, стійким і адаптивним.

Нейронні мережі

Так як завдання виявлення атак можна розглядати як задачу розпізнавання образів (або завдання класифікації), то для її вирішення також застосовуються нейронні мережі.

Для цього функціонування системи і взаємодіючих з нею зовнішніх об'єктів представляється у вигляді траєкторій в деякому числовому просторі ознак. В якості методу виявлення зловживань, нейронні мережі навчаються на прикладах атак кожного класу і, в подальшому, використовуються для розпізнавання приналежності спостережуваної поведінки одному з класів атак. Основна складність у використанні нейромереж полягає в коректній побудові такого простору ознак, що дозволило б розділити класи атак між собою і відокремити їх від нормальної поведінки. Крім того, для класичних нейронних мереж характерно довгий час навчання, при цьому час навчання залежить від розміру навчальної

вибірки. Відповідно до введених критерій, нейронні мережі використовуються на мережевому і вузловому рівнях, є адаптивними, мають порівняно низьку обчислювальну складність. При цьому вони не є верифікованими і стійкими, як правило, тільки в межах тієї мережі, в якій вони навчалися, що істотно обмежує можливість застосування методу (тільки локальна стійкість).

Імунні мережі

Так як і нейронні, імунні мережі є механізмом класифікації і будуються за аналогією з імунною системою живого організму. Основна перевага імунних мереж полягати в можливості отримання «антитіл» до невідомих атак. В роботі було запропонована модель формального пептиду, для якої заявлена можливість використання в системах виявлення атак. Однак, пізніше було показано, що використання даного методу вимагає рішення системи диференціальних рівнянь в режимі виявлення, що дає обчислювальну складність порядку O(n^3) при використанні методу Рунге-Кутта. Відповідно до введених критерій, дана група методів може бути застосована для мережевого і вузлового рівнів, вона не є верифікованою, адаптивна, стійка тільки локально, має високу обчислювальну складність[10].

Support vector machines (SVM)

SVM - це спосіб розпізнання шаблонів, який дозволяє формувати шаблони в результаті навчання. Даний метод вимагає невеликої кількості даних для навчання і дозволяє обробляти вектори ознак великої розмірності, що корисно для підвищення точності систем виявлення атак і зниження тимчасових витрат на навчання і перенавчання. Метод застосуємо як для виявлення зловживань, так і для виявлення аномалій[11].

SVM має такі ж переваги і недоліки для вирішення нашої задачі, як і нейронні мережі, тобто є адаптивним, але неверіфікованим.

Експертні системи

Використання експертних систем для виявлення атак засноване на описі функціонування системи у вигляді безлічі фактів і правил, в тому числі для атак. На вхід експертна система отримує дані про спостережувані події в системі у вигляді фактів. На підставі фактів і правил, система робить висновок про наявність чи відсутність атаки.

Дана група методів задовольняє практично всім критеріям (верифіковані, адаптивна, стійка), але в загальному випадку має дуже велику обчислювальну складність, так як для неї може спостерігатися явище «комбінаторного вибуху» і повного перебору великого числа альтернатив.

Методи, засновані на специфікаціях

В основі цього методу лежить опис обмежень на заборонене поводження об'єктів в системі, що захищається, в вигляді специфікацій атак. У специфікацію може входити: обмеження на завантаження ресурсів, на список заборонених операцій і їх послідовностей, на час доби, протягом якого застосовуються ті чи інші обмеження. Відповідність поведінки специфікації вважається атакою. Специфікації використовуються для мережевого рівня, є верифікованими, локально стійким і мають низьку обчислювальну складність. Даний підхід близький до класу методів виявлення аномалій. Основні недоліки - низька адаптивність і складність розробки специфікацій.

Multivariate Adaptive Regression Splines (MARS)

Один з методів апроксимації функцій, заснований на сплайнах. Аналогічно нейронним мережам і кластерному аналізу MARS оперує в багатовимірному просторі ознак. Поведінка мережевих об'єктів відображається в послідовності векторів даного простору. Завдання процедури MARS полягає в побудові оптимальної апроксимації поведінки по заданій історії у вигляді навчальної множини векторів, при цьому в якості апроксимуючої функції використовуються сплайни зі змінним числом вершин. В ході «навчання», за допомогою переборного процесу, вибирається оптимальне число вершин для заданої вибірки.

Побудований сплайн є «шаблоном» атаки. У режимі розпізнавання, поведінка що спостерігається відображається в параметричний простір і порівнюється з апроксимуючої функцією. Переваги та недоліки даного методу аналогічні SVM і нейронних мереж.[12]

Сигнатурні методи

Найбільш часто використовувана група методів, суть яких полягає в складанні деякого алфавіту зі спостережуваних в системі подій і описі безлічі сигнатур атак у вигляді регулярних виразів (в загальному випадку) в побудованому алфавіті. Як правило, сигнатурні методи працюють на найнижчому рівні абстракції і аналізують безпосередньо дані які передаються по мережі, параметри системних викликів і записи файлів журналів. У найбільш розвиненому вигляді являє собою реалізацію регулярних виразів над різними трасами (мережевий трафік, системні виклики, в журналах додатків і т.п.). Сигнатурні методи примітні тим, що для них добре застосовні апаратні прискорювачі, але при цьому метод не є адаптивним. За іншими критеріями дана група методів є гібридної, глобально стійкою.

1.7.2. Методи виявлення аномалій

Статистичний аналіз

Дана група методів заснована на побудові статистичного профілю поведінки системи протягом деякого періоду «навчання», при якому поведінка системи вважається нормальною. Для кожного параметра функціонування системи будується інтервал допустимих значень, з використанням деякого відомого закону розподілу. Далі, в режимі виявлення, система оцінює відхилення спостережуваних значень від значень, отриманих під час навчання. Якщо відхилення перевищують деякі задані значення, то фіксується факт аномалії (атаки).

Для статистичного аналізу характерний високий рівень помилкових спрацьовувань при використанні в локальних мережах, де поведінка об'єктів не має гладкого, усередненого характеру. Крім того, даний метод стійкий тільки в межах конкретної системи.

Кластерний аналіз

Суть даної групи методів полягає в розбитті безлічі спостережуваних векторів-властивостей системи на кластери, серед яких виділяють кластери нормальної поведінки. У кожному конкретному методі кластерного аналізу використовується своя метрика, яка дозволяє оцінювати приналежність спостережуваного вектора властивостей системи одному з кластерів або вихід за межі відомих кластерів. Кластерний аналіз є адаптивним, але не верифікованим і стійким в межах конкретної системи, в якій збиралися дані для побудови кластерів.

Нейронні мережі

Нейронні мережі для виявлення аномалій навчаються протягом деякого періоду часу, коли вся поведінка вважається нормальною. Після навчання нейронна мережа запускається в режимі розпізнавання. У ситуації, коли у вхідному потоці трафіку інформації не вдається розпізнати нормальну поведінку, фіксується факт атаки. У разі використання репрезентативної навчальної вибірки нейронні мережі дають хорошу стійкість в межах заданої системи; але складання подібної вибірки є серйозною і складним завданням. Класичні нейронні мережі мають високу обчислювальну складність навчання, що ускладнює їх застосування на великих потоках даних.

Імунні мережі

Виявлення аномалій є одним з можливих додатків імунних методів. Так як кількість прикладів нормальної поведінки зазвичай на порядки перевищує число

прикладів атак, використання імунних мереж для виявлення аномалій має велику обчислювальну складність.

Експертні системи

Інформація про нормальну поведінку представляється в подібних системах у вигляді правил, а спостерігається поведінка у вигляді фактів. На підставі фактів і правил приймається рішення про відповідність спостережуваної поведінки «нормальній», або про наявність аномалії. Головний недолік подібних систем - висока обчислювальна складність (в загальному випадку). В тому числі при виявленні аномалій.[13] Натомість є ряд переваг, таких як висока стійкість, адаптивність та верифікованість.

Поведінкова біометрія

Включає в себе методи, які не потребують спеціального обладнання (сканерів сітківки, відбитків пальців, сканерів 3д моделі обличчя), тобто методи виявлення атак, засновані на спостереженні клавіатурного почерку і використання миші. В основі методів лежить гіпотеза про відмінність «почерку» роботи з інтерфейсами введення-виведення для різних користувачів. На базі побудованого профілю нормальної поведінки для даного користувача виявляються відхилення від цього профілю, викликані спробами інших осіб працювати з клавіатурою або іншими фізичними пристроями введення. Поведінкова біометрія має строго локальну стійкість (в межах однієї мережі) і слабо верифікована, однак має високу адаптивність.

Support vector machines (SVM)

Support vector machines застосовний як для виявлення зловживань, так і для виявлення аномалій, при цьому метод має переваги і недоліки, аналогічні нейронним мережам, наприклад у ситуації, коли у вхідному потоці трафіку інформації не вдається розпізнати нормальну поведінку, фіксується факт атаки. SVM дає хорошу стійкість в межах заданої системи. Даний метод виявлення аномалій має високу обчислювальну складність, але й гарну адаптивність. Також

слід зазначити, що support vector machines відноситься до методів з високою верифіікованістю.

Результати порівняння різних методів виявлення аномалій та зловживань наведені у табл. 1.1.

Таблиця 1.1 – Результати порівняння методів виявлення атак

Критерій / Метод	Рівень спостереження	Аномалії/ Зловживання	Верифікованість	Адаптивність	Стійкість	Обчислювальна складність
Системи переходів	Hybrid	-/+	+	-	+	O(n)
Графи атак	Hybrid	-/+	+	+	+	NP
Нейронні мережі	NIDS, HIDS	+/+	-	+	-	O(n) і вище
Імунні мережі	NIDS, HIDS	+/+	-	+	-	O(n) і вище
SVM	NIDS, HIDS	+/+	-	+	-	ln(n)
Експертні системи	NIDS, HIDS	+/+	+	+	+	У загальному випадку NP
Специфікації	NIDS, HIDS	-/+	+	-	-	ln(n)
MARS	NIDS	-/+	-	+	-	O(n) і вище
Сигнатурні методи	Hybrid	-/+	+	-	+	ln(n)
Статистичні методи	NIDS, HIDS	+/-	-	+	-	O(n) і вище
Кластерний аналіз	Hybrid	+/+	-	+	-	O(n) і вище
Поведінкова біометрія	HIDS	+/-	-	+	-	O(n) і вище

Таким чином, аналіз показує, що для більшості методів виявлення аномалій характерна слабка верифікованість і слабка глобальна стійкість (або її відсутність). Основна перевага методів виявлення аномалій полягає в їх адаптивності та здатності виявляти раніше невідомі атаки.

Серед глобально стійких і верифікованих методів, що мають при цьому низьку обчислювальну складність, можна відзначити метод аналізу системи

переходів і простий сигнатурний метод. Жоден з розглянутих методів не володіє одночасно адаптивністю, стійкістю і верифікованістю, маючи при цьому прийнятну обчислювальну складність.

1.8. Сучасні відкриті системи виявлення атак

В даному розділі розглянуті доступні на сьогоднішній день системи виявлення атак з відкритим вихідним кодом.

Всього розглянуто 5 систем виявлення атак. У табл. 1.2. наведена коротка інформація по кожній з них.

Таблиця 1.2 – Відкриті системи виявлення комп'ютерних атак

Назва системи	Розробник
Bro	University of California, Lawrence Berkeley National Laboratory
OSSEC	Daniel B. Sid , OSSEC.net
STAT	University of California at Santa Barbara
Prelude	Yoann Vandoorselaere and Laurent Oudot
Snort	Martin Roesch

Частина розглянутих систем (Bro, NetSTAT) розроблені в університетах і базуються на дослідженнях в області виявлення атак, проведених в цих університетах.

1.8.1. Результати порівняльного аналізу

В розділі наводяться результати порівняння розглянутих СВА. Системи порівнюються окремо по кожному з вищевказаних критеріїв. Зведена таблиця

порівняння СВА за всіма критеріями дана в кінці розділу. Всі розглянуті системи використовують в якості основного методу виявлення атак сигнатурний метод (порівняння рядків, шаблонів).

Система Bro використовує регулярні вирази над трасами, які формуються мережевими протоколами. Набір регулярних виразів створюється експертами. Крім того, до складу системи входить транслятор сигнатур з формату системи Snort в сценарії Bro (хоча в даний час цей транслятор може не підтримувати деякі конструкції мови Snort).

Система OSSEC є монолітною - в сенсори і аналізатори «зашиті» знання розробників системи виявлення атак про те, які послідовності повідомлень в журналах можуть бути ознаками атаки. Така архітектура системи є важко розширювана з точки зору бази знань про атаки.

Система NetSTAT використовує мову опису сценаріїв атак STATL, особливістю якої є можливість опису сценарію атаки у вигляді послідовності дій над ресурсом що атакується. Таким чином, ця система використовує метод виявлення, близький до методу аналізу переходів станів.

Система Prelude використовує різні аналізуючі компоненти для мережевих даних і журналів реєстрації. Для аналізу мережевих даних можна використовувати систему Snort. Також використовується набір спеціалізованих модулів для виявлення специфічних атак, таких як сканування портів, некоректні ARP-пакети і т.п. Спеціальні модулі роблять дефрагментацію IP, складання TCP-потоку, декодування HTTP-запитів.

Система Snort використовує базу сигнатур відомих атак. У ній також використовується набір спеціалізованих модулів для виявлення специфічних атак, таких як сканування портів або відправка великої кількості фрагментованих пакетів. Спеціальні модулі роблять дефрагментацію IP, декодування HTTP-запитів. Сторонні розробники часто реалізують інші методи виявлення атак у вигляді модулів (препроцесорів) Snort. Але в основну версію системи вони не входять.[14]

Клас атак, що виявляються

Всі досліджені системи можуть виявляти атаки кількох класів. Тому для поліпшення читабельності і скорочення обсягу тексту вводяться поняття об'єднання, перетину і вкладення класів атак. Для позначення об'єднання і перетину класів атак будемо використовувати символи «∪» і «∩» відповідно. Системи розглянуті в даній роботі, призначені для виявлення атак різних класів. Частина систем орієнтована на виявлення вузлових атак, і використовує для аналізу такі джерела як журнали реєстрації додатків, ОС, журнали систем аудиту (OSSEC). Інші системи виявляють тільки зовнішні (мережеві) атаки і використовують для аналізу інформацію, що отримується з каналів передачі даних в мережі (Bro, Snort). Решта системи є гібридними і виявляють як локальні, так і зовнішні атаки (STAT, Prelude).

Система Bro є мережевою системою виявлення атак. Вона являє собою набір модулів декомпозиції даних різних мережевих протоколів (від мережевого до прикладного рівня) і набір сигнатур над подіями відповідних протоколів. Сигнатури Bro фактично представляють собою регулярні вирази в алфавітах протоколів.[15]

Дана система може знаходити атаки наступних класів (L, R, A, D):

1. L = { li ∪ le } (внутрішні і зовнішні атаки);

2. R = { rn } ∩ { ru ∪ rs } (атаки на мережеві призначені для користувача ресурси і системні ресурси);

3. A = { as ∪ au ∪ ar ∪ ad } (збір інформації про систему, спроби отримання прав користувача, спроби отримання прав адміністратора і порушення працездатності ресурсу);

4. D = {dn∪dd} (нерозподілені і розподілені).

Система OSSEC, єдина з розглянутих в даній роботі, є орієнтованою на виявлення атак рівня системи (вузлових). Вона найбільш «молода» з розглянутих систем. Її остання версія призначена, зокрема, для аналізу журналів реєстрації UNIX, типових програм (ftpd, apache, mail, etc), а також журналів міжмережевих

екранів і мережевих СВА. OSSEC включає в себе набір аналізаторів для різних джерел даних, контроль цілісності файлової системи, сигнатури відомих троянських закладок (rootkits) та ін.

