НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова
праця на правах рукопису

Балакін Сергій В'ячеславович

УДК 004.056.53 (043.5)

ДИСЕРТАЦІЯ
МЕТОДИ ТА ЗАСОБИ ПІДВИЩЕННЯ ДОСТОВІРНОСТІ ІДЕНТИФІКАЦІЇ НЕСАНКЦІОНОВАНИХ ДІЙ ТА АТАК В КОМП'ЮТЕРНІЙ МЕРЕЖІ

05.13.05 – комп'ютерні системи та компоненти

Дисертація на здобуття наукового ступеня
кандидата технічних наук

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело


С. В. Балакін

Науковий керівник:
Жуков Ігор Анатолійович
доктор технічних наук, професор,
заслужений винахідник України

Київ – 2018

АНОТАЦІЯ

Балакін С. В. Методи та засоби підвищення достовірності ідентифікації несанкціонованих дій та атак в комп'ютерній мережі. **–** Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.05 – комп'ютерні системи та компоненти. – **Національний авіаційний університет, МОН України, Київ, 2018.**

Дисертаційну роботу присвячено вирішенню актуального науково-технічного завдання – підвищенню достовірності ідентифікації несанкціонованих дій і атак в комп'ютерній мережі.

Для ефективної, надійної та високошвидкісної ідентифікації несанкціонованих дій та атак в комп'ютерній мережі потрібно впроваджувати та використовувати методи, основані як на штучних імунних системах, так і на можливості діагностування вторгнень. Такий підхід дозволить підвищити ефективність ідентифікації несанкціонованих дій і дасть можливість автономно виявляти підозрілу активність.

Дана робота описує один із варіантів недоліків захисту від несанкціонованих дій у комп'ютерні мережі, який можна впровадити в уже існуючі системи.

Аналіз літературних джерел показав, що відомі методи протидії несанкціонованим діям (НД) обмежені й малоефективні, а також потребують постійної модернізації та оновлення для підтримки роботи з новими несанкціонованими діями. Це свідчить про те, що задача підвищення ефективності виявлення НД в комп'ютерних мережах є актуальною.

Способи та методи виявлення вторгнень в комп'ютерних мережах описуються науковцями такими, як: Я. Янг, І.В. Котенко, М. Якобссон, С. Ветзель, В.І. Городецький, Н. Репп, Р. Хекманн, Г. Вігна, К. Вішал, Р.А. Кеммерер, К. Крюгель, Р. Вегнер, А.П. Демпстер, А.А. Романюхи, Л.Н. Кастро та ін.

Незважаючи на прогрес і численні роботи, присвячені виявленню НД в комп'ютерні мережі, треба відзначити, що вони мають особливості, які значно обмежують ефективність сучасних інструментальних засобів. Тому питання розробки методів і моделей виявлення НД набуває актуальності. Завдання, які при цьому виникають, зумовили напрямок досліджень дисертаційної роботи.

Сформульовано необхідні критерії та вимоги для забезпечення своєчасного виявлення вторгнень у комп'ютерні мережі. Визначено основні напрями розвитку сучасних методів аналізу вторгнень і можливості автономного виявлення НД. Аналіз сучасних вторгнень показав доцільність розроблення методів, котрі дадуть змогу виявляти як відомі, так і нові НД.

Проведено порівняльний аналіз моделей і методів, які можливо використовувати при розпізнаванні НД в комп'ютерній мережі. Порівняно характеристики методів і вказано на їх сильні та слабкі сторони. Сформовано вимоги до вибраних методів на основі збереження швидкодії та можливості автономного виявлення НД (без використання і звернення до сигнатурних баз даних). Розглянуто методи штучних імунних систем, котрі дають змогу підтримувати автономне виявлення нових НД при високій швидкості обробки інформації та адаптивності. Виявлення НД за допомогою діагностування дає можливість розширити спектр потенціальних вторгнень за допомогою використання операторів ТДШ. При поєднанні різних технологій при використанні методів ТДШ можливо досягнути високої швидкості та надійності виявлення НД в комп'ютерних мережах.

Описано основні недоліки і обмеження розглянутих інструментів. Сформовано основні задачі дослідження, визначені шляхи виявлення несанкціонованих дій у комп'ютерні мережі засобами ШІМ і ТДШ. Визначено основні напрямки та сформовані основні завдання дисертаційного дослідження.

У роботі визначено методи виявлення несанкціонованих дій і атак в комп'ютерній мережі за рахунок використання засобів штучних імунних систем та діагностування на основі теорії Демпстера-Шафера, котрі дають можливості ефективно протидіяти вторгненням. Досліджено можливості використання

операторів імунних систем для моделювання роботи запропонованих методів. На основі цих властивостей запропоновано процедури ідентифікації несанкціонованих дій і атак в комп'ютерній мережі.

