УДК 004.94
DOI https://doi.org/10.32838/TNU-2663-5941/2020.6-1/14

Кисіль Т.М.
Хмельницький національний університет

РОЗПІЗНАВАННЯ КІНЦЕВИХ ПРИСТРОЇВ КОРПОРАТИВНОЇ МЕРЕЖІ ЗА ПРИНЦИПОМ СВІЙ / ЧУЖИЙ

Недоліками відомих способів організації взаємозв'язку компонентів розподілених систем для виявлення зловмисного програмного забезпечення в корпоративних комп'ютерних мережах є використання централізованої архітектури, що контролюється адміністратором. Це призводить до недостатньо високої достовірності виявлення і локалізації зловмисних дій, бо збір інформації про стан мережі, визначення присутності шкідливих дій і їх блокування здійснюється для обробки єдиним центром.

Імунна система є високорозподіленою, високоадаптивною, самоорганізованою за своєю суттю, зберігає пам'ять про минулі зустрічі та має можливість постійно дізнаватися про нові. Імунна система може надихати вчених і комп'ютерних інженерів. Оскільки обчислювальні проблеми ускладнюються, люди дедалі частіше шукають нові підходи до цих проблем, часто звертаючись до природи по натхнення. Зараз велика увага приділяється імунній системі хребетних як потенційному джерелу такого натхнення. Існує думка, що можна отримати різні ідеї й альтернативні рішення, крім інших біологічно натхненних методів. Подібно до того, як ІС розпізнає чужі молекули, штучна імунна система виявлятиме чужий пристрій на основі порівняння певної інформації із шаблоном за допомогою або правила Хеммінга, або правила r-послідовних збігів.

На жаль, остаточне рішення щодо ідентифікації пристрою корпоративної мережі за принципом свій / чужий спирається на досвід і думку адміністратора мережі, тому є необхідність розробити автоматизовану систему прийняття рішень, яка може ґрунтуватися на нечіткій логіці та спиратися на результати роботи вже наявної СВВ. У роботі запропоновано аналізувати бітовий рядок інформації як основу для подальшої побудови нечіткої системи прийняття рішень.

Ключові слова: *корпоративна мережа, свій / чужий, правило Хеммінга, правило r-послідовного збігу, штучні імунні системи.*

Постановка проблеми. Вибір концепції побудови конкретної корпоративної мережі визначається цілою низкою чинників: затребуваними інформаційними послугами, обсягами переданого трафіку, інфраструктурою і т. д., але існують і загальні вимоги до корпоративних мереж. Мережі підприємств повинні бути побудовані на основі перевірених технологій, що володіють такими якостями, як масштабованість, гнучкість, мультисервісність і, найголовніше, – надійність.

Мережа сучасного підприємства зазвичай повинна підтримувати ряд найбільш затребуваних для бізнесу додатків і керованих сервісів. Насамперед це:

– можливість високошвидкісного доступу до мережі Інтернет;

– створення віртуальних приватних мереж (VPN);

– захист інформації та зберігання даних.

Слід зазначити, що пристрої, під'єднані до корпоративної мережі за допомогою бездротового зв'язку, будуть вважатися підозрілими, і лише їх аномальна поведінка (як то спроба звернення до забороненої IP-адреси, використання недозволених портів з'єднання тощо) буде вирішальною для ідентифікації за принципом свій / чужий.

Крім того, «чужим» може виявитися також і стаціонарний пристрій мережі, який виявляє нетипову поведінку, а «своїм» – пристрій, що під'єднався до мережі, але не виявляє зловмисних дій.

Необхідно щоразу для кожного конкретного пристрою у разі нетипової поведінки приймати рішення – чужий чи свій – і відповідним чином реагувати.

Зазвичай «чужий» пристрій у мережі намагається здійснити вторгнення, причинами якого можуть бути політичні, економічні, злочинні мотиви або навіть випадковість.