Виявляються атаки наступних класів:

1. L = { li } (атакуючі об'єкти знаходяться всередині системи);

2. R = { rl } ∩ { ru ∪ rs } (вузлові призначені для користувача і системні ресурси);

3. A = { au ∪ ar ∪ ai } (спроби отримання прав користувача, спроби отримання прав адміністратора, порушення цілісності ресурсу);

4. D = { dn ∪ dd } (нерозподілені і розподілені).

Система STAT є експериментальною університетської розробкою, і найбільш «старою» з розглянутих систем. Система включає в себе набір компонентів виявлення атак різних рівнів - мережевий (NetSTAT), вузловий (USTAT, WinSTAT), додатків (WebSTAT), тобто є класичною гібридною системою.

СВА виявляє атаки наступних класів (L, R, A, D):

1. L = { li ∪ le } (внутрішні і зовнішні атаки);

2. R = { rl ∪ rn } ∩ { ru ∪ rs } (атаки на вузлові або мережеві призначені для користувача ресурси і системні ресурси);

3. A = { as ∪ au ∪ ar ∪ ad } (збір інформації про систему, спроби отримання прав користувача, спроби отримання прав адміністратора і порушення працездатності ресурсу);

4. D = { dn } (нерозподілені).

Система Prelude, як і NetSTAT, є гібридної, тобто здатна виявити атаки як на рівні системи, так і на рівні мережі. Дана система спочатку розроблялася в якості самостійної СВА, але в даний час є високорівневою надбудовою над відкритими СВА і системами контролю цілісності (AIDE, Osiris і т.п.). Вузлова частина Prelude має досить широкий набір описів атак і, як джерело інформації, використовує різні журнали реєстрації:

- журнали реєстрації брандмауера IPFW;

- журнали реєстрації ОС Linux;
- журнали реєстрації маршрутизаторів Cisco and Zyxel;
- журнали реєстрації GRSecurity;
- журнали реєстрації типових сервісів ОС UNIX та інші.

СВА виявляє атаки наступних класів (L, R, A, D):

1. L = { li ∪ le } (внутрішні і зовнішні атаки);

2. R = { rl ∪ rn } ∩ { ru ∪ rs ∪ rp } (атаки на локальні або мережеві призначені для користувача ресурси, системні ресурси і ресурси захисту);

3. A = { as ∪ au ∪ ar ∪ ad } (збір інформації про систему, спроби отримання прав користувача, спроби отримання прав адміністратора і порушення працездатності ресурсу);

4. D = { dn } (нерозподілені)

Система Snort це найбільш популярна на сьогоднішній день некомерційна СВА. Вона активно і динамічно розвивається, оновлення бази відомих атак відбуваються з частотою, порівнянною з комерційними аналогами (зазвичай поновлення Snort випереджають комерційні). Snort є чисто мережевою СВА і, крім основної бази описів атак, має набір модулів для виявлення специфічних атак або реалізують альтернативні методи виявлення.

Система здатна виявити атаки наступних класів (L, R, A, D):

- L = "внутрішні" ∪ "зовнішні";
- R = {rl ∪ rn} ∩ {ru ∪ rs ∪ rp } (атаки на локальні або мережеві призначені для користувача ресурси, системні ресурси і ресурси захисту);
- A = {as ∪ au ∪ ar ∪ ad } (збір інформації про систему, спроби отримання прав користувача, спроби отримання прав адміністратора і порушення працездатності ресурсу);
- D = {dn ∪ dd } (нерозподілені і розподілені).

Таким чином, жодна з розглянутих систем не покриває всю множину класів атак. Слід також зазначити, що ці системи використовують неадаптивні методи виявлення атак.

Рівень спостереження за системою

Всі розглянуті вище системи працюють з даними додатків і операційної системи на вузловому рівні, а так само з мережевими даними. Тобто інформація, що аналізується, виходить з вторинних джерел, таких як журнали реєстрації додатків, ОС, або з мережевого каналу передачі даних. Система OSSEC працює виключно з журналами реєстрації додатків і операційної системи. Системи Bro, Snort аналізують тільки мережеві дані. Системи NetSTAT і Prelude аналізують як дані з локальних системних джерел, так і мережеві дані.

З розглянутих систем жодна не покриває всі рівні спостереження, і інформація, що аналізується, кожною системою неповна з точки зору можливості виявлення атак всіх класів. Для виявлення атак всіх класів необхідно аналізувати інформацію на всіх трьох рівнях одночасно.

Адаптивність до невідомих атак

На даний момент ця можливість в розглянутих СВА відсутня. Можливе використання експериментального модуля статистичного аналізу системи Snort, але його ефективність не вивчена. За рахунок контролю цілісності ресурсів вузла в системі OSSEC присутня умовна адаптивність. Проте, слід визнати, що контроль цілісності вирішує не завдання виявлення атак, а лише завдання виявлення їх наслідків. Таким чином, адаптивність до невідомих атак в розглянутих СВА, в цілому, відсутня.

Масштабованість

Система Bro є нерозподіленою і управляється централізовано на тому вузлі, де вона встановлена, за допомогою файлів конфігурації. При збільшенні числа вузлів що захищаються і каналів зв'язку необхідно встановлювати додаткові незалежні екземпляри системи Bro, що означає фактичну не масштабованість.

Система OSSEC є розподіленою і управляється розподіллено на вузлах, де встановлені агенти (за допомогою файлів конфігурації), або централізовано за

допомогою спеціалізованої утиліти адміністрування (manage_agents) з центрального сервера OSSEC. Система є добре масштабована.

Система NetSTAT також є розподіленої і управляється розподіллено через файли конфігурації на всіх вузлах, де розташовані компоненти системи. Індивідуальне управління компонентами системи робить процес управління і настройки складним і тривалим, причому з ростом числа компонентів складність настройки і внесення змін в конфігурацію ускладнюється.

Система Prelude є розподіленою і управляється централізовано за допомогою керуючої консолі. Компоненти системи самі надають керуючій консолі ті параметри їх функціонування, які можуть змінюватися. Управління проводиться по захищеному каналу (SSL). Також управління може здійснюватися через локальні конфігураційні файли на тих вузлах, де встановлені компоненти СВА. Дана система є добре масштабована.

Система Snort управляється централізовано через файли конфігурації, консольні команди і команди UNIX. Сама по собі система не є масштабованою, але в разі використання Snort як сенсор системи Prelude цей недолік зникає.[16]

Відкритість

Три з розглянутих систем, за винятком Bro і OSSEC, мають відкритий інтерфейс для додавання нових аналізуючих модулів, а також використовують стандартний для систем виявлення атак формат обміну повідомленнями (IDMEF).

Система Bro дозволяє користувачеві і стороннім розробникам розширювати набір сигнатур.

NetSTAT має відкритий інтерфейс для додавання нових агентів і фільтрів.

Prelude має відкритий інтерфейс для додавання нових модулів аналізу і реагування, а так само ведення журналів реєстрації. Обмін повідомленнями між компонентами системи відбувається за стандартом IDMEF (Intrusion Detection Message Exchange Format), оптимізованому для високошвидкісної обробки.

Snort має відкритий інтерфейс для додавання нових модулів аналізу; є модуль, який реалізує протокол SNMPv2. За сукупністю використовуваних

стандартних інтерфейсів, системи Prelude і Snort краще за інших дозволяють нарощувати функціональність по виявленню атак.

Формування відповідної реакції на атаку

Вбудовану можливість реагування на атаку мають всі розглянуті системи.

В системі NetSTAT це реалізовано лише в тестовому варіанті.

Система Prelude має набір відповідних реакцій, які можуть блокувати атакуючого за допомогою брандмауера. Ведуться роботи по агентам-модулям, здатним або повністю ізолювати атакуючого, або зменшити пропускну здатність його каналу.

Система Snort має вбудовану обмежену можливість реагування на атаку шляхом відправки TCP-пакетів, що розривають з'єднання (з встановленим флагом RST), а також ICMP-пакетів, які повідомляють атакуючому вузлу про недоступність вузла, мережі або сервісу.

Аналогічна функціональність з реагування доступна в системі Bro.

Система OSSEC дозволяє використовувати довільні команди для реагування - для цього необхідно статично задати відповідність між подією, командою і параметрами її виклику.

Захищеність

Всі системи, які пересилають будь-які дані, використовують для цього захищені канали.

STAT і Prelude використовують бібліотеку OpenSSL для шифрування каналу між компонентами. Snort реалізує протокол SNMPv2, в якому присутні функції шифрування паролів при передачі даних. СВА Prelude має додаткові механізми, що забезпечують безпеку її компонентів. В системі використовується спеціалізована бібліотека, яка робить безпечними такі бібліотечні функції алгоритмічної мови С як printf, strcpy, які не перевіряють розмір переданих їм даних. Бібліотека запобігає класичні помилки виходу за межі масивів і переповнення буферів. Додаткові модулі аналізу мережевих даних роблять

систему стійкою до некоректних мережевих пакетів на різних рівнях стека і виходу її компонентів з ладу. Такі атаки, як відправка пакетів з неправильними контрольними сумами, обнуленими прапорами TCP, розсинхронізація сесій, помилкова відправка і «обрізання» сегментів системою ігноруються.

З розглянутих систем питання безпеки найбільш опрацьоване в системі Prelude. В табл. 1.3 наведено зведені результати порівняння розглянутих систем за обраними критеріями.

Таблиця 1.3 – Порівняльна характеристика систем виявлення атак

	Bro	OSSEC	STAT	Prelude	Snort
Класи атак	({li ∪ le}, {rl} ∩ {ru ∪rs ∪ rc}, {as ∪ au ∪ ar ∪ ad}, {dn})	({li }, {rl} ∩ {ru ∪ rs}, {au ∪ ar ∪ ai}, {dn})	({li ∪ le}, {rl ∪ rn) ∩ (ru ∪ rs}, {as ∪ au ∪ ar ∪ ad}, {dn})	({li ∪ le}, {rl ∪ rn) ∩ {ru ∪ rs ∪rp}, {as ∪ au ∪ ar ∪ ad}, {dn∪ dd})	({li ∪ le}, {rl ∪ rn) ∩ {ru ∪rs∪rp}, {as ∪ au ∪ ar ∪ ad}, {dn})
Рівень спостережності за системою	Системний	Системний	Системний, мережевий	Системний, мережевий	Мережевий
Метод виявлення	Сигнатурний	Сигнатурний	Сигнатурний, аналіз переходів стану	Сигнатурний	Сигнатурний
Адаптивність	-	+/-	-	-	-
Масштабованість	-	+	+	+	-
Відкритість	Відкрите API	Відкрите API	Відкрите API	Відкрите API, IDMEF	Відкрите API, SNMPv2
Реакція	-	-	+	+	+

1.9.Висновки

Проведений огляд і порівняльний аналіз методів і систем виявлення атак дозволяють зробити висновок про те, що в даний момент не існує відкритої загальнодоступної системи виявлення, яка мала б адаптивністю до невідомих атак. Незважаючи на наявність великої кількості методів виявлення аномалій, значна кількість помилкових спрацьовувань, слабка стійкість і неверіфікованість не дозволяє їх використовувати в системах загального призначення.

Крім того, до сих пір використання методів виявлення аномалій обмежена дослідними і вузькоспеціалізованими системами. Наприклад, робляться спроби використовувати методи класифікації даних, такі як кластерний аналіз і нейронні/імунні мережі. Головний недолік цих методів, в порівнянні з експертними методами і аналізом систем переходів, в принциповій неможливості верифікації результату (виходу) - класи поведінки будуються на основі навчання і не представляється можливим аналітично розрахувати рівень помилок I і II роду для таких методів, а також проаналізувати коректність прийнятих рішень.

Для вирішення даної проблеми необхідно розробити адаптивний метод виявлення атак, який при низькій (лінійної) обчислювальної складності, стійкості і верифікованості матиме низький рівень помилкових спрацьовувань.

Також можна зробити висновок про те, що в області захисту від атак спостерігається перехід від виявлення атак до запобігання атак: всі розглянуті відкриті СВА вже включають в себе засоби реагування на атаки, які в обов'язковому порядку включають в себе розрив з'єднання і взаємодія з зовнішніми засобами захисту - міжмережевими екранами і т.д.

Дана тенденція, в поєднанні з постійним підвищенням пропускної спроможності каналів передачі даних, пред'являє підвищені вимоги до обчислювальної складності алгоритмів виявлення атак. При цьому більшість методів виявлення аномалій має високу обчислювальну складність у порівнянні з найбільш поширеним сигнатурним методом.

Зважаючи на вищенаведене представляється перспективним розробити гібридний метод виявлення атак, який об'єднає сигнатурний метод і метод аналізу систем переходів для виявлення відхилень від нормальної поведінки. Об'єднання

цих методів допоможе зберегти верифікованість, стійкість і низьку обчислювальну складність при виявленні зловживань і доповнити їх властивістю адаптивності до невідомим атакам.

РОЗДІЛ 2. МОДЕЛЬ ВИЯВЛЕННЯ АТАК

2.1. Модель функціонування РІС

У роботі терміни «несанкціонований доступ» і «несанкціонований вплив» ми будемо розуміти:

Захист інформації від несанкціонованого впливу - діяльність по запобігання впливу на інформацію з порушенням встановлених прав і/або правил на зміну інформації, що призводить до спотворення, знищення, копіювання, блокування доступу до інформації, а також до втрати, знищення або збою функціонування носія інформації.

Захист інформації від несанкціонованого доступу - діяльність по запобіганню отримання інформації, що захищається зацікавленим суб'єктом з порушенням встановлених правовими документами чи власником інформації прав або правил доступу до інформації, що захищається.

Взаємодія об'єктів - виконання операцій доступу одного об'єкта до іншого. Наприклад, читання, запис, запуск на виконання і т.д.

2.1.1. Основні поняття і визначення

Уявімо розподілену інформаційну систему (далі РІС) як безліч взаємодіючих типів об'єктів (далі просто об'єктів). Для кожного типу об'єкта визначені операції доступу (далі просто операції), наприклад, по відкриттю, читанню, запису, запуску на виконання і т.д. Доступ до об'єкту може бути як безпосереднім, так і непрямим - через інші об'єкти.

Під доступом за деякою операцією до об'єкту мається на увазі виконання послідовності елементарних операцій – відкрити доступ, виконати операцію, закрити доступ.

Припускаємо, що в ході реалізації доступу елементарні операції виконуються миттєво, але можуть відставати одна від одної в часі.

Для кожного об'єкта також визначено кінцевий набір прав доступу - правила визначають, до яких типів об'єктів він може мати доступ. Будь який доступ до об'єкту, який не відповідає цим правилам, будемо називати несанкціонованим.

Завдання систем захисту РІС - виявлення несанкціонованого доступу до об'єктів РІС і протидія будь яким спробам несанкціонованої зміни прав доступу.