Основні наукові результати заключаються в тому, що: *удосконалено* модель виявлення несанкціонованих дій в комп'ютерній мережі, в якій розпізнавання відбувається за допомогою аналізу поведінкових ознак, що дає можливість автономно розпізнавати невідомі вторгнення і мінімізувати помилкові спрацьовування; *отримав подальший розвиток* метод виявлення вторгнень у комп'ютерні мережі, який базується на використанні операторів штучних імунних мереж для побудови структурованої мережі антитіл, що зі свого боку позитивно впливає на швидкодію і достовірність автономного виявлення як відомих, так і нових вторгнень; *вперше запропоновано* представлення моделі виявлення вторгнень в комп'ютерній мережі засобами інспектування стану системи на виникнення аномальної поведінки, використовуючи дерево діагностування, що дозволяє відстежувати активність системи й симптомізувати дії користувача, а шляхом введення нових елементів і діапазонів роботи досягається підвищення достовірності виявлення атак; *вперше розроблено* метод розпізнавання несанкціонованих дій засобами діагностування на основі операторів теорії Демпстера-Шафера, де на відміну від існуючих пропонується відстежувати часові фрагменти на заданих діапазонах часу і з них, за допомогою операторів злиття, формувати діагнози, за рахунок чого досягатиметься можливість автономного виявлення невідомих системі вторгнень.

Сформульовано вимоги до несанкціонованих дій у системі та описано методи їх виявлення та обробки. Розглянуто вирішення завдань, пов'язаних з організацією виявлення вторгнень шляхом діагностування за допомогою використання інструментів теорії Демпстера-Шафера, що забезпечують оптимальний розподіл ресурсів системи та гарантують виявлення невідомих системі вторгнень, що не входили до набору наперед заданих несанкціонованих дій.

Проведено дослідження ефективності методу виявлення НД в комп'ютерній

мережі на основі штучних імунних систем і діагностування. На основі аналізу отриманої інформації зроблений наступний висновок: при коректній навчальній вибірці та вірному виборі параметрів навчання метод ШІМ має однаково високу достовірність виявлення нових НД як і метод діагностування. ШІМ потребує додаткового часу на утворення навчальної вибірки, але це дозволяє системі швидше реагувати на нові види НД і знизити кількість помилкових спрацювань. Метод діагностування менше навантажує систему користувача, але частіше визначає підозрілу активність як НД. Результати порівняльного аналізу НД показують, що запропоновані методи перевершують відомі антивірусні продукти, використані в порівняльному тесті та здатні виявити невідомі НД.

Проведено порівняльний аналіз запропонованих рішень, котрий експериментально підтвердив, що дані методи підвищують достовірність ідентифікації несанкціонованих дій та атак в комп'ютерній мережі.

Розроблені в дисертаційній роботі методи і моделі можуть бути використані для підвищення ефективності виявлення несанкціонованих дій в комп'ютерній мережі та доведені до рівня програмних засобів. Експериментальні дослідження підтверджують основні положення, що виносяться на захист. Розроблені методи підвищення ефективності виявлення вторгнень можуть бути застосовані для забезпечення роботи засобів керування мережею, а також відповідних систем операційної підтримки. Новизну запропонованих рішень захищено патентами на корисну модель України № 110330 та № 123634.

Результати дисертаційної роботи впроваджено в навчальний процес на кафедрі комп'ютерних мереж та систем Національного авіаційного університету та використовуються в навчальних курсах: «Телекомунікаційні технології комп'ютерних мереж» і «Архітектура комп'ютерів» з 2017 - 2018 рр., що підтверджено відповідним актом про впровадження. Результати роботи використані для підвищення ефективності діагностування несанкціонованих дій в комп'ютерній мережі ТОВ «Газбудсервіс» (акт впровадження від 26.06.2017 р.), а також для аналізу несанкціонованих дій в комп'ютерній мережі ДП «Коропласт» (акт впровадження від 26.06.2017 р.).

Ключові слова: комп'ютерна мережа, трафік, несанкціоновані дії, виявлення вторгнень, діагностування, штучна імунна система, ідентифікація.

SUMMARY

Balakin S. Methods and ways of increasing the reliability of the identification of unauthorized actions and attacks in the computer network. – Manuscript.

Master's thesis initiated to get degree a candidate of technical science on speciality 05.13.05 – computer systems and components. – National Aviation University, MES of Ukraine, Kiev, 2018.