Тому необхідно насамперед виявляти такі вторгнення та приймати рішення щодо рівня безпеки такого вторгнення.

Вважається, що багато механізмів, присутніх у біологічній імунній системі, добре підходять для використання у сфері комп'ютерного виявлення вторгнень у вигляді штучної імунної системи (ШІС).

Аналіз останніх досліджень і публікацій. У літературі про ШІС, у якій йдеться про системи виявлення вторгнень (СВВ), моделювання агентів

і лімфоцитів часто об'єднують у загальну сутність детектора [2; 3].

Здатність розрізняти своїх і чужих є, мабуть, найголовнішою рисою імунної системи (ІС). Це робиться шляхом розпізнавання лімфоцитами різних агентів. Розпізнавання агентів у біологічній ІС відбувається, коли між рецепторами на поверхні імунних клітин та епітопами на поверхні патогенних мікроорганізмів встановлюються хімічні зв'язки, збіг на низькому рівні зводиться до узгодження білків або фрагментів білка, що називаються пептидами. Далі слово «пептид» буде використовуватися для представлення як штучних рецепторів, так і штучних агентів.

Використання клональної селекції та соматичної гіпермутації для моделювання дозрівання аффінності у штучних імунних системах, застосованих до мережних систем виявлення вторгнень (МСВВ), було запропоновано, але не реалізовано Хофмейром і Форестом [4]. Хоча було проведено деякі експериментальні роботи, що вивчають роль соматичної гіпермутації в ІС [2].

Постановка завдання. Мета роботи – проаналізувати можливість використання штучної імунної системи для виявлення «чужих» пристроїв, запропонувати структуру такої імунної системи.

Під «чужим» пристроєм будемо розуміти під'єднаний до мережі пристрій, що виявляє аномальні дії, як то спробу несанкціонованого доступу до забороненої частини мережі, виклик або запуск нетипових програм (використання нетипових портів доступу) і т. д. На основі аналізу такої інформації розробити структуру штучної імунної системи

Виклад основного матеріалу дослідження. У ШІС пептиди часто представлені у вигляді рядків довжиною l, що складаються із символів алфавіту, який містить m символів. Цей підхід найчастіше використовують для $m = 2$ (тобто бітових рядків).

Пептиди, що представляють агентів, будуть кодувати деяку інформацію, яка стосується проблемного домену, до якого застосовується ШІС. Оскільки ІС повинна розрізняти своїх і чужих на основі пептидів, ШІС повинна робити це на основі рядків фіксованої довжини l. Кожен такий рядок буде називатися агентом a. Сукупність усіх агентів утворює універсум, $U = \{a_1, a_2, ..., a_n\}$, який містить дві підмножини, що не перетинаються; тобто сукупність своїх, U_S і сукупність чужих, U_N, тож $U = U_S \cup U_N, U_S \cap U_N = \varnothing$. Як зазначено в [4], ШІС тоді стикається із проблемою класифікації; отримавши довільний рядок із U, класифікує його як свій чи чужий. Класифікація на своїх і чужих може також розглядатися як розподіл на нормальних та аномальних.

Ця модель пептидів дотримується вимоги про те, що вся відповідна інформація у проблемній області може бути представлена якимось чином і що повинен існувати певний спосіб компактного кодування узагальнень цієї інформації.

Слід також зазначити, що, коли реальні проблеми відображаються у таких уявленнях, свій і чужий не можуть бути роз'єднаними, оскільки два випадки можуть бути відображені в одному представленні.

Подібно до СВВ ІС також може допускати два типи помилок розпізнавання. Це справедливо і для ШІС. Помилково позитивний результат виникає, коли нормальний агент класифікується як чужий, а помилково негативний – коли аномальний агент класифікується як свій.