Будемо називати такі дії зловмисними, а об'єкт, який ініціює такі дії - зловмисником або порушником.

Кожен об'єкт в системі характеризується станом. Стан об'єкта – це безліч об'єктів РІС, що мають доступ до нього, а також характеристика його поточної завантаженості.

Далі ми строго визначимо поняття стану об'єкта. Об'єкти можна розділити на дві підмножини: підмножина активних і підмножина пасивних об'єктів.

Пасивний об'єкт не може здійснювати доступ до інших об'єктів, тоді як активний може.

2.1.2. Модель поведінки об'єкта і модель атаки

Тепер від визначення відносин між екземплярами об'єктів РІС перейдемо до опису функціонування РІС. Назвемо траєкторією екземпляра об'єкта деяку непорожню кінцеву послідовність станів об'єкта.

Траєкторію екземпляра активного об'єкта можна уявити як послідовність відрізків траєкторій взаємодіючих з ним екземплярів об'єктів і власних дій екземпляра над екземплярами інших об'єктів РІС, так як будь яка операція відкриття або закриття доступу викликає зміну стану взаємодіючих об'єктів.

Траєкторію екземпляра пасивного об'єкта можна уявити як послідовність відрізків траєкторій взаємодіючих з ним екземплярів активних об'єктів.

Визначимо траєкторію РІС як непорожню кінцеву послідовність станів РІС, в якій будь-які два сусідніх стану РІС відмінні один від одного.

Функціонування РІС представимо як деяку послідовність станів РІС, таку, що кожний наступний стан відрізняється від попереднього стану хоча б одного з об'єктів або об'єктів РІС. Для цього введемо поняття повної траєкторії РІС.

Тобто для будь-якого екземпляра об'єкта РІС, такого що його стан і його сусідні стани різні, і який взаємодіє з екземпляром об'єкта РІС в більш ранньому стані, яка викликала зміну станів. Ці умови означають, що якщо деяка траєкторія РІС повна, то вона містить всі траєкторії екземплярів об'єктів, що входять в неї.

Тепер визначимо поняття небезпечного і безпечного стану екземпляра об'єкта РІС. РІС знаходиться в інформаційно безпечному стані, якщо всі екземпляри об'єктів РІС знаходяться в інформаційно безпечному стані. Інформаційно безпечний стан РІС - це множина станів РІС, в яких відсутні такі порушення: порушення конфіденційності екземплярів об'єктів РІС, порушення цілісності екземплярів об'єктів РІС, порушення доступності екземплярів об'єктів РІС. У термінах даної моделі інформаційно безпечний стан РІС - це множина станів РІС, в яких всі екземпляри об'єктів, що мають доступ до інших екземплярів об'єктів РІС, мають права доступу, і завантаження кожного об'єкта менше його ємності.

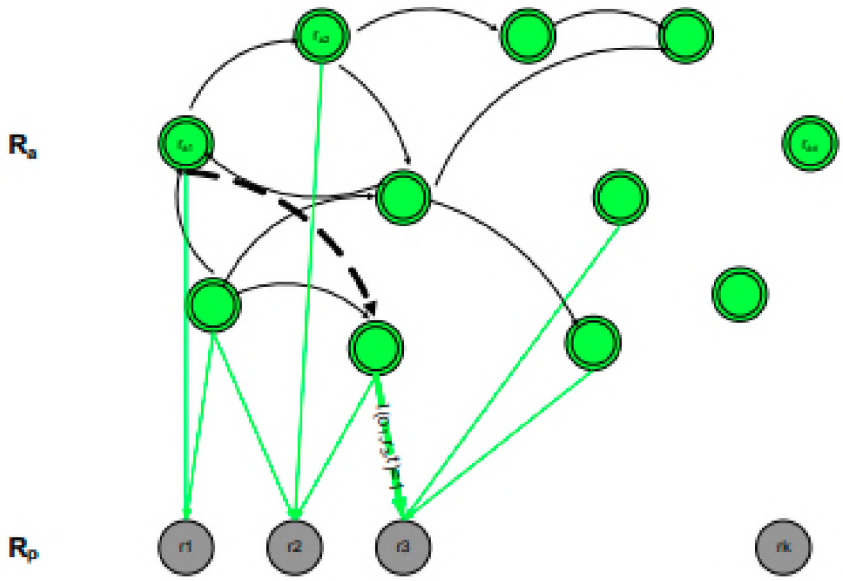
Приклад небезпечного стану: при вичерпанні максимального числа відкритих TCP-з'єднань можна говорити про наявність атаки, що порушує доступність об'єкта (атаки типу DoS). Визначимо порушення інформаційної безпеки РІС як перехід РІС з деякого безпечного стану в будь-який небезпечний.[17]

Множина небезпечних станів РІС - це множина станів, в яких хоча б один екземпляр об'єкта знаходиться в небезпечному стані. Перехід будь-якого екземпляра об'єкта в небезпечний стан за визначенням означає перехід всієї РІС в небезпечний стан.

Таке визначення небезпечного стану РІС означає, що для його виявлення немає необхідності збирати інформацію про стан кожного об'єкта РІС в кожен момент часу, а досить спостерігати тільки за змінами станів критичних, з точки

зору захисту інформації, об'єктів. Слід зазначити, що для будь-якого стану РІС можна зіставити граф доступу.

На рис. 2.1. показаний граф доступу на безлічі екземплярів об'єктів РІС в деякому стані РІС.



«Рисунок 2.1 – Граф доступа в РІС»

Атакою на РІС назвемо таку послідовність дій, зроблених деяким екземпляром об'єкта, або набір послідовностей дій групи екземплярів об'єктів. Таким чином, атака - це траєкторія деякого екземпляра об'єкту або набір ділянок траєкторій деякої групи екземплярів об'єктів, що виводить РІС з інформаційно безпечного стану.

Атака переводить систему з безпечного стану в небезпечний, або залишає систему в раніше досягнутому небезпечному стані.

Нормальну поведінку об'єкта визначимо як множину траєкторій всіх екземплярів об'єктів даного типу, які не виводять РІС зі стану інформаційної безпеки.

Будемо вважати, що жодна траєкторія, що входить в нормальну поведінку об'єкта, не може містити небезпечних станів, тобто станів, в яких ресурс

перевантажений або існує ланцюжок транзитивного доступу, який порушує відносини прав доступу.

Запропонований метод виявлення атак використовує множину траєкторій атак і описів нормальної поведінки для виявлення зловживань і аномалій поведінки об'єктів на основі зіставлення спостережуваної поведінки екземплярів об'єктів із заданими прикладами.

Задачу виявлення атаки можна розділити на дві підзадачі: виявлення зловживань і виявлення аномалій як відхилення від нормальної поведінки.

2.2. Розпізнавання нормальних і аномальних траєкторій

Назвемо класом атак множину атак, результатом яких є один і той же небезпечний стан будь-якого екземпляра об'єкта певного типу. Окрему атаку класу будемо називати екземпляром класу. Екземпляр атаки відповідає атаці на екземпляр об'єкта відповідного типу. Таким чином, для кожного типу об'єктів PIC визначено деяку множину класів атак.

Будемо розглядати клас атак як мову, в якій траєкторії є словами, а дії об'єктів – алфавітом.

Прирівняємо кожному класу атак кінцевий автомат першого роду таким чином, щоб він приймав будь-яку послідовність станів екземплярів об'єктів.

Після побудови автомата проводиться процедура його мінімізації.

Для вирішення задачі розпізнавання атак необхідно побудувати новий автомат, який повинен розпізнавати ті ж послідовності, але без останнього символу.

Для цього видаляємо термінальний стан автомата і переходи, що ведуть в нього, а попередні стани робимо термінальними.

Формально автомат для кожного *i*-го класу атак є п'ятірку наступного виду:

$$K\ S\ T\ s0\ Q\ i\ r\ \Sigma, \qquad (2.1)$$

де S - безліч станів;

Σ - вхідний алфавіт (безліч дій);

T - функція переходів;

S0- початковий стан;

Q - безліч заключних станів.

Конструктивно автомат першого роду будемо будувати за таким алгоритмом:

1. Визначити множину реалізацій атак одного класу (задана множина прикладів атак);

2. Виділити послідовність станів атакуючого об'єкта і об'єкта що атакується, які входять в будь-яку траєкторію атаки із заданої множини реалізацій;

3. Кожному стану отриманої послідовності зіставити стан автомата першого роду. Безлічі дій, які ведуть в якийсь стан, зіставити переходи в цей стан;

4. Замінити кінцевий (небезпечний) стан на термінальний стан, залишивши всі переходи в ньому.

Відзначимо, що автомати першого роду дозволяють розпізнавати приналежність траєкторії до деякого класу атак. У той же час, можна використовувати аналогічний механізм для розпізнавання приналежності траєкторії до класу нормальної поведінки деякого об'єкту. Відповідно, в силу безлічі можливих дій в кожному стані об'єкта, автомат розпізнавання нормальної поведінки можна доповнити безліччю кінцевих станів таким чином, щоб він приймав всі можливі відхилення від нормальної траєкторії об'єкта.

Прирівняємо кожному об'єкту PIC кінцевий автомат другого роду таким чином, щоб він приймав будь-яку послідовність станів об'єкта відповідного типу, яка закінчується небезпечним станом і містить всі стани об'єкта, які входять в будь-яку нормальну траєкторію для об'єктів даного типу.

Автомат другого роду будується на основі опису нормальної поведінки об'єктів. При цьому опис нормальної поведінки об'єкта має бути визначено

заздалегідь у вигляді специфікації в текстовому вигляді і реалізації у вигляді тексту на мові програмування. Для побудови автомата другого роду необхідно доповнити автомат нормальної поведінки заданого типу об'єктів додатковими станами і для кожного нормального стану визначити безліч аномальних переходів, що ведуть в небезпечний стан.[18]

Дана операція завжди можлива в силу того, що безліч дій над даним об'єктом завжди звичайна і повністю визначається безліччю операцій.

Для кожного спостережуваного екземпляра об'єктів заданого типу необхідно створювати окрему копію автомата другого роду, де буде відбуватися аналіз відповідності спостережуваної поведінки екземпляру нормальної поведінки об'єктів цього типу.

Загальна кількість кінцевих автоматів для розпізнавання N класів атак на M екземплярів об'єктів N * M.

Обчислювальна складність розпізнавання атак кінцевим автоматом не залежить від числа станів і визначається лише від довжиною входу. Максимальну довжину входу можна оцінити як максимальне число одночасно спостережуваних траєкторій (M + K) * *l*, де K - число типів об'єктів, тобто максимальна складність розпізнавання буде дорівнює N * M * (M + K) * *l* . отже, складність виявлення N класів атак на M об'єктів для K об'єктів за допомогою кінцевих автоматів дорівнює O (N * M 2 * K * l), де *l* - максимальна довжина спостерігається траєкторії.

Таким чином, складність розпізнавання атак кінцевими автоматами першого і другого роду має лінійну залежність від довжини траєкторії, тобто від потоку аналізованих подій.

2.3. Мова опису автоматів першого і другого роду

Важливою особливістю поведінки мережевих об'єктів PIC є спосіб обслуговування запитів до сервісів за моделлю взаємодії клієнт-сервер. Для цього при появі нового «клієнтського» об'єкта породжується новий логічний екземпляр

«серверного» об'єкта. В одних РІС це реалізується породженням нового екземпляра об'єкта на кожен об'єкт, в інших - створенням і підтриманням незалежних контекстів обслуговування декількох об'єктів в одному об'єкті.[19]

Ця особливість означає, що траєкторія поведінки РІС може мати складну структуру: деякі дії об'єктів РІС можуть створювати екземпляри об'єктів так, що всі наступні дії об'єктів будуть породжувати зміни станів тільки в цих породжених екземплярах об'єктів. Відповідно, існує безліч дій, які ведуть до знищення екземплярів об'єктів. Як правило, до породжувальних дій відносяться запити на обслуговування до об'єкта, а до знищувальних дій - будь-які дії, що викликають припинення обслуговування об'єкта. Таким чином, один «серверний» об'єкт може мати одночасно кілька траєкторій, а «клієнтський» об'єкт завжди має лише одну спостережувану траєкторію.

Для кожної РІС можна описати два типи автоматів:

- автомати першого роду описують нормальну поведінку об'єкта
- автомати другого роду описують атаки

Для параметризації автомата використовуються змінні конфігурації, які можуть бути використані в предиката станів і переходів. Безліч таких зовнішніх змінних є для автомата глобальним оточенням, тоді як набір внутрішніх змінних і їх значень є локальним оточенням.

Формально автомат для кожного класу атак A і являє собою структуру наступного виду:

$$K_R^i: (S, P_S, T, P_T, S_0, I, g, q), \qquad (2.2)$$

де S- множина станів;

P_S- множина предикатів станів;

T- множина переходів;

P_T- множина предикатів переходів;

S_0- початковий стан;

I- множина екземплярів автомата;

g- глобальне оточення;

q- глобальна чергу таймера.

Перехід виконується в тому випадку, якщо предикати переходу і цільового стану приймають значення «true» одночасно

Приклад опису автомата першого класу наведено у додатку Д. Даний приклад описує автомат, який виявляє мережеве з'єднання на стандартний порт FTP. У додатку Е наведено приклад автомата другого роду, який описує нормальну поведінку сервера FTP .

Автомат побудований за описом протоколу FTP в стандарті RFC 959 і його реалізації в сервері . Для кожного стану сервера FTP виділено окремий стан автомата другого роду. Переходи між станами відповідають переходам сервера FTP відповідно до специфікації протоколу. При цьому в кожному стані виявляються переходи, які викликають породження нового процесу і виконання в ньому довільного коду (fork () і exec ()).

Це дозволяє виявляти будь-які атаки класу «remote root» для сервера , включаючи раніше невідомі.[20]

2.4. Алгоритми виявлення атак

Алгоритм виявлення атак зводиться до формування послідовності спостережуваних станів PIC і пошуку в цій послідовності траєкторій атак і відхилень від нормальної поведінки:

1. Для кожного об'єкта, що захищається формуємо множину автоматів першого роду(кожен автомат в цій множині моделює атаку деякого класу) і автомат другого роду.

2. Нехай від спостерігача надійшов деякий символ:

a. Визначаємо, до якої множини дій належить даний символ.

b. Визначаємо підмножину автоматів, які можуть прийняти даний символ.

c. Подаємо символ на вхід кожного автомату з обраної підмножини.

3. Для кожного автомата з обраної підмножини обчислюємо предикати переходу з поточного стану по даному символу і предикат підсумкового стану. Якщо предикат підсумкового стану «істина» і предикат переходу «істина».

4. Якщо підсумковий стан автомата є кінцевим, виконуємо дії, визначені для кінцевого стану (формуємо повідомлення про атаку), і знищуємо цей екземпляр автомата, якщо він не єдиний, інакше повертаємо його в початковий стан.

Атака на об'єкт, що захищається буде виявлена при одночасному виконанні наступних умов:

Траєкторія атаки належить одному з класів, які розпізнаються безліччю автоматів першого роду або не належить нормальній поведінці об'єктів даного типу - це є вимогою коректності побудови автомата другого роду.

Всі дії з траєкторії атаки були побачені системою розпізнавальних автоматів, при цьому дії спостерігалися строго в порядку виконання.