The thesis is devoted to solving the actual scientific and technical problem - increasing the reliability of identification of unauthorized actions and attacks in the computer network.

For effective, reliable and high-speed identification of unauthorized actions and attacks in a computer network, methods should be implemented and used based on both artificial immune systems and the ability to diagnose intrusions. Such an approach will increase the effectiveness of identifying unauthorized actions and will provide an opportunity to autonomously detect suspicious activity.

This work describes one of the options for defect protection from unauthorized actions in computer networks, which can be implemented in existing systems.

The analysis of literary sources has shown that known methods of counteracting intrusions are limited and ineffective, and also need constant updating and updating to support work with new unauthorized actions. This indicates that the task of increasing the effectiveness of detection of intrusions in computer networks is relevant.

Methods and methods for detecting intrusions in computer networks are described by scholars such as Y. Yang, I. V. Kotenko, M. Jacobsson, S. Vetzel, V.I. Gorodetsky, N. Repp, R. Heckmann, G. Vigna, C. Vishal, R.A. Kemmerer, K. Kruegel, R. Wegner, A. P. Dempster, A. A. Romanyukh, L.N. Castro et al.

Despite the progress and numerous work devoted to discovering intrusions in computer networks, it should be noted that they have features that significantly limit the effectiveness of modern tools. Therefore, the issue of developing methods and models

of detection of intrusions becomes relevant. The tasks that arise in this context have led to the direction of research in this work.

The necessary criteria and requirements are formulated for ensuring timely detection of intrusions in computer networks. The basic directions of development of modern methods of analysis of intrusions and possibilities of autonomous detection of intrusions are determined. An analysis of modern intrusions has shown the feasibility of developing methods that will be able to detect both known and new intrusions.

A comparative analysis of models and methods that can be used for intrusion recognition in a computer network is carried out. The comparative characteristics of the methods are indicated on their strengths and weaknesses. The requirements for the selected methods are formed on the basis of maintaining the speed and the ability to independently identify the intrusion (without the use and access to signature databases). Detection of intrusions by means of diagnostics enables to expand the spectrum of potential intrusions by using Dempster-Shafer operators. When combining different technologies with the use of Dempster-Shafer methods it is possible to achieve high speed and reliability of detection of intrusions in computer networks.

The main drawbacks and limitations of the considered tools are described. The basic research tasks are formed, ways of detecting unauthorized actions in computer networks by means of artificial immune systems (AIS) and Dempster-Shafer theory are determined. The basic directions and the basic tasks of the dissertation research are determined.

The work defines methods for detecting unauthorized actions and attacks in a computer network through the use of artificial immune systems and diagnostics based on the Dempster-Shafer theory, which makes it possible to effectively detect intrusions. The possibilities of using the operators of immune systems for modeling the work of the proposed methods are explored. Based on these properties, procedures are proposed for identifying unauthorized actions and attacks in a computer network.

The main scientific results are that: the model of detection of unauthorized actions in a computer network is improved, in which recognition is carried out by means of analysis of behavioral features, which enables to independently recognize unknown

invasions and minimize false positives; received further development of the method of detecting intrusions in computer networks, which is based on the use of operators of artificial immune networks to build a structured network of antibodies, which, on its part, positively affects the speed and accuracy of the autonomous detection of both known and new intrusions; for the first time, the proposed model for detecting intrusions in the computer network is to inspect the state of the system for the occurrence of abnormal behavior using a diagnostic tree that allows monitoring the activity of the system and symptomizing the user's actions, and by introducing new elements and ranges of work it is achieved to increase the accuracy of detection of attacks; for the first time, a method for recognizing unauthorized actions by means of diagnosing on the basis of the operators of the theory of Dempster-Shafer, where, unlike the existing ones, it is suggested to trace time slots in the given time ranges and from them, with the help of the merger operators, to form the diagnoses, which will allow the autonomous detection of intrusions.

Requirements for unauthorized actions in the system are formulated and methods of their detection and processing are described. The solution of the problems connected with the organization of intrusion detection by diagnosing using the tools of the theory of Dempster-Shafer, providing the optimal distribution of system resources and guarantee the detection of unknown invasion system that were not included in the set of preassigned unauthorized actions is considered.

The research of the effectiveness of the method of detection of intrusions in a computer network on the basis of AIS and diagnostics was carried out. On the basis of the analysis of the information obtained, the following conclusion was made: with a correct training sample and a correct choice of learning parameters, the AIS method has the same high reliability as the diagnostic method. AIS requires additional time to create a training sample, but this allows the system to respond more quickly to new types of intrusions and reduce the number of false positives. The diagnostic method is less burdensome for the user system, but more often it identifies suspicious activity as intrusion. The results of the comparative analysis of intrusions show that the proposed

methods outperform the known antiviral products used in the comparative test and are capable of detecting unknown intrusions.