Описане вище кодування пептиду також використовується для моделювання рецепторів детекторів у ШІС. ШІС має сукупність детекторів D. Кожен детектор $d \in D$ має покриття C_D, яке описує кількість агентів, які він розпізнає. Якщо детектор d не розпізнає жодних агентів, його покриття $C_D = \varnothing$. З іншого боку, якщо d впізнає всіх інших агентів, його покриття є $C_D = U$; всі агенти в універсумі.

Це представлення пептидів дозволяє ШІС розпізнавати різні агенти за допомогою зіставлення рядків, але одна із приємних особливостей ІС полягає в тому, що вона здатна узагальнювати це зіставлення. Узагальнення своїх і чужих, яке відбувається в ІС, здійснюється за допомогою наближеного збігу рядків.

У найбільш загальній формі проблема наближеного узгодження рядків полягає у пошуку тексту, де виникає заданий шаблон тексту, допускаючи обмежену кількість «помилок» у збігах. Кожна програма використовує іншу модель помилки, що визначає, наскільки різними можуть бути рядки (Navarro 2001). Ці тексти можна розглядати як послідовності символів, складених з алфавіту довжини m.

Зазвичай використовують два такі правила відповідності – правило Хеммінга та правило r-послідовних збігів (рис. 1). Тут основна увага буде зосереджена на правилі r-послідовних збігів, оскільки це правдоподібна абстракція зв'язування рецепторів в імунній системі. ІС дуже ефективна тим, що їй вдається розрізняти своїх і чужих, маючи відносно невеликий набір детекторів.

а) б)

Рис. 1. Зіставлення рядків:

а) за правилом Хеммінга (між рядками довжиною в 16, що складаються із символів бінарного алфавіту, з відповідним обмеженням r=9. Два рядки a та b будуть збігатися для всіх r≤9)

б) за правилом r-послідовних збігів (між рядками довжиною 16, що складаються із символів бінарного алфавіту, з відповідним обмеженням r=5. Два рядки a та b будуть збігатися для всіх r≤5)

І правило збігу Хеммінга, і правило *r*-послідовних збігів контролюються пороговим параметром *r*, де $0 \text{"} r \text{"} l$. Якщо $r = 0$, покриттям d є всі рядки, $C_D = U$, а якщо $r = l$, то покриттям d є один рядок агента a, $C_D = \{a\}$. Чим вище значення *r*, тим конкретніший збіг. І конкретність збігу є аналогічною близькості зв'язування між Ag та лімфоцитом, або детектором.

Слід зазначити, що в ІС відповідність (або розпізнавання) між агентом і лімфоцитом базується на взаємодоповнюючих формах. У ШІС розглядатимуться бінарні рядки та їх «приблизна» рівність.

Правило порівняння Хеммінга базується на відстані Хеммінга між двома рядками. Якщо два рядки *a* і *b* мають однакові біти принаймні в *r* позиціях, вони збігаються (рис. 1.а). Згідно з правилом *r*-послідовного збігу два рядки *a* і *b* збігаються, якщо вони мають однакові біти принаймні в *r* послідовних позиціях (рис. 1.б).

Імовірність збігу за допомогою правила Хеммінга:

Нехай $Hamming_{l,r}(a,b)$ є оператором, який визначає, чи збігаються дві рядки *a* і *b*, обидва довжиною l, використовуючи правило збігу Хеммінга, з обмеженням, що *r* біт попарно рівні.

Тоді ймовірність збігу між двома випадково вибраними рядками a та b становить

$$P(Hamming_{l,r}(a,b)) = 2^{-l}\sum_{i=r}^{l} C_l^i \qquad (1)$$

Імовірність отримують з огляду на те, що 2^{-l} – це ймовірність одиничного збігу, а C_l^i – кількість рядків в U, які мають однакові біти і в позиціях.