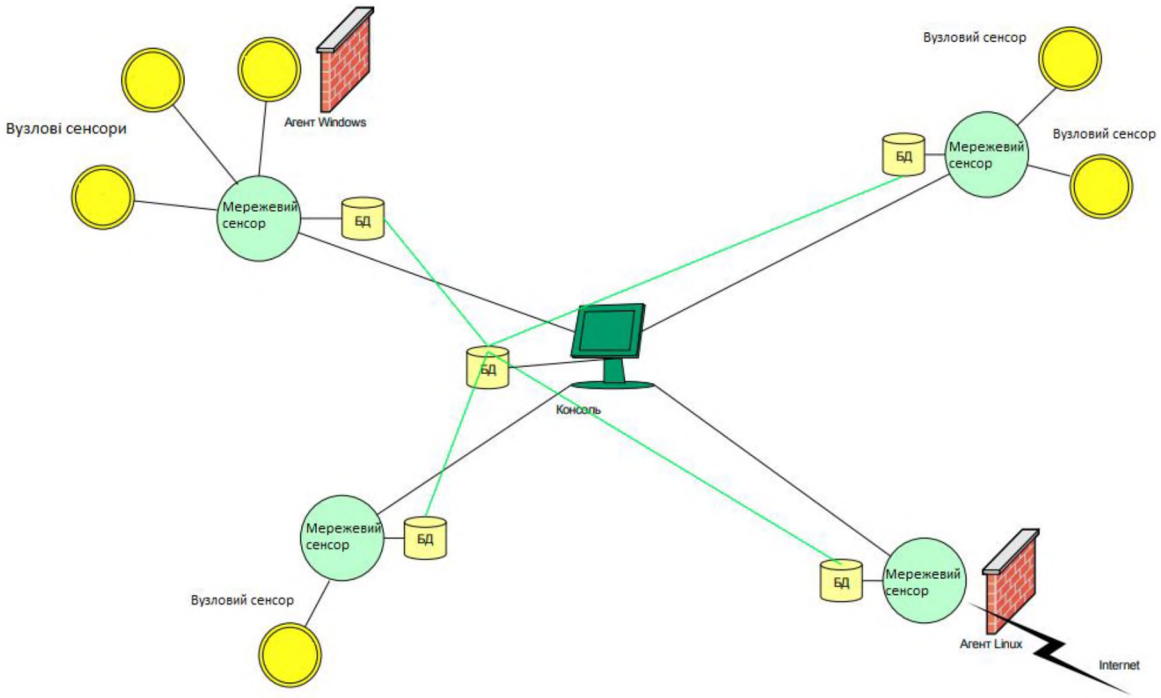
Дані умови є умовами, в яких запропонований алгоритм виявлення атак коректний, тобто для кожної атаки на вході алгоритму результатом виконання алгоритму є хоча б одне повідомлення про атаку.

Нагадаємо вимоги до «ідеальної» СВА:

- покриває всі класи атак (система повна);
- дозволяє аналізувати поведінку РІС на всіх рівнях: мережевому, вузловому і рівні окремих додатків;
- адаптивна до невідомих атак (використовує адаптивний метод виявлення атак);
- масштабується для РІС різних класів: від невеликих локальних мереж класу «домашній офіс» до великих багатосегментних і комутованих корпоративних мереж, забезпечуючи можливість централізованого управління всіма компонентами СВА;
- є відкритою;
- має вбудовані механізми реагування на атаки;
- недостатньо захищена від атак на компоненти СВА, в тому числі перехоплення

2.5 Архітектура і алгоритм методу виявлення атак для роботи системи виявлення атак

На рис. 2.2 представлена схема системи виявлення атак з керуючими зв'язками і зв'язками з передачі повідомлень.



«Рисунок 2.2 – Схема експериментальної СВА»

До складу СВА входять наступні модулі:

- мережевий сенсор;
- вузловий сенсор;
- консоль управління;
- база даних;
- агенти реагування.

Мережевий сенсор

Мережевий сенсор призначений для аналізу поведінки мережевих об'єктів на основі даних мережевого трафіку і повідомлень від вузлових сенсорів. Мережевий сенсор складається з спостерігача, який формує трасу подій на основі декомпозиції мережевого трафіку, розбору інформації різних протоколів, ядра

аналізу, яке отримує потік подій від спостерігача, підсистеми реагування та службової підсистеми, яка відповідає за управління та взаємодія між компонентами СВА.

Ядро аналізу являє собою систему виконання програм (СВП) і набір автоматів першого і другого роду. СВП виконує функції монітора аналізує автомати, формує чергу виконання тіла переходів і станів автоматів, планує порядок виконання автоматів, породжує і знищує екземпляри автоматів. Автомати, по досягненні кінцевого стану, породжують атомарні повідомлення про атаки в форматі IDMEF.

Підсистема реагування також складається з спостерігача, який отримує повідомлення про атаки і формує трасу подій, ядра аналізу (СВП + автомати) і службової підсистеми. Автомати реагування формують політику реагування: в кінцевому стані кожного автомата виконується задана процедура реагування. Це може бути розрив з'єднання, настройка брандмауера, кореляція повідомлень про атаки і формування більш високорівневих повідомлень.

Службова підсистема мережевого сенсора відповідає за організацію шифрованого каналу між компонентами СВА, збереження повідомлень про атаки і службові повідомлення в базі даних, реалізацію функцій віддаленого управління і настройки мережевого сенсора з консолі управління.[21]

Вузловий сенсор

Вузловий сенсор призначений для аналізу поведінки мережевих об'єктів PIC на основі даних системних журналів і подій ОС: траси дій додатків, користувачів, використання файлової системи контрольованої робочої станції або сервера.

Вузловий сенсор складається з спостерігача, який формує трасу подій, ядра аналізу, яке отримує потік подій від спостерігача, і службової підсистеми, яка відповідає за управління та взаємодію між компонентами СВА.

На вихід вузловий сенсор видає повідомлення про виявлені аномалії або зловживаннях в форматі IDMEF. Повідомлення пересилаються мережевому сенсору, який має з'єднання з вузловим сенсором.

База даних

База даних (БД) СВА є розподілене сховище описів нормальної і аномальної поведінки об'єктів РІС, повідомлень про атаки і журналу компонентів СВА. Дане сховище використовується мережевими і вузловими сенсорами для централізованого завантаження автоматів в СВП і зберігання повідомлень про атаки.

База даних побудована на основі відкритої СУБД PostgreSQL.

Агенти реагування

Агенти реагування СВА встановлюються на вузли РІС, де встановлені засоби реагування (міжмережеві екрани), або на контрольовані вузли РІС, і виконують команди від підсистеми реагування мережевого сенсора.

В рамках експериментальної СВА реалізовані наступні агенти:

- агент IPTables для ОС Linux з вбудованими можливостями пакетного фільтра і блокування процесів на вузлі;
- агент для ОС Windows з вбудованими можливостями пакетного фільтра і блокування процесів на вузлі.

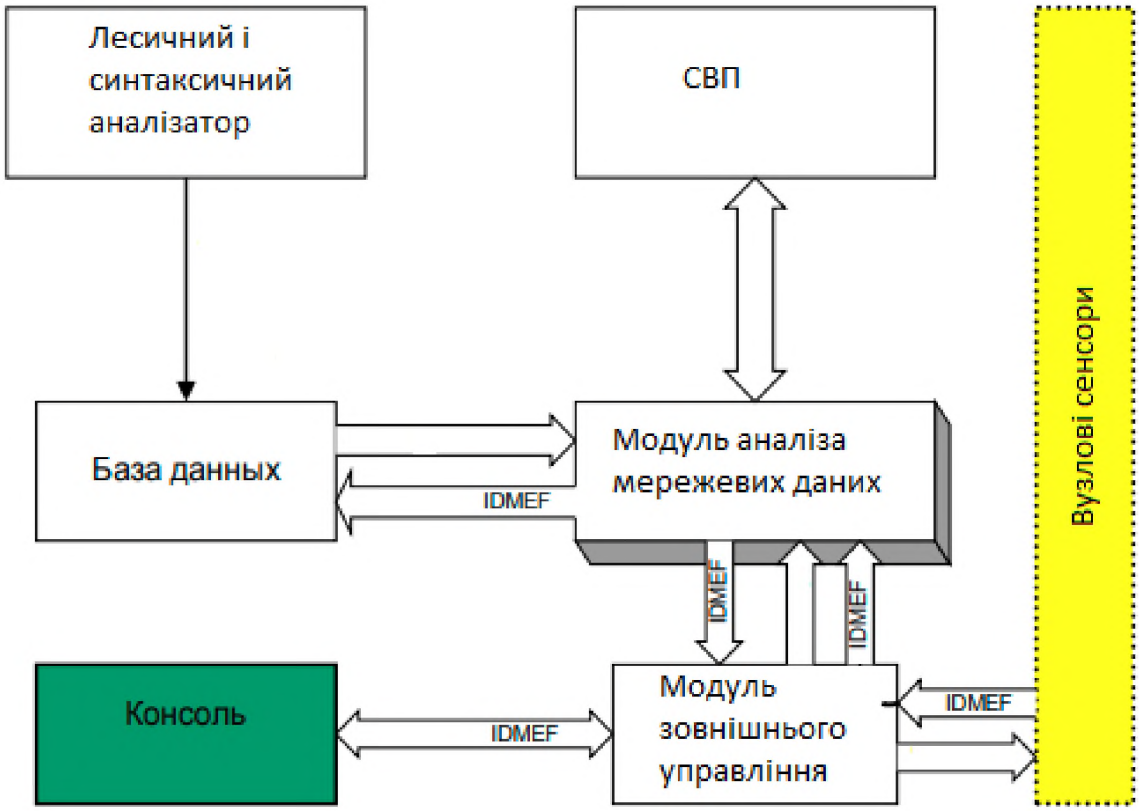
2.5.1. Структура і алгоритми роботи мережевого сенсора

До складу мережевого сенсора СВА входять наступні програмні модулі:

- модуль збору мережевих даних;
- модуль аналізу мережевих даних;
- лексичний і синтаксичний аналізатор мови опису сценаріїв;
- система підтримки виконання програм (СПВП);
- набір сценаріїв (автомати першого і другого роду);
- модуль зовнішнього управління.

На рис. 2.3 представлена програмна архітектурна структура мережевого сенсора і зв'язку між модулями (аналізу мережевих даних, системи виконання програм,

лексичний і синтаксичний аналізатор, тощо), включаючи зовнішні (вузлові сенсори та консоль).



«Рисунок 2.3 - Структура мережевого сенсора і його зв'язок з іншими компонентами»

Модуль збору мережевих даних

Модуль збору мережевої інформації здійснює захоплення всіх вхідних і вихідних пакетів з мережевого інтерфейсу на рівні IP стека TCP / IP. Реалізація модуля під ОС Linux виконує функції захоплення через механізм сокетів типу AF_PACKET.

Перед початком роботи інтерфейс (мережевий адаптер Ethernet) переводиться в режим "promiscuous mode", що дозволяє переглядати всі пакети, видимі в даному сегменті мережі, незалежно від їх отримувача та відправника.

Код ініціалізації мережевого інтерфейсу наведено у додатку Є:

З метою підвищення надійності системи збору мережевої інформації, тобто для уникнення можливої втрати пакетів при інтенсивному завантаженні процесора, застосовується буферизація пакетів.

Після отримання пакет або передається модулю аналізу мережевої інформації, або, якщо модуль аналізу не готовий прийняти пакет, буферизується для подальшого аналізу.

При передачі отриманого пакета модулю аналізу пакет аналізується на предмет відповідності його структури пакетам відомих протоколів стека TCP / IP;

структура зі списком полів передається в модуль аналізу мережевих даних.

Модуль аналізу мережевих даних

Модуль отримує пакети від модуля збору мережевих даних і аналізує їх за допомогою сценаріїв, написаних на мові опису сценаріїв. Кожен мережевий пакет перетвориться в структуру типу NetEvent і передається відповідному сценарію, або безлічі сценаріїв.

Для виконання сценаріїв використовується система виконання програм.

При виявлені атаки викликається функцію Alert(), яка реалізована в бібліотеці мережевого сенсора.[22]

Після отримання пакета від модуля збору мережевої інформації проводиться первинна (оптимізаційна) обробка даних з використанням хеш-таблиць адреси, портів і ідентифікаторів протоколу для визначення попереднього списку сценаріїв, які підходять для обробки даного пакета. Для оптимізації перегляду сценаріїв модуль аналізу мережевої інформації переглядає синтаксичне дерево функцій переходу для кожного із сценаріїв з метою отримання евристичної інформації про предикате переходу. Зокрема, якщо предикат є кон'юнкція декількох логічних виразів і деякі з кон'юнктів мають один з видів:

- IPMatch (event.src_ip, CONST1, CONST2)
- event.src_ip == CONST
- event.src_port == CONST
- IPMatch (event.dst_ip, CONST1, CONST2)

- event.dst_ip == CONST
- event.dst_port == CONST

то відбувається попереднє відсікання сценаріїв, свідомо не діючих для даної події. Відсікання відбувається за допомогою двох хеш-таблиць для пари (Src_ip, src_port) і пари (dst_ip, dst_port).

Повідомлення, що прийшли від вузлових сенсорів, перетворюються в структуру NodeEvent і обробляються аналогічним чином.

Сценарії

Кожен сценарій з набору сценаріїв являє собою реалізацію автомата першого або другого роду у відповідності з моделлю. Наведено короткий опис сценарію і переходів для розуміння алгоритмів роботи сенсорів СВА.

Визначення сценарію:

'Scenario' <ім'я-сценарію> '(' <список-аргументів> ')' '{'
[<Декларація-змінної> | <декларація-стану> | <декларація-переходу>]'};'

Список аргументів в неявному вигляді задає клас подій, оброблюваних сценарієм. Типи структур, що передаються в сценарій, сигналізують про події. Всі аргументи повинні бути структурами, успадкованими від структури Event.

У тілі сценарію можуть бути визначені (як і в структурі) змінні і методи.

Кожному сценарієм в процесі роботи зіставляється один або більше контекстів виконання. Контекст виконання - це значення змінних, визначених всередині сценарію плюс поточний стан. Кожен контекст виконання відповідає окремому сценарію.

Декларація стану:

['Initial'] 'state' <ім'я-стану> '{'<Логічний вираз><Оператор>'}'

Ім'я стану є ідентифікатором, унікальним в межах стану. Логічний вираз визначає умову переходу в стан. Оператор виконується при переході в стан. Ключове слово initial говорить про те, що стан є початковим для сценарію. У разі

якщо ні для одного стани не застосовано ключове слово initial, початковим станом є перший стан сценарію. Сценарій повинен містити хоча б один стан.

Декларація переходу:

'Transition' [<ім'я-переходу>] '(' <ім'я-стану> '->' <ім'я-стану> ')'['Consuming' | 'nonconsuming' | 'unwinding'] '{'<Предикат><Оператор>'}'

Якщо предикат має значення «не істина» або невизначений, подальших перевірок не відбувається. Здійснюється перевірка предиката стану.

Якщо він «істина», здійснюється перехід і виконується оператор, визначений в тілі стану.[23]

2.5.2 Структура і алгоритми роботи вузлового сенсора

Структура і алгоритми роботи вузлового сенсора аналогічні структурі та алгоритмам роботи мережевого сенсора з поправкою на драйвери збору подій та іншої спосіб формування траси подій для аналізу.