A comparative analysis of the proposed solutions has been carried out, which has experimentally confirmed that these methods increase the authenticity of the identification of unauthorized actions and attacks in the computer network.

The methods and models developed in the dissertation can be used to increase the effectiveness of detecting unauthorized actions in a computer network and bring them to the level of software tools. Experimental studies confirm the main provisions of the protection. Developed methods for improving the effectiveness of intrusion detection can be used to provide the operation of network management tools, as well as appropriate operating support systems. The novelty of the proposed solutions is protected by patents № 110330 and № 123634.

The results of the dissertation work are introduced into the educational process at the department of computer networks and systems of the National Aviation University and are used in the training courses: "Telecommunication technologies of computer networks" and "Architecture of computers" from 2017-2018, which is confirmed by the relevant act on implementation . The results of the work are used to increase the effectiveness of diagnosing unauthorized actions in the computer network of "Gazduservis" (implementation act dated 26.06.2017) and "Korop-plast" (Act of introduction from June 26, 2017).

Keywords: computer network, traffic, unauthorized actions, intrusion detection, diagnosis, artificial immune system, identification.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

1. Балакин С. В. Выявление компьютерных атак с помощью мониторинга сетевых объектов. *Технологический аудит и резервы производства*. Харьков, 2015. № 5-6(25). С.36-38. (Входить до міжнародних наукометричних баз Index Copernicus, РИНЦ, EBSCO Publishing, DOAJ).

2. Zhukov I. A., Balakin S.V. Detection of computer attacks using outliner. *Науковий журнал «Молодий вчений»*. К.: 2016. № 9(36). С. 91-93. (Входить до

міжнародних наукометричних баз РИНЦ, ScholarGoogle, OAJI, CiteFactor, Research Bible, Index Copernicus).

3. Балакин С. В. Организация пресечения вторжений в компьютерные сети алгоритмами выявления изменений. *Вісник НТУ «ХПІ». Серія: Механіко-технологічні системи та комплекси.* Харків, 2017. № 20(1242). С.3-7. (Входить до міжнародної наукометричної бази OAJI).

4. Жуков И. А., Балакин С. В. Обнаружение компьютерних атак с помощью метода отклонений. *Радіоелектронні і комп'ютерні системи: наук. – техн. жур.* Харків, 2016. №5(79). С. 33-37. (Реферується наукометричними базами ВАК, Index Copernicus, INSPEC IDEAS).

5. Жуков И. А., Балакин С. В. Идентификация атак в компьютерной сети методом усредненного времени обращений. *Проблеми інформатизації та управління: зб. наук. праць.* К.: НАУ, 2015. № 2(50). С.65–69. (Реферується наукометричною базою Україніка наукова, входить до міжнародних наукометричних баз РИНЦ, ScholarGoogle).

6. Балакін С. В. Застосування штучних імунних систем при виявленні шкідливих программ в комп'ютерній мережі. *Проблеми інформатизації та управління: зб. наук. праць.* К.: НАУ, 2017. № 1-2(57-58). С. 61-68. (Реферується наукометричною базою Україніка наукова, входить до міжнародних наукометричних баз РИНЦ, ScholarGoogle).

7. Жуков И. А., Балакин С. В. Исследование эффективности метода обнаружения вторжений в компьютерные сети на основе искусственных иммуных систем. *Проблеми інформатизації та управління: зб. наук. праць.* К.: НАУ, 2017. № 3(59). С. 65-69. (Реферується наукометричною базою Україніка наукова, входить до міжнародних наукометричних баз РИНЦ, ScholarGoogle).

8. Спосіб запобігання комп'ютерним атакам у мережі за допомогою фільтрації вхідних пакетів: пат. 110330 Україна: МПК G06F 12/14. №201602196; заявл. 09.03.16; опубл. 10.10.16, Бюл. №19. 4 с.

9. Спосіб діагностування несанкціонованих дій в комп'ютерній мережі: пат. 123634 Україна: МПК G06F 12/14. №201702719; заявл. 23.03.17; опубл. 12.03.18, Бюл. №5. 4 с.

10. Балакин С. В. Методы и средства повишения достоверности идентификации несанкционированных воздействий и атак в компьютерной сети. *Комп'ютерні системи та мережні технології :* зб. тез доп. VIII міжнар. наук.-техн. конф., м. Київ, 16-18 квіт. 2015 р. Київ, 2015. С. 11-12.