Механізм негативного відбору в ІС часто використовується в ШІС для проведення виявлення на основі аномалії. У [1; 4; 5] це змодельовано за умови, щоб дійсними детекторами були ті детектори, які не виявляють самоагентів під час толерування: По-перше, детектор генерується випадковим чином, а це означає, що його рецептори можуть розпізнавати що завгодно. Якщо детектор щось розпізнає під час допуску, він гине. Якщо детектор переживає термін допуску, він стає зрілим і наївним детектором (його називають наївним, оскільки він ще не виявив жодних збудників).

Це використання негативного відбору ґрунтується на припущенні, що, якщо детектор розпізнає що-небудь під час допуску, є своїм. Таким чином, ШІС імпліцитно дізнається, що все, що збігається з його зрілими детекторами та детекторами пам'яті, є чужим.

Оскільки детектори з часом контролюють кілька пептидів, це означає, що за низьких значень *r* детектори AIS будуть відповідати практично будь-чому. З іншого боку, за високих значень *r* детектори будуть відповідати набагато меншому набору агентів.

Оскільки набір детекторів ШІС генерується за допомогою негативного відбору, менші значення *r* призводять до вищої ймовірності відповідності «свій» під час толерування, а більш високі значення *r* зменшують ймовірність відповідності «свій». Таким чином, чим нижче значення *r*, тим більше спроб потрібно ШІС для генерації кожного зрілого детектора. З вищими значеннями *r* необхідна менша кількість спроб генерації детекторів, але для досягнення певного рівня покриття необхідний також більший набір детекторів.

Це призводить до ситуації компромісу, коли для нижчих значень *r* потрібен менший набір детекторів для досягнення певного покриття, тоді як AIS потребує більше спроб для кожного дійсного детектора, який він генерує. Виходячи з ролі, яку поєднання клональної селекції та соматичної гіпермутації відіграє в ІС, передбачається, що такі механізми збільшать різноманітність детекторів і схожість між детекторами й агентами.

СВВ на основі ШІС складається з двох основних частин – основного ядра ШІС і детекторів. Основне ядро ШІС розташоване на шлюзі кожної локальної мережі, а детектори – кожен користувач системи. Кожен із вказаних компонентів складається з агентів, що зіставляють інформацію один від одного, щоб виявити аномалії та вторгнення.

Ціль такої структури – зменшити час виявлення для кожного з'єднання з допомогою надання можливостей детектора (аналіз трафіку та повідомлення про небезпеку) кожному користувачу. Як наслідок, навантаження по обробці трафіку буде розподілятися на кожного користувача – кожен користувач сам відповідає за аналіз власного трафіку. Тому замість того, щоб аналізувати кожен пакет мережі (що створює велику потребу в обчислювальних можливостях і затримку в виявленні), центральне ядро буде обробляти сигнали небезпеки від користувачів мережі.

Основне ядро складається із двох частин: модуля навчання та модуля детекторів-користувачів, обидві частини разом виконують чотири основні завдання:

– створення шаблонів ознак;
– аналіз повідомлень від користувачів;
– запам'ятовування робочих шаблонів;
– розподілення, синхронізацію шаблонів ознак кожного детектора.

Кожен модуль є програмою на комп'ютері користувача чи шлюзові, що виконує одне із завдань (шлюз чи користувач може мати декілька програм одночасно).

Модуль навчання складається із програми-дешифратора та програми навчання, на ньому лежить відповідальність за створення основних випадкових шаблонів ознак на ранніх стадіях роботи системи. Модуль детекторів складається із програми аналізу та програми-диспетчера вторгнень. Перша програма обробляє сигнали від користувачів і в певних випадках запам'ятовує шаблони, на які користувач зреагував, схрещує їх для виконання генетичного алгоритму, друга програма відповідає за розповсюдження та синхронізацію шаблонів між користувачами-детекторами.