До складу вузлового сенсора СВА входять наступні програмні модулі:

- провайдери вузлових подій;
- модуль аналізу подій;
- система підтримки виконання програм (СПВП);
- набір сценаріїв (автомати першого і другого роду);
- модуль зовнішнього управління.

Провайдери вузлових подій

До складу вузлового сенсора експериментальної СВА входять наступні провайдери подій:

- підсистема збору мережевих даних;
- драйвер вузлових подій;
- драйвер подій апаратури.

Підсистема збору мережевих даних організована аналогічно підсистемі мережевого сенсора.

Драйвер вузлових подій дозволяє відстежувати такі класи подій:

- файлова система (створення, видалення,відкриття, закриття, зміна файлів і т.д.);
- процеси (запуск і завершення процесів);
- реєстр (тільки для операційної системи Windows: створення, видалення і зміна ключів);
- сокети (створення сокетів, прослуховування, читання і відправка даних);
- стандартні журнали (тільки для операційної системи Windows: події системи і додатків).

Модуль аналізу подій і інші модулі вузлового сенсора реалізовані аналогічно мережному сенсора.

2.5.3. Структура і алгоритми роботи підсистеми реагування

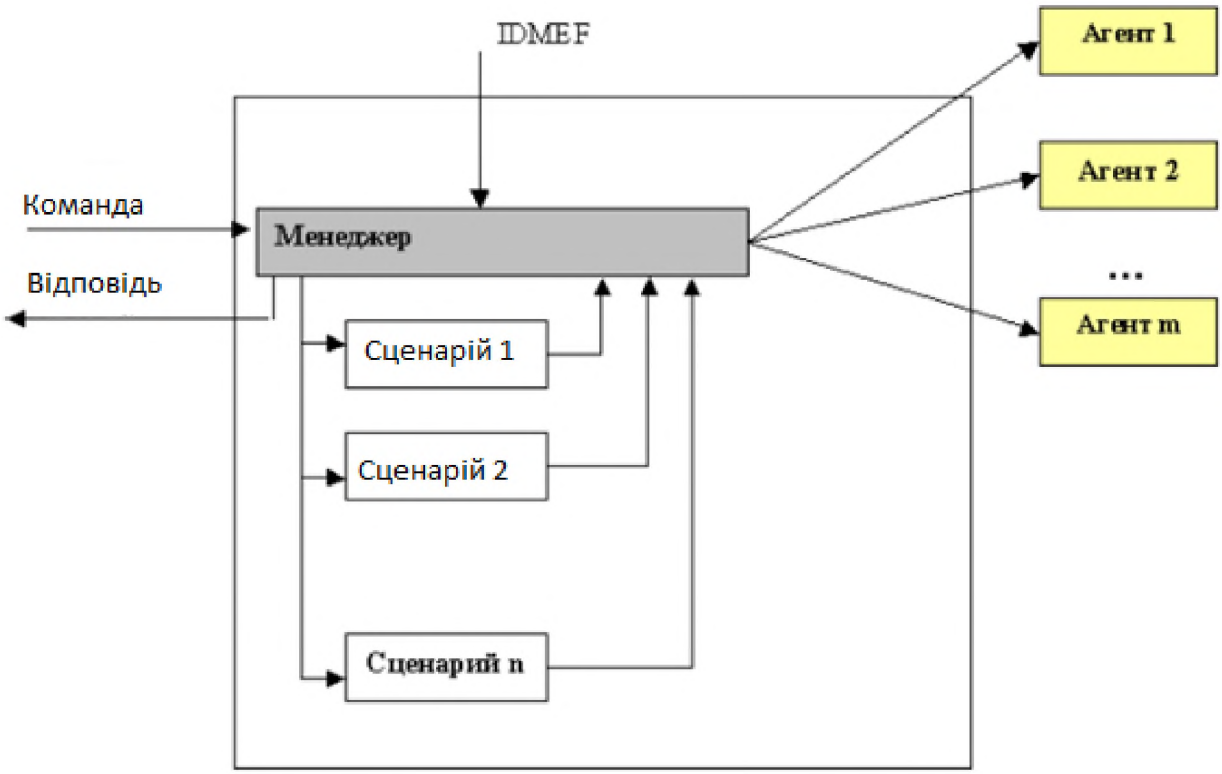
Підсистема реагування є ієрархічною і складається з компонентів трьох типів: вузлових менеджерів, центральних менеджерів і агентів реагування. Вузловий менеджер є підсистемою реагування мережевого сенсора. Центральний є підсистемою реагування консолі управління. Агент реагування встановлений на кожному підключеному засобі реагування і виконує функції управління і настройки даного засобу. Кожен менеджер реагування складається з наступних модулів:

- провайдер подій;
- система аналізу подій;
- модуль зовнішнього управління.

Структура менеджера реагування представлена на рис.2.4.

Провайдер подій

Від мережевих сенсорів менеджер отримує повідомлення в форматі IDMEF. Після отримання повідомлення провайдер подій формує подію і передає її системі аналізу подій для аналізу і формування реакції.[23]



«Рисунок 2.4 – Структура менеджера реагування»

Система аналізу подій

Даний модуль складається з СВП і набору сценаріїв реагування, реалізованих на мові опису сценаріїв СВА. Як і у випадку з підсистемою реагування, алфавітом сценаріїв є повідомлення про атаки, а в кінцевому стані кожного сценарію виконується відповідна йому функція реагування.

Приклад сценарію реагування:

```
scenario HIGHSeverityResponse (NetworkAlertEvent netEv)
{
initial state state0 {}
consuming transition state0-> state0
event NetworkAlertEvent (1)
{
if (netEv.severity == NET_ATTACK_HIGH)
{
sendFWRule (netEv.targetAddress, "REJECT -dir in -s"
```

```
+ Iptoa (netEv.sourceAddress) + "-reject-type icmp-host-unreachable");
}
}
};
```

В прикладі у відповідь на кожне повідомлення про атаку з рівнем небезпеки high (вищий за стандартом IDMEF) формується блокуючу правило для брандмауера IPTables і відправляється відповідному агенту реагування.

2.6. Висновки

У розділі була розглянута модель функціонування розподіленої інформаційної системи, її основні складові, модель поведінки об’єкта та атаки. Розглянуто розпізнавання нормальних та аномальних траєкторій, мова опису автоматів першого та другого класів, а також алгоритми виявлення атак.

Структурно розібрана робота мережевого сенсора, вузлового сенсора та підсистеми реагування.

РОЗДІЛ 3. ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МЕТОДУ ДЛЯ ЕКСПЕРИМЕНТАЛЬНОЇ СВА

В розділі наведені результати випробувань експериментальної СВА на інструментальному стенді з використанням тестових наборів комп'ютерних атак. Наводиться опис інструментального стенду і розробленої методики випробувань, в висновку наводяться результати випробувань.

Так як мета тестування - перевірка ефективності виявлення атак з допомогою гібридного методу аналізу переходів станів, то для тестування обраний обмежений набір атакованих ресурсів і реалізацій атак. В якості атакованих ресурсів були використані: сервер FTP, веб-сервер Apache і сервер SSH openssh.

Автомат опису нормальної поведінки був побудований для сімейства FTP - серверів . Опис даного автомата приведено нижче в описі тестових сценаріїв виявлення. Для інших ресурсів були побудовані описи атак для конкретних реалізацій атак з тестового набору.[4]

Розроблений метод був реалізований у вигляді модулю розпізнавання і реагування на атаки. Модуль було встановлено в систему інформаційної безпеки замість модуля що встановлюється за замовчанням. При дослідженні роботи модуля буде проведено аналіз ефективності модуля у порівнянні з модулем що постачається розробником. Також буде виміряний вплив робочої системи на апаратну платформу.

3.1. Набір тестових прикладів

Для реалізації атаки класу «сканування» був використаний поширений сканер безпеки nmap.

Нижче наведено список використаних в випробуваннях реалізацій атак:

- Атака на вразливість переповнення буфера в FTP
- Атака на web-сервер Apache яка використовує уразливість в обробці кодованих (chunk-encoded) HTTP-запитів;

- Атака що дозволяє отримати привілейований доступ (remote root) через SSH-сервер ОС RedHat Linux 14.0 з настройками за замовчуванням;

- Атака на Microsoft Windows DCOM, що дозволяє отримати віддалений cmd- сеанс з правами адміністратора на ОС;

- Атака Apache squ1rt на web-сервер Apache, що використовує витік пам'яті для переповнення пам'яті сервера і реалізації атаки «відмова в обслуговуванні»;

- Сканування привілейованих портів за допомогою утиліти nmap.[25]

3.2. Тестові сценарії виявлення

В даному розділі описаний набір тестових сценаріїв для дослідження ефективності експериментальної СВА. Найбільший інтерес представляє сценарій, який реалізує автомат другого роду для FTP-сервера , тому що він є прикладом використання запропонованого методу для виявлення невідомих атак.

Розглянуто лише чотири серверні команді, які можуть бути потенційно небезпечні в контексті виконання :
fork(), vfork(), execve () і socketcall ().

На основі аналізу вихідних текстів і трас системних викликів сервера був зроблений висновок, що дані системні виклики в ньому використовуються в наступних ситуаціях[26]:

- після створення керуючого каналу використовується fork, створюється потік для роботи з цим каналом;

- після виконання користувачем команди PASV для передачі даних в пасивному режимі сервер створює слухає сокет, використовуючи системний виклик socketcall з параметром SYS_LISTEN;

- також є команди, після яких, згідно з протоколом, встановлюється канал даних. При цьому використовуються системні виклики socketcall з параметром SYS_CONNECT.[27]

На основі специфікації протоколу FTP і отриманої інформації про реалізації будується сценарій нормального поведінки сервера .

Нагадаємо, що автомат другого роду є структурою наступного виду:

$$(S, P_S, T, P_T, S_0, I, g, q), \qquad (3.1)$$

де S - множина станів;

P_S - множина предикатів станів;

T - функція переходів;

P_T - множина предикатів переходів;

S_0 - початковий стан;

I - множина екземплярів автомата;

g - глобальне оточення;

q - глобальна черга таймера.

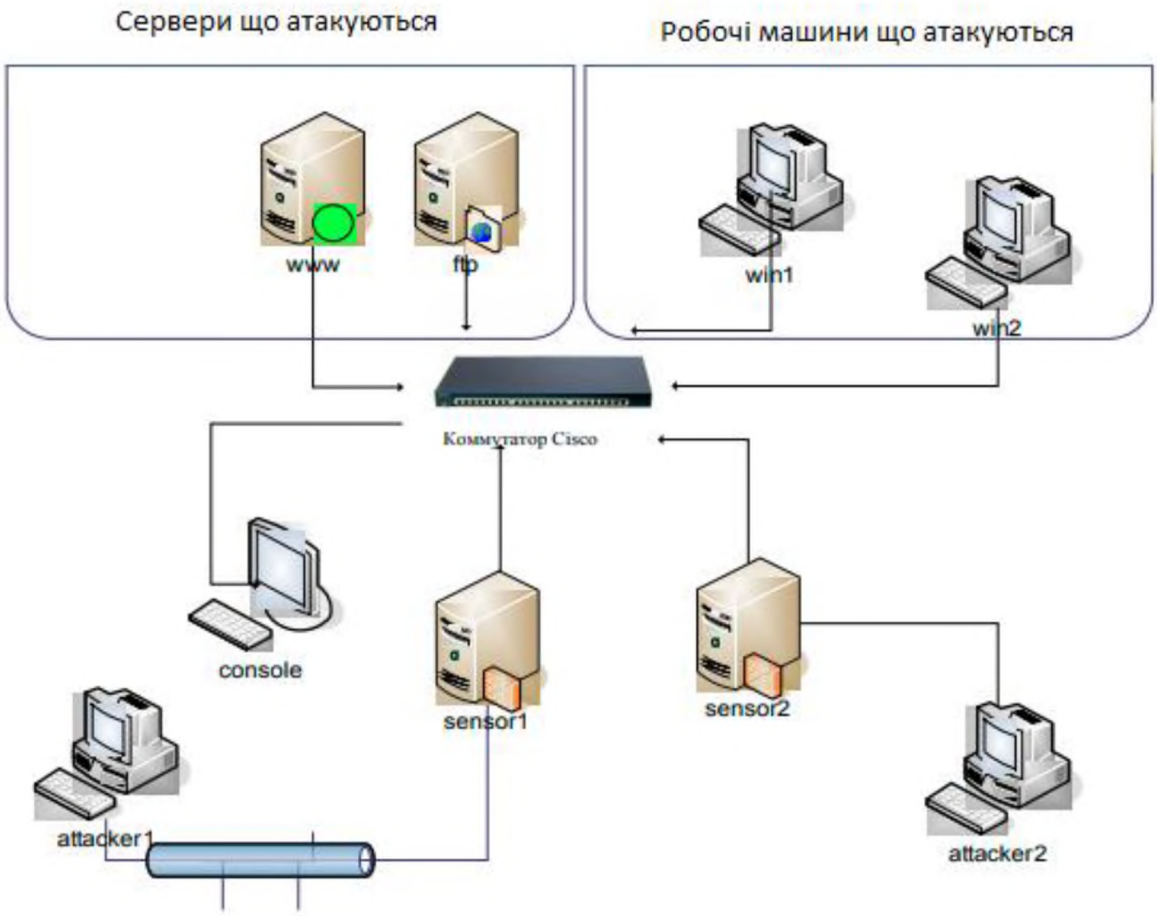
Сценарій нормального поведінки в ході випробувань функціонував в складі мережевого сенсора експериментальної СВА, встановленого на один вузол з сервісом що захищається.

Крім цього сценарію були побудовані сценарії, що реалізують автомати першого роду для кожної з використовуваних атак. Дані сценарії функціонували в складі мережевих і вузлових сенсорів на всіх вузлах випробувального стенду.[28]

3.3. Склад і структура інструментального стенду

Підсистема реагування на мережевих сенсорах налаштована на автоматичне блокування IP-адреси порушника на 1 хвилину.

Архітектура стенду (рис. 3.1)[29], склад програмних ресурсів, що захищаються і набір тестових прикладів на атакуючих вузлах стенду обрані виходячи з необхідності забезпечити тестове покриття прикладами комп'ютерних атак для кожного з розглянутих класів: зовнішні і внутрішні атаки, атаки призначені для користувача, системні та мережеві ресурси, розподілені і нерозподілені атаки.



«Рисунок 3.1 – Структура тестового стенда»

3.3.1. Мережева інфраструктура

При побудові стенду використаний комутатор Cisco. Пропускна здатність кожного каналу становить 1Гбіт / с.

3.3.2. Серверні вузли

Для імітації службової частини мережі, що захищається використані сервери HTTP(www) і FTP(ftp), підключені до комутатора за допомогою 100-мегабітного мережевого інтерфейсу.

На серверах встановлені різні версії ОС Linux і набір прикладних сервісів, які містять програмні уразливості. Використання даних вразливостей в тестових

прикладах атак дозволяє успішно реалізувати атаки на вузли стенда для кожного з розглянутих класів комп'ютерних атак.[30]

Використано наступні прикладні сервіси:

- HTTP: Apache;
- FTP сервер

3.3.3. Робочі станції

Для імітації робочих станцій, що захищається використані вузли з уразливими версіями ОС Windows 7/10. Використано наступні вразливі сервіси - будований сервіс Windows RPC.