11. Балакин С. В. Системы предотвращения атак в компьютерной сети на основе сигнатурных методов. *Комп'ютерні системи та мережні технології :* зб. тез доп. IX міжнар. наук.-техн. конф., м. Київ, 21-23 квіт. 2016 р. Київ, 2016. С. 12-13.

12. Балакин С. В. Средства диагностирования несанкционированных воздействий и атак в компьютерной сети. *Комп'ютерні системи та мережні технології :* зб. тез доп. X міжнар. наук.-техн. конф., м. Київ, 20-22 квіт. 2016 р. Київ, 2017. С. 13-14.

13. Balakin S. V. Traffic analysis for intrusion detection systems in telecommunication networks. *Політ. Сучасні проблеми науки :* зб. тез доп. XIV міжнар. наук.-практ. конф., м. Київ, 2-3 квіт. 2014. С. 59-60.

14. Балакин С. В. Оптимизация искусственных иммунных систем при идентификации несанкционированных сетевых воздействий. *Комп'ютерні системи та мережні технології :* зб. тез доп. XI міжнар. наук.-техн. конф., м. Київ, 19-21 квіт. 2018 р. Київ, 2018. С. 7-8.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

НД – несанкціонована дія;

ІА – імунний алгоритм;

НМ – нейронна мережа;

ЕА – евристичний аналіз;

ДК – дендритні клітини;

ШІМ – штучні імунні мережі;

МАС – мультиагентні системи;

ШНС – штучні нейронні системи;

CSM – алгоритм виявлення змін «загальних сум»;

ТДШ – теорія Демпстера-Шафера;

А – антитіло;

G – антиген;

F – подібність антитіл/антигенів;

SRT – структура проникливості;

ОП – основа переконання;

DT – дерево діагностики;

ЦП – центральний процесор.

ВСТУП

Актуальність теми. Технології глобальних комп'ютерних мереж, що стрімко розвиваються, формують в інформаційній області нову систему відносин, яка відображає реалії технічного рівня {сучасного людства. Інтенсивність змін в значній мірі диктується тим величезним значенням, якого набуває інформація в постіндустріальному суспільстві, де вона стає головним ресурсом та інструментом одночасно.

В такому інформаційному просторі стрімко зростає кількість шкідливих програм і атак на комп'ютерні мережі. З переважною їх більшістю можуть впоратися антивіруси та фаєрволи, але деякі атаки можуть обійти такий захист, приносячи шкоду користувачеві чи компанії. Частіше за все наявний захист спрацьовує з запізненням, коли система вже була атакована й відбулась втрата даних чи контролю над певними компонентами мережі. В силу технічного прогресу зростає і складність способів проникнути в систему користувача. Іноді навіть власник комп'ютера може роками і не підозрювати, що всі вироблені їм дії на робочому місці можу з легкістю відслідковуватися зловмисниками. Починаючи з 2000-х років, коли почалася стрімка комп'ютеризація більшості установ, збільшилась і кількість скарг на різного роду атаки робочих комп'ютерів і мереж.

На даному етапі користувачеві мало захисту яку надає більшість антивірусних компаній, оскільки часто вона не своєчасна (спочатку йде розповсюдження вірусу і тільки тоді антивіруси займаються його «лікуванням»), чого достатньо зловмиснику для отримання доступу потрібної інформації чи пошкодження існуючої. Саме своєчасне оповіщення системи та користувача допомогло б підвищити ефективність виявлення несанкціонованих дій як у локальній, так і в мережі інтернет. Дана праця описує один з варіантів як цього можна досягти і впровадити у вже існуючі системи.

Аналіз літературних джерел показав, що відомі методи протидії вторгненням в комп’ютерні мереж є доволі обмеженими і малоефективними, а також потребують постійної модернізації та оновлення для підтримки роботи з новими

несанкціонованими діями. Це свідчить про те, що задача підвищення ефективності виявлення вторгнень в комп'ютерних мережах є актуальною.

Способи та методи виявлення вторгнень в комп'ютерних мережах описуються науковцями, такими як: Я. Янг, І.В. Котенко, М. Якобссон, С. Ветзель, В.І. Городецький, Н. Репп, Р. Хекманн, Г. Вігна, К. Вішал, Р.А. Кеммерер, К. Крюгель, Р. Вегнер, А.П. Демпстер, А.А. Романюхи, Л.Н. Кастро та ін.