Перед оцінкою системи відбувається її попереднє навчання та налаштування параметрів. Попереднє навчання ШІС відбувається з допомогою використання набору безпечних (своїх) даних і небезпечних (чужих) даних. Для обробки пакетів трафіку їх спочатку необхідно розшифрувати та перетворити в оброблену інформацію – цим займається програма-дешифратор. Інформація містить такі поля, як ір надсилача, ір отримувача, порт надсилача, порт отримувача, протокол, розмір пакету. Ця інформація дістається з пакетів і перетворюється в послідовності зі 112 бітів. Табл. 1 показує правильно розшифрований приклад елементів інформації, що отримується, та їх розміром (у бітах).

Після дешифрування всіх тренувальних наборів інформації в бітові послідовності їх передають у програму навчання, що використовується для створення шаблонів для детекторів. Алгоритм негативного відбору використовується для створення першого покоління шаблонів.

Спочатку створюється і перевіряється «молодий» випадковий набір бітових послідовностей-шаблонів на базовому наборі тестових даних. Якщо шаблон спрацьовує на «своїх» пакетах, то він заміняється новим і так доти, поки шаблон не перестане реагувати на свої пакети. Далі відбувається наступний крок алгоритму негативного відбору, що відсіює шаблон, який не реагує на жоден із «чужих» пакетів. Все, що не відсіялося, додається до результуючого набору шаблонів. Цей процес повторюється, поки кожен із чужих пакетів не збігатиметься із хоча б трьома шаблонами з результуючого набору шаблонів. Для порівняння послідовностей використовується *r*-бітовий збіг.

Процес навчання основних детекторів вказаний на рис. 2.

Після навчання всі детектори мають отримати набір шаблонів. Цю роль бере на себе програма-диспетчер, яка синхронізує зміни шаблонів, а також встановлює для них шаблони ознак. Диспетчер отримує сигнали небезпеки та перенаправляє їх до програми-аналізатора для обробки.

Як тільки у користувача відбудеться вторгнення і користувач його помітить за шаблоном, він відправить повідомлення, що включатиме підозрілий пакет. Таким чином, інформація про вторгнення, наприклад, кількість детекторів, що помітили небезпеку, їх ставлення до підозрілого

Таблиця 1

Приклад можливої інформації з пакету

Назва поля	Мінімум – максимум	Кількість бітів
IP отримувача	0.0.0.0–255.255.255.255	32 біти
IP надсилача	0.0.0.0–255.255.255.255	32 біти
порт отримувача	0–65535	16 бітів
порт відправника	0–65535	16 бітів
час продовження	0–4096 секунд	12 бітів
протокол	0–16	4 біти