3.3.4. Атакуючі вузли

Атакуючі вузли attacker1 і attacker2 функціонують під управлінням ОС Debian Linux. Кожен атакуючий вузол підключений до мережі за допомогою гігабітного мережевого інтерфейсу. На атакуючих вузлах використовується однаковий набір тестових прикладів.

3.4. Порядок випробувань

Випробування проводилися в два етапи:

1. Імітація нормальної мережевої взаємодії між вузлами attacker1,2 і мережевими сервісами на вузлах www, ftp, win1,2. Імітація виконується за допомогою наступних стандартних клієнтів:
    a. *Wget* для HTTP-сервера і FTP-сервера;
    b. *Rdesktop* для Windows-машин (з включеною службою віддаленого доступу);[31]
2. Проведення атак на тлі імітації нормального мережевого взаємодії з вузлів attacker1,2 на вузли www, ftp, win1,2.

На першому етапі проводилася передача файлів з випадковими бінарними даними з вузлів attacker1,2 на вузли www, ftp по відповідним протоколам в циклі.

Даний етап призначений для оцінки рівня помилкових спрацьовувань експериментальної СВА. На другому етапі атаки проводилися на тлі тих же прикладів нормальної поведінки. Атаки проводилися з паузою між послідовними тестами в 1 хвилину (час, необхідний на розблокування вузла на мережевому екрані). В ході проведення атак замірялися (обчислювалися) наступні параметри:

- час атаки;
- клас атаки;
- число проведених атак;
- число повідомлень про атаки;
- успіх / блокування атаки.

3.5. Результати випробувань

Після проведення двох етапів випробувань були отримані результати, що представлені у табл. 3.1-3.2. Відсоток успішних виявлень атак є результатом першого етапу дослідження. Відсоток фальшивих спрацювань є результатом другого етапу, під час проведення якого було використано 50% повідомлень, які не несуть характер атаки.

Таблиця 3.1 - Результати випробувань експериментальної СВА з модулем виявлення атак розробленим в роботі

№	Назва атаки	N тестів	Ресурс	Тип	% успішних	N повідомлень про атаку	% фальшивих спрацювань
1	Squ1rt	400	http	Dos	98	392	1
2	apache-nosejob	400	http	Remote root	100	400	2
3	blackjack	400	Ssh	Remote root	95	400	5
4	nmap	400	0-1024	Remote root	100	400	5

Таблиця 3.2 - Результати випробувань експериментальної СВА з модулем виявлення атак встановленим за замовчанням в системі

№	Назва атаки	N тестів	Ресурс	Тип	% успішних	N повідомлень про атаку	% фальшивих спрацювань
1	Squ1rt	400	http	Dos	90	360	10
2	apache-nosejob	400	http	Remote root	92	368	4
3	blackjack	400	Ssh	Remote root	90	360	7
4	nmap	400	0-1024	Remote root	93	372	5

Для моделювання атаки була побудована мережа, дослідження потребувало цільового сервера, на якому працюють служби HTTP, FTP та SSH.

Для того, щоб генерувати шкідливий тип трафіку, використовується машина з встановленим на неї програмним забезпеченням Kali Linux Metasploit Framework. За допомогою цього програмного забезпечення було згенеровано сім типів шкідливого трафіку (SSH, Dos/DDoS, FTP, HTTP, ICMP, ARP, SCAN).

Далі за допомогою мережевого комутатора обидва трафіки змішувалися і направлялися до мережі. Далі вони надходили до СВА імітуючи атаку. СВА мала перевіряти законний та шкідливий трафік та генерувати сигнали тривоги тоді коли вхідний трафік був під підозрою атаки. Кількість тривожних сигналів мала показати наскільки точно розроблений метод і реалізований в мережевому модулі та може класифікувати зловмисний мережевий трафік, як атаку. Результати випробувань роботи СВА з різними видами трафіку представлені у табл. 3.3-3.4.

Таблиця 3.3 - Результати випробувань експериментальної СВА з модулем виявлення атак

Тип трафіку	N тестів	% успішних	N повідомлень про атаку	% фальшивих спрацювань	% втрачено пакетів даних
SSH	400	96	384	1	1
DoS/DDoS		95	380	1	2
FTP		97	388	2	1
HTTP		98	392	1	5
ICMP		99	396	1	2
ARP		97	388	2	2
Scan		94	376	1	1

Таблиця 3.4 - Результати випробувань експериментальної СВА з модулем виявлення атак розробленим в роботі з різними видами мережевого трафіку

Тип трафіку	N тестів	% успішних	N повідомлень про атаку	% фальшивих спрацювань	Втрачено пакетів даних
SSH	400	98	392	2	1
DoS/DDoS		97	388	2	1
FTP		95	380	4	1
HTTP		98	392	1	1
ICMP		99	396	1	1
ARP		99	396	1	1
Scan		98	392	4	1

Таблиця 3.5 – Значення витрати оперативної пам’яті експериментальної СВА з модулем виявлення атак за замовчанням

Мережевий трафік	Перший чотирьох годинний інтервал	Другий чотирьох годинний інтервал	Середнє значення
	Гбайт	Гбайт	Гбайт
SSH	1,8	1,9	1,9
DoS/DDoS	1,9	2	2

Продовження таблиці 3.5

Мережевий трафік	Перший чотирьох годинний інтервал	Другий чотирьох годинний інтервал	Середнє значення
FTP	1,9	1,9	1,9
HTTP	2	2,2	2,1
ICMP	2	2,1	2,1
ARP	2	2,1	2,1
Scan	2	2,1	2,1

Витрати комп'ютерних ресурсів, що були зафіксовані під час проведення експерименту наведені у табл. 3.5-3.7. Значення витрати оперативної пам'яті та процесорної потужності вузла були зафіксовані на кінці першого та другого 4х годинного інтервалу роботи.

Таблиця 3.6 – Значення витрати оперативної пам'яті експериментальної СВА з модулем виявлення атак розробленим в роботі

Мережевий трафік	Перший чотирьох годинний інтервал	Другий чотирьох годинний інтервал	Середнє значення
	Гбайт	Гбайт	Гбайт
SSH	1,7	1,5	1,6
DoS/DDoS	1,8	1,6	1,7
FTP	1,8	1,8	1,8
HTTP	1,9	1,8	1,8
ICMP	2	1,9	1,9
ARP	2,1	1,8	2
Scan	1,8	1,9	1,8

Таблиця 3.7 – Значення витрати процесорної потужності

Мережевий трафік	Перший чотирьох годинний інтервал		Другий чотирьох годинний інтервал		Середнє значення	
	%		%		%	
	Розроблений модуль	Модуль за замовченням	Розроблений модуль	Модуль за замовченням	Розроблений модуль	Модуль за замовченням
SSH	69	70	83	82	76	76
DoS/DDoS	66	65	74	76	70	71
FTP	40	47	54	53	47	50
HTTP	45	50	49	61	47	56
ICMP	59	70	68	81	63	76
ARP	69	67	79	87	74	77
Scan	51	70	87	90	69	80

3.6. Висновки

Число повідомлень про атаки у всіх тестах перевищує число проведених атак, що в загальному випадку знижує інформативність СВА для оператора, хоча і не впливає на ефективність виявлення атак. В даному випадку підвищену кількість повідомлень про атаки пояснюється надмірністю схеми розміщення компонентів СВА: вузли sensor1 і sensor2 були «транзитними» для всього мережевого трафіку між атакуючими вузлами і захищеними вузлами мережі.

Частка помилкових спрацьовувань для виявлення сканування ресурсів була невеликою. У реальних мережах з інтернет сервісами рівень помилкових спрацьовувань. Таким чином, настройка політики реагування для мережі є важливим завданням.

Також випробування показали, що сценарії, які реалізує автомат другого роду, виявляв тільки успішні атаки на сервіс FTP. У той час як сценарій, який реалізує автомат першого роду для відповідної атаки, виявляє всі спроби реалізації атаки, а не тільки успішні. Дана особливість методу також важлива для

розробки політики реагування PIC, так як реагування може бути ресурсномісткою процедурою в умовах конкретної PIC і для таких систем важливо мінімізувати число реакцій.

Властивість адаптивності до невідомих атак перевірено на прикладі проведення успішних атак на сервіс без завантаження сценаріїв, які реалізують автомати першого роду для відповідних атак. У цих тестах успішні атаки були виявлені сценарієм, що реалізує автомат другого роду (без ідентифікації версії атаки). В тестах завантаження центрального процесора на мережевих сенсорах, на які припадало максимальне обчислювальне навантаження з аналізу мережевого трафіку, не перевищувала 70%. При цьому пікова пропускна здатність становила 100 Мб/сек при отриманні файлів по протоколу HTTP. Варто зазначити, що обчислювальна складність аналізатора зі збільшенням числа сценаріїв росте пропорційно і при великому числі сценаріїв, час на обробку мережевого трафіку може перевищити можливості обчислювача. Дана проблема може бути вирішена різними способами: розпаралелювання, побудова узагальненого сценарію і його мінімізація, або просте зменшення кількості сценаріїв за рахунок вибору тільки тих, які аналізують поведінку реально існуючих об'єктів мережі.

В цілому, експериментальна СВА показала високу ефективність виявлення, низький рівень помилкових спрацювань (одиниці повідомлень на 10Гб мережевого трафіку) і адаптивність до невідомих атак, що і було основною метою даної роботи. Також можна зробити висновок що система виявлення атак з розробленим модулем працює з деяким трафіком краще ніж система з модулем від розробників системи, а споживає менше процесорної потужності і оперативної пам’яті, що є ключовим для апаратних засобів.

РОЗДІЛ 4 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ І ВПРОВАДЖЕННЯ МЕТОДУ ВИЯВЛЕННЯ АТАК

4.1 Обґрунтування витрат на розробку і впровадження комплексу виявлення атак.

Однією з головний цілей захисту інформаційних ресурсів від загроз є мінімізація збитків від порушення інформаційної безпеки підприємства.

Підприємства з розробки програмного забезпечення зараз дуже стрімко розвиваються і виходять за межі внутрішнього ринку і починають співпрацювати з іноземними партнерами. Зазвичай при розробці, а особливо при підтримці програмного забезпечення програмісти мають віддалений, а іноді і локальний (при розробці нового функціоналу у програміста може бути локальна база даних клієнта, ця база представляє собою конфіденційну інформацію) доступ до конфіденційної інформації. Економічно доцільним рішення про захист інформації слід вважати, якщо витрати на забезпечення інформаційної безпеки не перевищують збитків від реалізації загрози. Щоб обґрунтувати економічну доцільність впровадження системи виявлення атак порівняємо величину витрат на впровадження комплексу з величиною можливої шкоди, яку може понести підприємство внаслідок втрати інформаційних ресурсів.

4.2 Розрахунки витрат на розробку і впровадження системи виявлення атак

Витрати на розробку і впровадження на підприємство системи виявлення атак складаються з капітальних (одноразових) витрат, та тих, що необхідні для підтримки дієздатності та технічного супроводження нововведень.

4.2.1 Розрахунок капітальних (фіксованих) витрат

Капітальні інвестиції:

- вартість розробки проекту інформаційної безпеки (розробка схем пристроїв, політики функціонування комплексу тощо);

- витрати на первісні закупівлі апаратного забезпечення;
- витрати на розробку системи виявлення атак;
- витрати на навчання технічних фахівців і обслуговуючого персоналу.

Спершу розрахуємо час, який буде витрачено на створення комплексу:

$$t = t_{\text{тз}} + t_{\text{в}} + t_{\text{а}} + t_{\text{пр}} + t_{\text{опр}} + t_{\text{д}}, \text{годин} \quad (4.1)$$

де $t_{\text{тз}}$ – тривалість складання технічного завдання на розробку комплексу виявлення вторгнень;

$t_{\text{в}}$ – тривалість вивчення ТЗ, літературних джерел за темою тощо;

$t_{\text{а}}$ – тривалість розробки блок-схеми алгоритму створення комплексу;

$t_{\text{пр}}$ – тривалість створення комплексу виявлення вторгнень за готовою схемою;

$t_{\text{опр}}$ – тривалість опрацювання, налаштування та тестування комплексу виявлення вторгнень на апаратній платформі;

$t_{\text{д}}$ – тривалість підготовки технічної документації на комплекс виявлення вторгнень.

Умовна кількість операторів у комплексі:

$$Q = q * c(1 + p), \text{штук} \quad (4.2)$$

де q – очікувана кількість операторів- 27, кількість модулів комплекса котрі мають бути реалізовані та налаштовані;

c – коефіцієнт складності налаштування комплексу-1.9;

p – коефіцієнт корекції програми в процесі її опрацювання – 0.06.

$$Q = 27 * 1.9(1 + 0{,}06) = 54 \text{ штуки}$$

Оцінка тривалості складання технічного завдання на розробку і впровадження комплексу виявлення атак – 2 год.

Тривалість вивчення технічного завдання:

$$t_{\text{в}} = \frac{Q*B}{(75..85)k}, \text{годин} \quad (4.3)$$

де В – коефіцієнт збільшення тривалості етапу внаслідок недостатнього опису завдання, В = 1,35;

k – коефіцієнт, що враховує кваліфікацію адміністратора безпеки і визначається стажем роботи за фахом - до 2 років – 0,8;

$$t_{\text{в}} = \frac{54 * 1.35}{80 * 0.8} = 1.2, \text{години}$$

Тривалість розробки блок-схеми алгоритму створення системи:

$$t_{\text{в}} = \frac{Q}{(20..25)k}, \text{годин} \quad (4.4)$$

$$t_{\text{в}} = \frac{54}{22 * 0.8} = 3, \text{години}$$

Тривалість створення комплексу за готовою блок-схемою:

$$t_{\text{пр}} = \frac{Q}{(20..25)k}, \text{годин} \quad (4.5)$$

$$t_{\text{в}} = \frac{54}{22 * 0.8} = 3, \text{години}$$

Тривалість опрацювання системи:

$$t_{\text{пр}} = \frac{1.5*Q}{(4..5)k}, \text{годин} \quad (4.6)$$

$$t_{\text{в}} = \frac{1.5 * 54}{5 * 0.8} = 20, \text{годин}$$

Тривалість підготовки технічної документації на ПЗ:

$$t_{\text{опр}} = \frac{Q}{(15..20)k} + \frac{0.75*Q}{(15..20)}, \text{годин} \quad (4.7)$$

$$t_{\text{опр}} = \frac{54}{20 * 0.8} + \frac{0.75 * 54}{20} = 5.3, \text{години}$$

Отже час, який буде витрачено на створення системи:

$$t = 2 + 1.2 + 3 + 3 + 20 + 5.3 = 34.5, \text{годин}$$

Розрахунок витрат на створення системи виявлення атак

$$К_{\text{пз}} = З_{\text{зп}} + З_{\text{мч}}, \text{грн} \quad (4.8)$$

Заробітна плата виконавця враховує основну і додаткову заробітну плату, а також відрахування на соціальне потреби (пенсійне страхування, страхування на випадок безробіття, соціальне страхування тощо) и визначається за формулою:

$$З_{\text{зп}} = t * З_{\text{пр}}, \text{грн} \quad (4.9)$$

де t – загальна тривалість створення системи виявлення атак, годин;

$З_{пр}$ – средньогодинна заробітна плата адміністратора безпеки з нарахуваннями, грн/годину.

$$З_{пр} = \frac{З_м}{160}, грн/год. \quad (4.10)$$

де $З_м$ – середня заробітна плата на місяць – 16000 грн.

$$З_{пр} = \frac{16000}{160} = 100, грн/год.$$

Отже заробітна плата виконавця дорівнює:

$$З_{зп} = 34{,}5 * 100 = 3450, грн$$

Вартість машинного часу для налагодження системи визначається за формулою:

$$З_{мч} = t_{опр} * С_{мч} * t_д, грн \quad (4.11)$$

де $t_{опр}$ – трудомісткість налагодження, годин;

$t_д$ – трудомісткість підготовки документації, годин;

$С_{мч}$ – вартість 1 години машинного часу, грн./година.