Незважаючи на прогрес і числені роботи присвячені виявленню вторгнень у комп'ютерні мережі, НД мають числені особливості, котрі значно обмежують ефективність сучасних інструментальних засобів. Тому питання розробки методів і моделей виявлення НД, що дадуть змогу підвищити ефективність виявлення НД в комп'ютерних мережах має сьогодні важливе теоретичне та практичне значення. Завдання, які при цьому виникають, зумовили напрямок досліджень дисертаційної роботи. Актуальною є проблема розробки та дослідження методів виявлення вторгнень в комп'ютерних мережах, а задачі і проблеми при виконанні описані в даній роботі.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконувалась в рамках науково-дослідницьких робіт кафедри комп'ютерних систем та мереж факультету комп'ютерних систем: НДР № 682-ДБ13 за темою «Розроблення теорії, методів та технологій оптимального управління гарантоздатною комп'ютерною мережею» (номер державної реєстрації 0113U000028), у процесі виконання якої автор дисертації брав участь у розробці методики проведення експериментальних досліджень та обробці отриманих даних за 2014 рік; кафедральна НДР № 17/09.01.04 за темою «Системна інтеграція науково-навчального забезпечення другого рівня підготовки фахівців спеціальності 123 – комп'ютерна інженерія», у процесі виконання якої автор дисертації брав участь у реалізації методики підготовки фахівців у 2017 р.

Мета і завдання дослідження. Метою дослідження є забезпечення виявлення несанкціонованих дій в комп'ютерних мережах за рахунок розширення можливостей по їх обробці.

Завданням дослідження є обґрунтування актуальності використання зазначених методів і можливості їх впровадження у виробництво.

Основні задачі дослідження відповідно до поставленої мети полягають у наступному:

• проаналізувати існуючі методи і засоби розпізнавання несанкціонованих дій в комп'ютерних мережах;

• проаналізувати інструментальну базу для діагностування вторгнень у комп'ютерні мережі;

• розробити моделі виявлення несанкціонованих дій в комп'ютерних мережах на основі діагностування симптомів вторгнень;

• розробити метод діагностування вторгнень в комп'ютерні мережі за допомогою операторів теорії Демпстера-Шефера (ТДШ);

• розробити моделі аналізатора вторгнень в комп'ютерні мережі на основі даних поведінкового аналізу трафіку користувача;

• розробити метод виявлення несанкціонованих дій в комп'ютерних мережах на основі роботи елементів штучної імунної мережі;

• порівняти запропоновані методи;

• впроваджувати результати роботи на підприємствах та в навчальному процесі.

Об'єктом дослідження є розпізнавання несанкціонованих дій в комп'ютерних мережах.

Предметом дослідження є методи, системи та моделі ефективного виявлення несанкціонованих дій в комп'ютерних мережах.

Методи досліджень базуються на використанні теорії штучних імунних систем та операторів теорії Демпстера-Шафера, що дозволили синтезувати нові методи та моделі розпізнавання НД в комп'ютерній мережі з можливістю їх реалізації, засоби евристичного аналізу забезпечили можливість навчання запропонованих моделей і налаштування їх на виявлення НД та статистичні засоби математичного оброблення результатів комп'ютерних експериментів для відбору та порівняння ефективності запропонованих методів.

Наукова новизна одержаних результатів заключається в тому, що:

- *удосконалено* модель виявлення НД в комп'ютерній мережі, в якій розпізнавання відбувається за допомогою аналізу поведінкових ознак, що дає можливість автономно розпізнавати невідомі вторгнення і мінімізувати помилкові спрацьовування;

- *отримав подальший розвиток* метод виявлення вторгнень у комп'ютерні мережі, який базується на використанні операторів штучних імунних мереж для побудови структурованої мережі антитіл, що зі свого боку позитивно впливає на швидкодію і достовірність автономного виявлення як відомих, так і нових вторгнень;

- *запропоновано* модель виявлення вторгнень в комп'ютерній мережі засобами інспектування стану системи на виникнення аномальної поведінки, використовуючи дерево діагностування, що дозволяє відстежувати активність системи й симптомізувати дії користувача, а шляхом введення нових елементів і діапазонів роботи досягається підвищення достовірності виявлення атак;

- *вперше розроблено* метод розпізнавання НД засобами діагностування на основі операторів теорії Демпстера-Шафера, де на відміну від існуючих пропонується відстежувати часові фрагменти на заданих діапазонах часу і з них, за допомогою операторів злиття, формувати діагнози, за рахунок чого досягатиметься можливість автономного виявлення невідомих системі НД.

Практичне значення одержаних результатів. Розроблені в дисертаційній роботі методи і моделі можуть бути використані для підвищення виявлення НД в комп'ютерній мережі та доведені до рівня програмних засобів. Експериментальні дослідження підтверджують основні положення, що виносяться на захист. Розроблені методи підвищення ефективності виявлення НД можуть бути застосовані для забезпечення роботи засобів керування мережею, а також відповідних систем операційної підтримки. Новизну запропонованих рішень захищено патентами на корисну модель України № 110330 та № 123634.