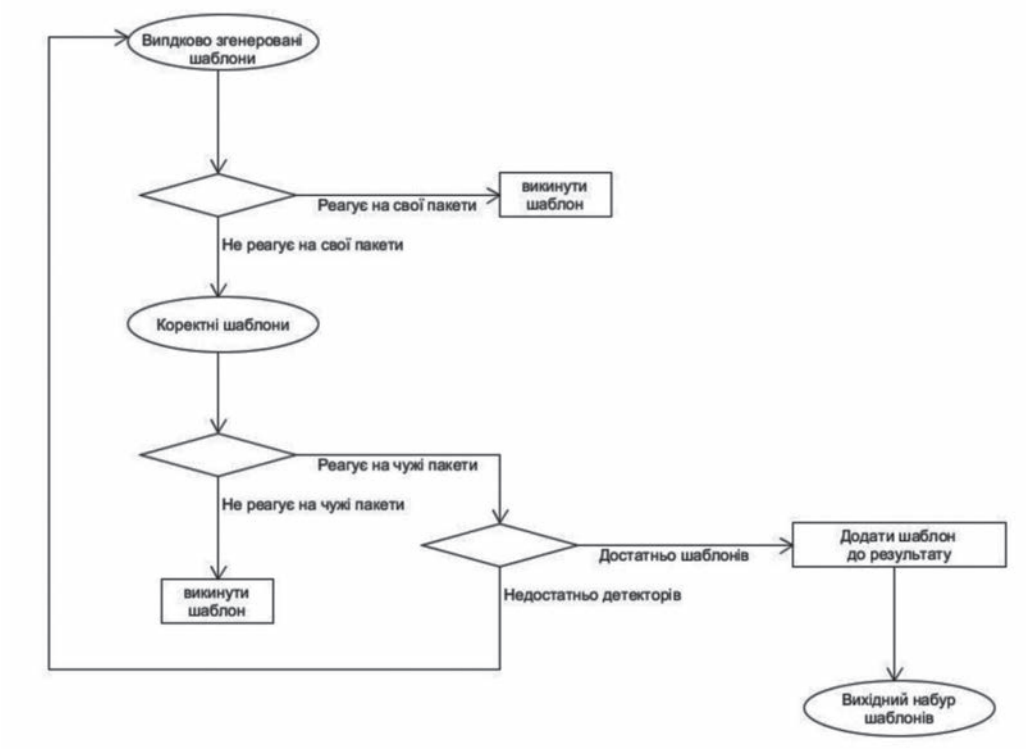

Рис. 2. Навчання детекторів

пакету та їх профіль буде надіслано до основного ядра СВВ на аналіз та обробку. Програма-аналізатор із модуля-детектора цим і займається. Якщо кількість підозрілих пакетів виходить за межі норми, то відбувається покращення шаблонів. Генетичний алгоритм виконується для створення шаблонів, що краще виділяють певні види аномалій. Водночас аналізатор спробує заблокувати пакети певного типу через фаєрвол. Якщо кількість повідомлень про небезпеку менша за нижню границю небезпеки, то пакет все одно додасться для подальшої обробки в пасивному режимі. Після створення нових шаблонів вони будуть передані програмі-диспетчеру для розповсюдження між вузлами-користувачами.

Коли задіяні шаблони надсилаються в аналізатор, то генетичний алгоритм використовується для створення з них шаблонів, що будуть скопійовані для початкового покоління для відбору.

Формула, яка визначає, чи нам треба відбирати шаблон для генетичного алгоритму:

[Мінімальна оцінка принадності для відбору]=
=[сума оцінок шаблонів] / [кількість шаблонів]

Шаблони, оцінка яких вища за мінімальну, використовуються для створення наступного покоління за допомогою генетичного алгоритму. Кожен шаблон може копіюватися певну кількість разів – кількість визначається за формулою:

[кількість копій]=ціла частина(10*[оцінка шаблону] / [сума оцінок шаблонів])

Після виконання клонувань виконується генетичний алгоритм – вибрані детектори проходять через операції кросоверу, мутації та репродукції певну кількість поколінь. У кожному поколінні визначається нова сума оцінок шаблонів і вибирається новий кандидат на додавання. І якщо його оцінка менша за максимальну з початкового шаблону, то генетичний алгоритм зупиняється, і кандидат на додавання розповсюджується між користувацькими вузлами-детекторами. Якщо через певну кількість поколінь не можна зробити кращий шаблон, то розповсюджується кращий зі створених шаблонів.

Для покращення механізму СВВ і збільшення ефективності шаблони розповсюджуються на всі вузли в мережі. Це також зумовлює простоту і розширюваність такої системи. У системі присутні два типи вузлів-детекторів: вузли пам’яті й активні детектори. Дешифратор використовується для перетворення пакетів для аналізу активними детекторами.