Вартість 1 години машинного часу ПК визначається за формулою:

$$С_{мч} = P * С_е + \frac{Ф_{поч} * Н_а}{F_р} + \frac{К_{лпз}}{F_р}, грн/год \quad (4.12)$$

де Р – встановлена потужність ПК, 0.031 кВт;

$С_е$– тариф на електричну енергію, 1.68 грн/кВт·година;

$Ф_{поч}$– первісна вартість ПК на початок року, 2000 грн.;

$Н_а$– річна норма амортизації на ПК, 0.1 частки одиниці;

$К_{лпз}$ – вартість ліцензійного програмного забезпечення, грн.;

$F_р$– річний фонд робочого часу (за 40-годинного робочого тижня Fp = 1920 годин).

Отже:

$$С_{мч} = 0.031 * 1.68 + \frac{2000 * 0}{1920} + \frac{2000}{1920} = 1{,}09, грн/год$$

Тож витрати на створення системи виявлення атак становитимуть:

$$К_{пз} = 3450 + 115 = 3565, грн$$

Таким чином, капітальні (фіксовані) витрати на проектування та впровадження проектного варіанта системи інформаційної безпеки складають:

$$К = К_{пр} + К_{пзн} + К_{аз}, грн \quad (4.13)$$

де $К_{пр}$ – заробітна плата адміністратора безпеки, грн;

$К_{аз}$ – вартість закупівлі апаратного забезпечення та допоміжних матеріалів, грн;

$К_{пзн}$ – витрати на встановлення обладнання та налагодження комплексу інформаційної безпеки, грн.

Таблиця 4.1 – Вартість закупівлі апаратного забезпечення

Назва комплектуючих	Вартість грн.
Компьютер RasberryPi model 4	2000
Блок живлення для компьютера RasberryPi	200

Продовження таблиці 4.1

Назва комплектуючих	Вартість грн.
SSD накопичувач Kingston SSD HyperX Fury 3D 240GB	1579
Картка пам’яті Samsung microSDXC 128GB	430
Всього	4209

$$К_{аз}=4209, \text{ грн}$$

$$К = 3450 + 3565 + 4209 = 11224, \text{грн}$$

4.2.2 Розрахунок експлуатаційних витрат

Під експлуатаційними витратами розуміються витрати які будуть при роботі та підтримці системи виявлення атак протягом року.

$$С_к = С_н + С_а + С_з + С_{ел} + С_{ел} + С_{тос}, \text{грн/рік} \quad (4.14)$$

де $С_н$-витрати на навчання персоналу, курси для адміністратора безпеки. Визначаються за даними підприємства – 1000 грн.

$С_а$-річний фонд амортизаційних відрахувань визначається у відсотках від суми капітальних інвестицій – 10% або 1122грн.

$С_з$-річний фонд заробітної плати інженерно-технічного персоналу, що обслуговує систему інформаційної безпеки, складає:

$$С_з = З_{осн} * 12 + З_{дод} * 12, \text{грн/рік} \quad (18)$$

де Зосн, Здод – основна та додаткова середня заробітна плата на 01.12.2019, грн/міс.

$$С_з = 16000 * 12 + 16000 * 0.1 * 12 = 211200, грн/рік$$

$С_{ел}$-вартість електроенергії, що споживається комплексом інформаційної безпеки протягом року визначається за формулою:

$$С_{ел} = P * F_p * С_е, грн/рік \quad (4.15)$$

де P – встановлена потужність апаратури інформаційної безпеки, кВт;

F_p– річний фонд робочого часу системи інформаційної безпеки (визначається виходячи з режиму роботи системи інформаційної безпеки);

$С_е$ – тариф на електроенергію, грн/кВт·годин.

$$С_{ел} = 0.031 * 1920 * 1.68 = 100, грн/рік$$

Витрати на технічне адміністрування та сервіс системи виявлення атак визначаються у відсотках від вартості капітальних витрат 2%.

А саме:

$$С_{тос} = К * 0.02, грн/рік \quad (4.16)$$

$$С_{тос} = 11224 * 0.02 = 224.4, грн/рік$$

Отже, експлуатаційні витрати становлять:

$$С_к = 1000 + 1122 + 211200 + 100 + 224.4 = 213646, грн/рік$$

4.3 Оцінка величини можливого збитку від атаки на вузол або сегмент корпоративної мережі

Кінцевим результатом впровадження й проведення заходів щодо забезпечення інформаційної безпеки є величина відвернених втрат, що розраховується, виходячи з імовірності виникнення інциденту інформаційної безпеки й можливих економічних втрат від нього. По суті, ця величина відображає ту частину прибутку, що могла бути втрачена.

Загалом можливо виділити такі види збитку, що можуть вплинути на ефективність комп'ютерної системи інформаційної безпеки:

- порушення конфіденційності ресурсів (тобто неможливість доступу до них неавторизованих суб'єктів або несанкціонованого використання каналів зв'язку);
- порушення доступності ресурсів (тобто можливість доступу до них авторизованих суб'єктів (завжди, коли їм це потрібно);
- порушення цілісності ресурсів (тобто їхня неушкодженість);
- порушення автентичності ресурсів (тобто їхньої дійсності, непідробленості).

Вихідні дані:

$t_{п}$ = 8 годин – час простою вузла або сегмента корпоративної мережі внаслідок атаки;

$t_{в}$ = 8 години – час відновлення після атаки персоналом, що обслуговує корпоративну мережу;

$t_{ви}$ = 3 години – час повторного введення загубленої інформації співробітниками атакованого вузла або сегмента корпоративної мережі;

$З_{о}$ = 16000 грн/місяць – місячна заробітна плата обслуговуючого персоналу з нарахуванням єдиного соціального внеску;

$З_{с}$ = 10000 грн/місяць – місячна заробітна плата співробітника атакованого вузла або сегмента корпоративної мережі з нарахуванням єдиного соціального внеску;

$Ч_о$ = 1 особа – чисельність обслуговуючого персоналу мережі;

$Ч_с$ = 6 осіб – чисельність співробітників атакованого вузла або сегмента корпоративної мережі;

О = 600 000 грн – обсяг чистого прибутку атакованого вузла або сегмента корпоративної мережі;

I = 1 – число атакованих вузлів або сегментів корпоративної мережі;

Для оцінки економічної доцільності можна не враховувати всі загрози та втрати від них. Якщо ми розглянемо тільки деякі загрози і втрати від них перевищать видатки на впровадження комплексу виявлення вторгнень - це вже буде означати доцільність інвестицій.

Виходячи з загроз змоделюємо 2 ситуації та припустимо що кожна відбудеться раз на рік:

- DDos атака - сервери перестали відповідати і команда підтримки клієнта не може працювати;
- виток інформації - виток конфіденційної інформації клієнта.

Упущена вигода від простою атакованого вузла або сегмента корпоративної мережі становить:

$$В_1 = П_п + П_в + V, \text{грн} \quad (4.17)$$

де $П_п$ – оплачувані втрати робочого часу та простої співробітників атакованного вузла або сегмента корпоративної мережі, грн;

$П_в$ – вартість відновлення працездатності вузла або сегмента корпоративної мережі (переустановлення системи, зміна конфігурації та ін.), грн;

V – втрати від зниження обсягу продажів за час простою атакованого вузла або сегмента корпоративної мережі, грн.

Втрати від зниження продуктивності 6 співробітників з ЗП атакованого вузла або сегмента корпоративної мережі являють собою втрати їхньої заробітної плати (оплата непродуктивної праці) за 8 годин простою внаслідок атаки:

$$П_п = \frac{З_с * 6}{F_p} * t_п, \text{ грн} \quad (4.18)$$

де F – місячний фонд робочого часу (при 40-а годинному робочому тижні становить 160-176 ч).

$$П_п = \frac{1000 * 6}{160} * 8 = 3000, \text{ грн}$$

Витрати на відновлення працездатності вузла або сегмента корпоративної мережі включають кілька складових:

$$П_в = П_{ви} + П_{пв}, \text{ грн} \quad (4.19)$$

де $П_{ви}$ – витрати на повторне введення інформації, грн;

$П_{пв}$ – витрати на відновлення вузла або сегмента корпоративної мережі, грн;

Витрати на повторне введення інформації $П_{ви}$ розраховуються виходячи з розміру заробітної плати 10000 грн 6 співробітників атакованого вузла або сегмента корпоративної мережі, які зайняті повторним введенням втраченої інформації, з урахуванням необхідного для цього часу $t_{ви}$ = 3 години:

$$П_{пв} = \frac{З_с * 6}{F_p} * t_{ви}, \text{ грн} \quad (4.20)$$

$$П_{ви} = \frac{1000 * 6}{160} * 3 = 1125, \text{ грн}$$

Витрати на відновлення вузла або сегмента корпоративної мережі Ппв визначаються часом відновлення після атаки $t_в$ = 8 години і розміром середньо годинної заробітної плати обслуговуючого персоналу:

$$П_{ви} = \frac{з_с * 1}{F_p} * t_в, грн \quad (4.21)$$

$$П_{ви} = \frac{16000 * 1}{160} * 8 = 800, грн$$

Отже сумарна вартість відновлення вузла становить:

$$П_в = П_{ви} + П_{пв} = 800 + 1125 = 1925, грн$$

За 1 робочий день простою, при неможливості працювати відділу підтримки програмних продуктів, внаслідок цього програмне забезпечення у клієнта може працювати з перебоями або не працювати взагалі..

Отже сумарний збиток від такої атаки становить:

$$В_1 = П_п + П_в + V, грн \quad (4.22)$$

Втрати від простою атакованого вузла або сегмента корпоративної мережі, грн.

$$V = \frac{O}{F} * (t_п + t_в + t_{ви}) = \frac{600000}{2080} * (8 + 8 + 3) = 5480, грн$$

Отже сумарний збиток від такої атаки становить:

$$В_1 = \sum_i \sum_n B = 1925 + 3000 + 5480 = 10405, грн$$

В випадку витоку інформації за умовами договору, при витоку конфіденційних даних обов'язковий штраф 70000 грн., а також клієнт має право

розірвати контракт. Також слід враховувати що за умовами договору при простої і непідтримці клієнтів штраф 50000 грн.

Отже:

$$В_2 = П_ш + П_{вк}, грн \quad (4.23)$$

де $П_ш$- витрати на оплату штрафа;

$П_{вк}$- збиток від втрати клієнта.

$$В_2 = 70000 + 50000 + 600000 = 720000, грн$$

Сумарний можливий збиток за умови що кожна атака відбудеться один раз на рік і буде успішною дорівнює:

$$B = B_1 + B_2 = 720000 + 10405 = 730405, грн$$

4.4 Загальний ефект від впровадження системи

Загальний ефект від впровадження системи виявлення атак становить:

$$E = B * R - C, грн \quad (4.24)$$

де B – загальний збиток від атаки на вузол або сегмент корпоративної мережі, грн;

R – очікувана ймовірність атаки на вузол або сегмент корпоративної мережі, частки одиниці. В нашому випадку по експертній оцінці адміністратора безпеки, вірогідність вторгнення складає 40%.

C – щорічні витрати на експлуатацію системи інформаційної безпеки.

$$E = 730405 * 0.4 - 215236 = 76926, \text{грн}$$

4.5 Визначення та аналіз показників економічної ефективності системи інформаційної безпеки.

Коефіцієнт повернення інвестицій ROSI:

$$ROSI = \frac{E}{K}, \text{частки одиниць} \quad (4.25)$$

де Е – загальний ефект від впровадження системи інформаційної безпеки;

К – капітальні інвестиції за варіантами, що забезпечили цей ефект.

$$ROSI = \frac{76926}{11224} = 6.9, \text{частки одиниць}$$

Термін окупності капітальних інвестицій T_0 показує, за скільки років капітальні інвестиції окупляться за рахунок загального ефекту від впровадження системи інформаційної безпеки.

Термін окупності:

$$T_0 = \frac{K}{E}, \text{років} \quad (4.26)$$

$$T_0 = \frac{11224}{76926} = 0.14, \text{років}$$

4.6 Висновки

Виходячи з загальних міркувань та спираючись на сукупність усіх вищезгаданих факторів, економічних розрахунків - можна вважати розробку та

впровадження модуля для системи виявлення атак для компанії є економічно доцільним.

Загальний економічний ефект від впровадження системи інформаційної безпеки організації, що розрахований в цьому розділі, дорівнює додатному значенню - це підтверджує, що корисний економічний ефект перевищує видатки.

Ще одним важливим показником є коефіцієнт повернення інвестицій. В даному випадку він більше одиниці - це демонструє прибутковість вкладень.

При цьому основні економічні показники наступні:

- Капітальні витрати склали : К = 11224 (грн);
- Поточні витрати склали : С = 213646(грн/рік);
- Величина можливого збитку: В=730405 (грн);
- Загальний ефект від впровадження системи: Е = 76926 (грн);
- Рентабельність інвестицій у безпеку складає: ROSI = 6.9 (частки одиниці);
- Термін окупності капітальних інвестицій То = 0.14 (роки).

ВИСНОВКИ

Основні результати роботи полягають в наступному:

Побудована модель функціонування РІС в умовах впливу комп'ютерних атак, в рамках якої завдання виявлення атак зведена до задачі пошуку підланцюгів символів. Дана модель дозволила формально оцінити обчислювальну складність запропонованого в роботі методу і показати його коректність;

Запропоновано гібридний метод виявлення атак на основі методу аналізу систем переходів, що дозволяє виявляти невідомі атаки як відхилення поведінки мережевих об'єктів від нормального стану;

На основі запропонованих методів реалізовано метод виявлення атак для система виявлення атак для ОС Linux і Window.

Дана експериментальна система показала високу ефективність виявлення на випробувальному стенді.

Експериментально показана адаптивність системи до невідомих атак.

Запропонований метод виявлення атак може бути використаний для побудови систем захисту розподілених обчислювальних систем в умовах функціонування в мережах загального доступу, де висока ймовірність появи нових реалізацій атак. Найбільша ефективність методу досяжна в тих системах, де безліч класів об'єктів (використовуваних сервісів і програмного забезпечення) обмежена і не змінюється з часом істотно, що дозволяє використовувати моделі нормальної поведінки для виявлення атак.

ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи

№	Формат	Найменування	Кількість листів	Примітки
Документація				
1	А4	Реферат	3	
2	А4	Список умовних скорочень	1	
3	А4	Зміст	2	
4	А4	Вступ	1	
5	А4	Стан питання . Постановка задачі.	28	
6	А4	Спеціальна частина	31	
7	А4	Економічний розділ	13	
8	А4	Висновки	1	
9	А4	Перелік посилань	3	
10	А4	Додаток А	1	
11	А4	Додаток Б	1	
12	А4	Додаток В	1	
13	А4	Додаток Г	2	
14	А4	Додаток Д	2	
15	А4	Додаток Е	2	
16	А4	Додаток Є	2	

Додаток Б. Перелік документів на оптичному носії.

1. Титульний аркуш.docx;
2. Завдання на кваліфікаційну роботу.docx;
3. РЕФЕРАТ.docx;
4. СПИСОК УМОВНИХ СКОРОЧЕНЬ.docx;
5. ЗМІСТ.docx;
6. ВСТУП.docx;
7. СТАН ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ.docx;
8. СПЕЦІАЛЬНА ЧАСТИНА.docx;
9. ЕКОНОМІЧНИЙ РОЗДІЛ.docx;
10. ВИСНОВКИ.docx;
11. ПЕРЕЛІК ПОСИЛАНЬ.docx;
12. ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи.docx;
13. ДОДАТОК Б. Перелік документів на оптичному носії.docx;
14. ДОДАТОК В. Відгуки керівників розділів.docx;
15. ДОДАТОК Г. Відгук керівника кваліфікаційної роботи.docx;
16. ДОДАТОК Д. Опис автомата першого класу.docx;
17. ДОДАТОК Е. Приклад автомата другого класу.docx;
18. ДОДАТОК Є. Код ініціалізації мережевого інтерфейсу.docx.

ДОДАТОК В. Відгук керівника розділу

Відгук керівника економічного розділу:

Керівник розділу _____ _____

(підпис) (ініціали, прізвище)

ДОДАТОК Г. ВІДГУК

на дипломну роботу магістра на тему:

«Метод виявлення зовнішніх комп'ютерних атак за допомогою моніторингу мережевих об'єктів»

студентки групи 125м-19-2

Кулікової Катерини Ігорівни

Представлена магістерська кваліфікаційна робота присвячена розробці методу виявлення зовнішніх комп'ютерних атак за допомогою моніторингу мережевих об'єктів. Стрімкий розвиток технологій змушує усі компанії переводити, створювати, зберігати та обробляти свою інформацію в електронному вигляді, тому виникає гостра потреба в її захисті. З огляду на це кваліфікаційна робота характеризується своєю актуальністю та своєчасністю.

Основний недолік роботи полягає у невеликій кількості тестів під час дослідження системи через відсутність доступу до більших апаратних можливостей.

Позитивними рисами роботи є системність та послідовність викладення матеріалу, а також використання прогресивного досвіду в сфері кібербезпеки та його практичного застосування.

Студентом було проведено аналіз та порівняння можливих методів розв'язання поставленої задачі, а також запропоновано метод виявлення зовнішніх комп'ютерних атак. Крім того було досліджено існуючі реалізації вирішення подібних задач. Під час виконання магістерської кваліфікаційної роботи студентка Кулікова К.І. проявила грамотним спеціалістом, здатним приймати самостійно складні технічні рішення.

Вважаю, що магістерська кваліфікаційна робота заслуговує на оцінку «добре», а Кулікова К.І. – присвоєння кваліфікації «магістра» з кібербезпеки.

Керівник дипломної роботи, к.т.н., доц. Сафаров О.О._____

Керівник спец. частини ас. Ковальова Ю.В._____

ДОДАТОК Г. ВІДГУК

на дипломну роботу магістра на тему:

«Метод виявлення зовнішніх комп'ютерних атак за допомогою моніторингу мережевих об'єктів»

студентки групи 125м-19-2

Кулікової Катерини Ігорівни

Представлена магістерська кваліфікаційна робота присвячена розробці методу виявлення зовнішніх комп'ютерних атак за допомогою моніторингу мережевих об'єктів. Стрімкий розвиток технологій змушує усі компанії переводити, створювати, зберігати та обробляти свою інформацію в електронному вигляді, тому виникає гостра потреба в її захисті. З огляду на це кваліфікаційна робота характеризується своєю актуальністю та своєчасністю.

Основний недолік роботи полягає у невеликій кількості тестів під час дослідження системи через відсутність доступу до більших апаратних можливостей.

Позитивними рисами роботи є системність та послідовність викладення матеріалу, а також використання прогресивного досвіду в сфері кібербезпеки та його практичного застосування.

Студентом було проведено аналіз та порівняння можливих методів розв'язання поставленої задачі, а також запропоновано метод виявлення зовнішніх комп'ютерних атак. Крім того було досліджено існуючі реалізації вирішення подібних задач. Під час виконання магістерської кваліфікаційної роботи студентка Кулікова К.І. проявила грамотним спеціалістом, здатним приймати самостійно складні технічні рішення.

Вважаю, що магістерська кваліфікаційна робота заслуговує на оцінку «добре», а Кулікова К.І. – присвоєння кваліфікації «магістра» з кібербезпеки.

Керівник дипломної роботи, к.т.н., доц. Сафаров О.О._____

Керівник спец. частини ас. Ковальова Ю.В._____

Додаток Д. Опис автомата першого класу.

```
/ *

* Оголошення назва сценарію. У дужках вказані типи аналізованих подій. Події тільки таких типів відправляються системою прогону автоматів на вхід таким сценарієм.

* /

scenario MyScenario (NetTCPEvent tcpEv, TimerEvent) {

timer t;

// оголошення локального таймера

u_int_16 srcPort; // змінна для зберігання номера порту

/ *

* Оголошення початкового стану, в даному випадку воно порожнє.

* /

initial state state0 {}

/ *

* Оголошення робочого стану.

* /

state state1 {

ids_time tm; // змінна для ініціалізації таймера

tm.sec = 2;

tm.msec = 0;

SetTimer (t, tm); // встановлюємо таймер на значення змінної

}
```

```
/ *
* Перехід з початкового стану в state1 за подією приходу TCP-пакета
* На порт 21.
* /
nonconsuming transition state0-> state1
event NetTCPEvent (tcpEv.tcpDestPort == 21) {// умова переходу
DebugPrint ( «З'єднання на порт FTP \ n");
srcPort = tcpEv.tcpSrcPort; // збереження номера порту джерела
}
/ *
* Перехід-згортка зі стану state1 в початковий стан за подією
* Повторного приходу TCP-пакета на порт 21 протягом 2 секунл.
* /
unwinding transition state1-> state0
event NetTCPEvent (tcpEv.tcpDestPort == 21 &&
tcpEv.tcpSrcPort == srcPort) {
DebugPrint ( «Повторна спроба з'єднання протягом 2 сек. \ N");
}
event TimerEvent (t) {
DebugPrint ( «Протягом 2 сек. Немає даних. \ N");
}
};
```

Додаток Е. Приклад автомата другого класу.

```
/ *
* Сценарій нормального поведінки сервера  версії 1.2.7+
* Вхідні події - пакети TCP, події таймера, події системних
* Викликів fork (), vfork (), exec ().
* /
scenario  (NetTCPEvent te, TimerEvent, NetLinkEvent nl)
{
timer t1; // локальний таймер
ids_time tm; // змінна для ініціалізації таймера
TCPState ts; // змінна для зберігання стану з'єднання
initial state statepre {} // початковий стан
state state0 {} // перше робоче стан
state login {} // стан після отримання команди login
state welcome {} // стан після успішного відправлення відповіді
state waitPASS {} // стан очікування введення пароля
state sendPASS {} // стан після введення пароля
state sendUSER {} // стан після отримання команди USER
state sendSITEHELP {} // стан після отримання команди SITEHELP
state wait221 {} // стан після відправки коду 221
state wait214SITE {} // стан після відправки коду 214
state wait150 {} // стан після відправки коду 150
state sendPORT {} // стан після отримання команди PORT
```

```
state sendPASV {} // стан після отримання команди PASV

...

consuming transition welcome-> wait221 // перехід по отриманню команди QUIT

event NetTCPEvent (te.CheckState (ts) == -1 && bs (te.tcpPayload,

"QUIT"))

{

tm.sec = TIMEOUT;

SetTimer (t1, tm);

d ( "welcome-> wait221", te, 0);

}

unwinding transition welcome-> state0 // перехід по отриманню відповіді 421

event NetTCPEvent (te.CheckState (ts) == 1 && bs (te.tcpPayload, "421"))

{

d ( "unw welcome-> state0", te, 0);

}

}ь илрнм
```

Додаток Є. Код ініціалізації мережевого інтерфейсу

```
int sock = socket (AF_PACKET, SOCK_RAW, htons (ETH_P_ALL));
if (sock <0) {
perror ( "Can not open raw socket");
exit (1);
}
struct ifreq ifr;
memset (& ifr, 0, sizeof (ifr));
strncpy (ifr.ifr_name, ifName, sizeof (ifr.ifr_name));
if (ioctl (sock, SIOCGIFINDEX, & ifr) <0) {
perror ( "Can not obtain interface id");
close (sock);
exit (1);
}
sockaddr_ll
saddr;
memset (& saddr, 0, sizeof (saddr));
saddr.sll_family = AF_PACKET;
saddr.sll_protocol = htons (ETH_P_ALL);
saddr.sll_ifindex = ifr.ifr_ifindex;
if (bind (sock, (struct sockaddr *) & saddr, sizeof (sockaddr_ll)) <0) {
perror ( "Can not bind to the interface");
close (sock);
```

```
exit (1);
}
struct packet_mreq
mreq;
memset (& mreq, 0, sizeof (mreq));
mreq.mr_ifindex = ifr.ifr_ifindex;
mreq.mr_type = PACKET_MR_PROMISC;
if (setsockopt (sock, SOL_PACKET, PACKET_ADD_MEMBERSHIP, & mreq,
sizeof (mreq)) <
0)
perror ( "Can not change promiscuous mode");
```

ПЕРЕЛІК ПОСИЛАНЬ

1. M.B. Shahbaz, X. Wang, A. Behnad, J. Samarabandu, On efficiency enhancement of the correlation-based feature selection for intrusion detection systems, in: 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, 2016.

2. НД ТЗІ 2.5-005-99. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу.;

3. НД ТЗІ 2.5-004-99. Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу.;

4. NIST, E. Aroms, NIST Special Publication 800-94 Guide to Intrusion Detection and Prevention Systems (Idps), CreateSpace, Paramount, CA, 2012.

5. M. Pihelgas, A comparative analysis of open source intrusion detection systems, Tallinn University of Technology, Estonia, 2012 (Master's thesis);

6. Лукацкий А. В. Новые подходы к обеспечению информационной безопасности сети. 2002;

7. Домарєв В. В. Обґрунтування основних функцій системи управління інформаційною безпекою / В. В. Домарєв, Д. В. Домарєв, С. Б Гордієнко. // Вісник Державного університету інформаційно-комунікаційних технологій. – 2012. – Т. 10, No 2. – С. 102-104;

8. Соціальна інженерія // Сучасна західна соціологія: Словник. М.,2015.;

9. Шкарлет С.М. Економічна безпека підприємства: інноваційний аспект: монографія / С.М. Шкарлет. – К.: НАУ, 2007. – 436 с.;

10. N. Heikura, Analyzing offensive and defensive networking tools in a laboratory environment, Tampere University of Technology, Finland, 2015 (Master of Science thesis);

11. Петренко С. А. Политики информационной безопасности / С. А. Петренко, В. А. Курбатов. – М.: Компания АйТи. 2006. – 400 с;

12. Домарєв В. В. Обґрунтування основних функцій системи управління інформаційною безпекою / В. В. Домарєв, Д. В. Домарєв, С. Б Гордієнко. // Вісник Державного університету інформаційно-комунікаційних технологій. – 2012. – Т. 10, No 2. – С. 102-104;

13. Аудит и мониторинг сети. Адаптивное управление безопасностью. [Електронний ресурс]. – Режим доступу : http://www.isecuritv.ru/technologies/audit.php;

14. FAQ: Network Intrusion Detection Systems. Windows Security. Access : http://www.secinf.net/intrusion detection/FAQ Network Intrusion Detection Systemshtml;

15. Про інформацію [Електронний ресурс]. – Режим доступу : http://zakon3.rada.gov.ua/laws/show/2657-12;

16. Про захист персональних даних [Електронний ресурс]. – Режим доступу : http://zakon3.rada.gov.ua/laws/show/2297-17;

17. D.M. Farid and M.Z. Rahman, Learning intrusion detection based on adaptive bayesian algorithm, in: 2008 11th International Conference on Computer and Information Technology, Khulna, 2008;

18. Гапоненко В.Ф. Экономическая безопасность предприятий. Подходы и принципы / В.Ф. Гапоненко, А.Л. Беспалько, А.С. Власков. – М.: Издательство «Ось-89», 2007. – 208с;

19. C. Kolias, G. Kambourakis, A. Stavrou, S. Gritzalis, Intrusion detection in 802, IEEE Commun. Surv. Tutor. 2015;

20. Ghali, N.I.: Feature selection for effective anomaly-based intrusion detection. Int. J. Comput. Sci. Netw. Secur. (IJCSNS) 9, 285–289 (2009);

21. Классификация сетевых атак [Електронний ресурс]. – Режим доступу : http://www.internet-technologies.ru/articles/article_237.html;

22. Смелянский Р.Л., Качалин А.И., “Применения нейросетей для обнаружения аномального поведения объектов в компьютерных сетях”. // Факультет Вычислительной Математики и Кибернетики, МГУ им. М. В. Ломоносова, Москва, 2014;

23. I. Stromberger, N. Bacanin and M. Tuba, "Hybridized krill herd algorithm for large-scale optimization problems," 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI), Herl'any, 2017;

24. Denning, Dorothy E., "An Intrusion Detection Model, " Proceedings of the Seventh IEEE Symposium on Security and Privacy, May 1986, pages 119—131;

25. Debra Anderson, Teresa F. Lunt, Harold Javitz, Ann Tamaru, and Alfonso Valdes, “Detecting unusual program behavior using the statistical component of the next generation intrusion detection system (NIDES)”. // Technical Report SRI-CSL95-06, Computer Science Laboratory, SRI International, Menlo Park, CA, USA, May 2016;

26. Lunt, Teresa F., "IDES: An Intelligent System for Detecting Intruders, " Proceedings of the Symposium on Computer Security; Threats, and Countermeasures; Rome, Italy, November 22-23, 1990;

27. Srinivas Mukkamala, Andrew H. Sung, Ajith Abraham, “Intrusion detection using an ensemble of intelligent paradigms.” // Journal of Network and Computer Applications, 2015;

28. Mattord H., Whitman M. Principles of Information Security // Course Technology. – 2008;

29. J. Victor, M.S. Rao, V.C.H. Venkaiah, IDS — analysis and containment of false positive alerts, Int. J. Comput. Appl. 5 (8) (2010);

30. A. Munoz, S. Sezer, D. Burns, G. Douglas, An approach for unifying rule based deep packet inspection, in: 2011 IEEE International Conference on Communications (ICC), 2011;

31. Scarfone, Karen; "Guide to Intrusion Detection and Prevention Systems (IDPS)" / Scarfone, Karen, Mell, Peter. Computer Security Resource, Austin, 2007;

32. J.S. White, T.T. Fitzsimmons, J.N. Matthews, Quantitative analysis of IDS: Snort and Suricata, Proc. SPIE 8757 (2013);