Результати дисертаційної роботи впроваджено в навчальний процес на

кафедрі комп'ютерних мереж та систем Національного авіаційного університету та використовуються в навчальних курсах: «Телекомунікаційні технології комп'ютерних мереж» і «Архітектура комп'ютерів» з 2017 - 2018 рр., що підтверджено відповідним актом про впровадження. Результати роботи використані для підвищення ефективності діагностування несанкціонованих дій в комп'ютерній мережі ТОВ «Газбудсервіс» (акт впровадження від 26.06.2017 р.), а також для аналізу несанкціонованих дій в комп'ютерній мережі ДП «Коропласт» (акт впровадження від 26.06.2017 р.).

Особистий внесок здобувача. Основний зміст дисертаційної роботи та її результати повністю відображені в опублікованих наукових роботах автора та отримані здобувачем самостійно. Всі теоретичні та практичні результати, які складають основний зміст дисертаційної роботи опубліковано в 14 наукових працях. Усі основні положення та результати дисертаційної роботи отримані автором самостійно. У роботах, виконаних у співавторстві, автору належать такі результати: у праці [2] – розроблено й обґрунтовано використання методу виявлення відхилень для ідентифікації вторгнень; [5] – виконано аналіз особливостей і вимог, які повинна задовольняти комп'ютерна мережа для можливості ідентифікації вторгнень методом усередненого часу звернення, обґрунтовано його ефективність і коректність; [6] – аналіз ефективності виявлення вторгнень на основі імунних систем; [4] – ідея використання відхилень в отриманих значеннях від усереднених для виявлення можливих вторгнень та розробка теоретичних засад для їх ідентифікації; [8] – запропоновано використання фільтрації вхідних пакетів за допомогою діапазонів активності; [9] – запропоновано механізм діагностування несанкціонованих дій в комп'ютерній мережі.

Дисертаційна робота виконана на кафедрі комп'ютерних систем та мереж НАУ. Науковий керівник д.т.н., професор Жуков І.А.

Апробація результатів дисертації. Основні положення дисертаційної роботи доповідалися на XIV-й міжнародній науково-практичній конференції

«Політ.Сучасні проблеми науки» (Київ, 2-3 квітня 2014 р.), на VIII міжнародній науково-технічній конференції «Комп'ютерні системи та мережні технології» (м. Київ, 16 – 18 квітня 2015 р.), міжнародній науково-технічній конференції «Cyber forum DESSERT 2016 B2S – S2B » (Чернівці, 18 – 23 травня 2016 р.), на IX міжнародній науково-технічній конференції «Комп'ютерні системи та мережні технології» (м. Київ, 21 – 23 квітня 2016 р.), на X-й міжнародній науково-технічній конференції «Комп'ютерні системи та мережні технології» (м. Київ, 20 – 22 квітня 2017 р.).

Публікації. За результатами досліджень опубліковано 14 наукових праць, у тому числі 7 статей у наукових фахових виданнях України, які включені до міжнародних наукометричних баз, 2 патенти України на корисну модель, 3 статті у наукових збірниках, що додатково відображають наукові результати дисертації та 5 тез доповідей в збірниках матеріалів конференцій.

Структура та обсяг дисертації. Дисертація складається із вступу, чотирьох розділів, загальних висновків, додатків, списку використаних джерел і має 131 сторінку основного тексту, 22 рисунки, 7 таблиць, 6 сторінок додатків. Список використаних джерел містить 146 найменування і займає 14 сторінок. Загальний обсяг роботи 151 сторінка.

РОЗДІЛ 1
АНАЛІЗ СТАНУ ПИТАННЯ І ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Науково-технічні проблеми виявлення несанкціонованих дій в комп'ютерних мережах

Завдяки інтенсивному використанню інтернету безпека мереж стає ключовим фундаментом для всіх веб-додатків. Виявлення вторгнень за допомогою аналізу записів в мережевих процесах є важливим способом вирішення проблем в області мережевої безпеки.

Вторгнення може поставити під загрозу не лише цілісність даних, а і саму систему. З розвитком інформаційних технологій і збільшенням швидкості передачі даних, виникають загрози некоректного використання інтернету. Необхідні більш надійні системи контролю, які вирішують проблему захисту мереж без втручання людини.

Однією з перших робіт в цій галузі була [1], вона і визначила основні поняття та вирішення проблем. Перші роботи були скоріше концептуальними – в них намагалися не збудувати інструментальні фільтри або методи, а спробувати застосувати теорію ймовірності для вирішення даних проблем.