Вузли пам’яті дозволяють робити адаптивну відповідь ШІС на вторгнення. Вузли пам’яті містять набір шаблонів, що створюється і змінюється, використовуючи генетичний алгоритм. Аналізатор виконує раніше вказаний генетичний

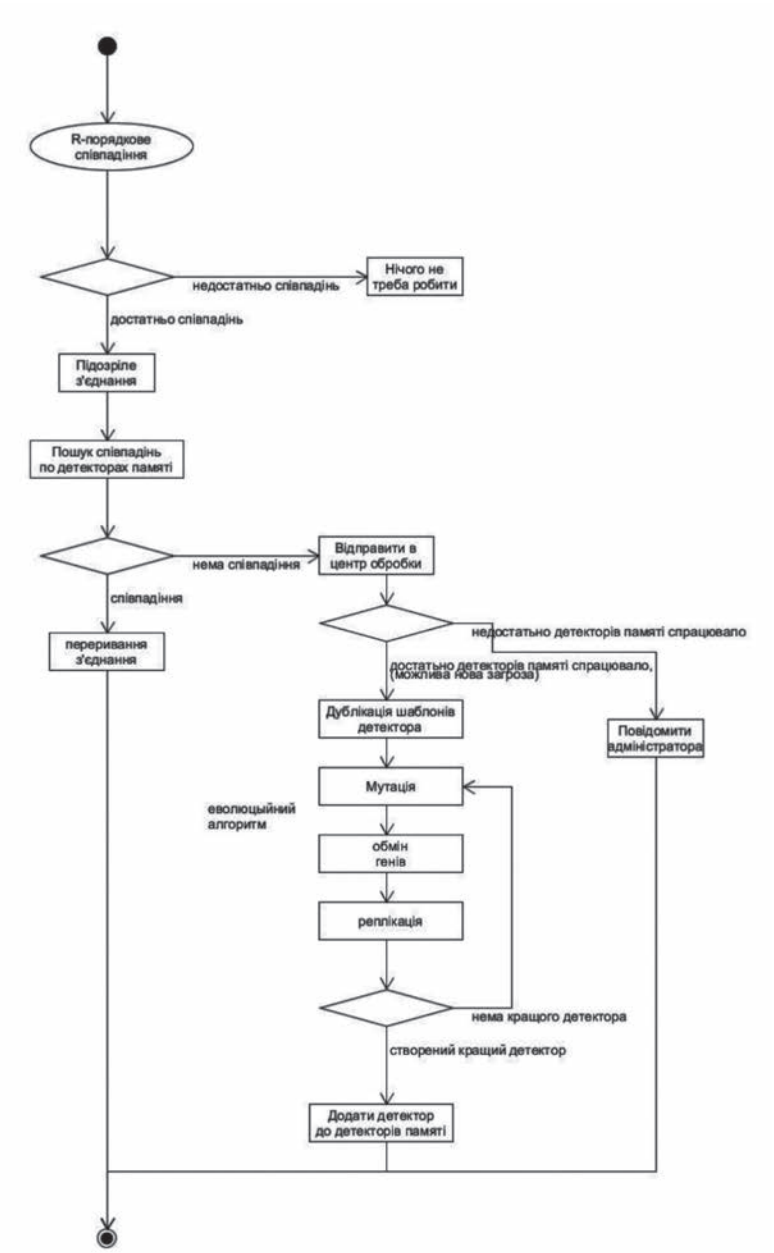

Рис. 3. Блок-схема процедури виявлення

алгоритм, який дозволяє детекторам краще ідентифікувати втручання. Використання вузлів пам'яті дозволяє зменшувати час відповіді та краще реагувати на раніше помічені види втручань. Також такий підхід збільшує ефективність СВВ, зменшуючи час обробки пакетів. Вузли пам'яті також добре працюють для зменшення кількості неправильних позитивних і неправильних негативних відкликів. Як тільки аномалія була помічена на вузлі, і будь-який шаблон із вузлів пам'яті підійде під трафік у мережі, підходящий трафік буде направлено на аналіз в основне ядро ССВ. Весь процес аналізу із серверної та клієнтської сторони показаний на рис. 3.

Активний детектор містить набір шаблонів, що точно розрізняють трафік за схемою свій-чужий. Всі вхідні пакети перевіряються цими детекторами. Якщо будь-який пакет визнано аномальним за будь-яким шаблоном, то вказаний пакет передається далі на аналіз в основне ядро СВВ для обробки. Кількість шаблонів, задіяних на підозрілому пакеті, оцінка кожного із задіяних шаблонів, властивості пакету – все це необхідно щоразу передавати для аналізу в основне ядро. Межа підозрілості пакету – властивість, що дозволяє збільшувати точність визначення втручання і відсіяти неправильні позитивні спрацьовування. Якщо кількість шаблонів, зачеплених при аналізі, більша за мінімальну межу підозрілості, то сесія із цим пакетом буде примусово відключена фаєрволом.

Висновки. Попереднє навчання ШІС відбувається за допомогою використання набору безпечних (своїх) даних і небезпечних (чужих) даних. Для обробки пакетів трафіку їх спочатку необхідно розшифрувати та перетворити в оброблену інформацію, яка містить такі поля, як ір надсилача, ір отримувача, порт надсилача, порт отримувача, протокол, розмір пакету. Ця інформація дістається з пакетів і перетворюється в послідовності зі 112 бітів. Надалі цю інформацію використаємо для нечіткої системи управління, щоб дати відповідь на питання, чи цей пристрій свій чи чужий.

Список літератури:

1. Hightower Ron, Stephanie Forrest, and Alan S. Perelson. The Baldwin Effect in the Immune System: Learning by Somatic Hypermutation. *Adaptive Individuals in Evolving Populations: Models and Algorithms*, Addison-Wesley Publishing Company, Reading Massachusetts. 1996. P. 159–167.
2. Timmis J. Artificial immune systems: today and tomorrow. *Natural Computing*. № 6 (1). P. 1–18. March 2007.
3. Литвиненко В.І. Побудова штучних імунних систем. *Наукові праці. Комп'ютерні технології.* 2010. Вип. 121. Т. 134. С. 166–178

4. Hofmeyr S., Forrest S. Architecture for an Artificial Immune System. *Evolutionary Computation.* 2000. № 8 (4). P. 443–473.

5. Lysenko S., Savenko O., Bobrovnikova K., Kryshchuk A., Savenko B. Information technology for botnets detection based on their behaviour in the corporate area network. *Communications in Computer and Information Science*. 2017. Vol. 702. P. 83–95.

Kysil T.M. RECOGNITION OF END DEVICES OF CORPORATE NETWORK ON THE PRINCIPLE OF SELF / NONSELF

The disadvantages of known methods of interconnecting distributed system components to detect malware in corporate computer networks are the use of a centralized architecture controlled by the administrator. This leads to insufficient reliability of detection and localization of malicious actions, because the collection of information about the state of the network, determining the presence of malicious actions and blocking them is carried out for processing by a single center, which can be slowed down by transmitting collected data to this center. also the impact on his work as a network administrator.

The immune system is highly distributed, highly adaptable, self-organized in nature, preserves the memory of past meetings and has the opportunity to constantly learn about new meetings. From a computational point of view, the immune system can inspire scientists and computer engineers. As computational problems become more complex, people are increasingly looking for new approaches to these problems, often turning to nature for inspiration. Much attention is now being paid to the vertebrate immune system as a potential source of such inspiration, where it is thought that different ideas and alternative solutions can be obtained in addition to other biologically inspired methods. Given this increase in attention to the immune system, it seems appropriate to explore this area in some detail. By analogy, how the IS recognizes foreign molecules analyzed as an artificial immune system will detect a foreign device based on comparing certain information with a pattern using either the Hamming rule or the r-sequence matching rule.

Unfortunately, the final decision on identifying a corporate network device based on one's own experience is based on the experience and opinion of the network administrator, so there is a need to develop an automated decision-making system that can be based on fuzzy logic and based on existing IDS. In this paper, it is proposed to analyze the bit string of information as a basis for further construction of a fuzzy decision-making system.

Key words: *corporate network, self / nonself, Hamming match rule, r-contiguous match rule, artificial immune systems.*