У праці Шейнера [2] увага зосереджена на можливості автономного виявлення уразливості програм і протоколів аналізом критеріїв поведінки самої системи.

Багато моделей побудовано на неформальних способах [3] таких, як сигнатурні, в яких важко отримати коректну оцінку ефективності та завершеності [4]. Модель, описана в роботі [5], дає можливості для вирахування ефективності і надійності даних рішень. В її основу взято напрацювання робіт [5, 6].

Праці Вігни і Кемерера [7] стосуються мови STATL і моделей атак, що базуються на її операторах. Автори відштовхуються від того, що атака характеризується станами і переходами. Головний мінус запропонованої системи у відсутності інструментів для контролю потоків вторгнень і дій самої системи користувача.

Робота Котенко І.В. [8] описує атаку з точки зору зловмисника і базується на поняттях мети вторгнення.

Використання інструментів виявлення аномалій і атак ускладнено сферами призначення. Чим вужче сфера використання, тим простіше застосовувати до неї ті чи інші інструменти досліджень, оскільки легше підібрати відповідну модель поведінки мережевих об’єктів [9, 10, 105].

Перевірено можливість використання нейронних мереж [12, 13]. Відмінною особливістю НМ є те, що вони починають працювати лише після процесу навчання. Це одна з головних переваг НМ перед традиційними алгоритмами. Навчання базується на зв’язках між нейронами, які визначають співвідношення вхідних і вихідних сигналів нейрона [14]. НМ базується на "навченості" та не дозволяє аналітично прораховувати похибки. До недоліків можна віднести те, що топологія мережі та розташування вузлів визначаються тільки після досить великої кількості проб і помилок. Головні недоліки НМ:

- неефективні при вторгненнях U2R і R2L [11, 15, 16];

- низька надійність.

Для вирішення зазначених проблем запропоновано новий підхід до методу відхилень на основі виявлення вторгнень. Виявлення відхилень проводиться з метою підвищення стабільності виявлення вторгнень. Такий підхід складається з двох етапів: навчання з робочими наборами даних і тестування з наборами даних зі зразками вторгнень. Такі дані використовується для підготовки виявлення вторгнень на початковому етапі реалізації. Нормальні набори даних підвищують продуктивність протидії вторгненням. Якщо кількість помилок перевищує порогову величину, то тестований набір даних буде характиризуватися системою як несанкціонована дія.

Різні способи можуть бути використані для виявлення вторгнення, але кожен з них є специфічним для конкретного методу [9]. Основна мета системи виявлення вторгнень – ефективно виявляти атаки. Важливо виявити атаку на початковій стадії, щоб зменшити її негативні наслідки. У даній роботі

запропонований підхід відхиляючихся значень, при якому аномалія вимірюється факторами відхилень.

Модель навчання складається з масивів даних з розподіленим середовищем зберігання. Це є основною складовою підвищення продуктивності системи виявлення вторгнень. Отриманні експериментальні результати показали, що запропонований підхід виявляє аномалії ефективніше за відомі методи.

Розробці методу виявлення атак на основі інформації про поведінку відхиляючихся значень в мережі і присвячена робота.

В останні роки зросла не лише складність програмних продуктів, а й загроза зі сторони вірусного програмного забезпечення. Саме такі програмні елементи користуються великою популярністю на чорних ринках і на даний час дуже стрімко еволюціонують. Вони надають змогу корпораціям, а іноді й країнам, завдавати великої шкоди своїм економічним та політичним опонентам. Ринок вірусного програмного забезпечення постійно зростає і розвивається. Від невеликих програмних блоків, вбудованих у виконувані файли інших програм, до складних самостійних багаторівневих систем, що складаються з великої кількості компонентів, котрі мають різні цілі і задачі: інсталятори, завантажувачі, програми-маскувальники тощо. Основним середовищем розповсюдження такого програмного забезпечення у сучасному світі є інтернет.

Способів виявлення вторгнень дуже багато, але більшість з них або неможливо застосувати на практиці, або настільки громіздкі, що істотно знижують продуктивність системи користувача або самої мережі. Тому питання актуальності даних розробок криється в конкретизації та модернізації існуючих методів, які теоретично виконують поставлені цілі, але на практиці їх важко реалізувати.

Використання інструментів виявлення аномалій і атак ускладнене сферами призначення. Чим більше конкретизована спеціальність – тим простіше застосовувати до неї ті чи інші інструменти. Найчастіше від самої конкретики і виходять певні методи, які найкращим способом можуть «покрити» всі слабкі місця в забезпеченні працездатності системи.

Розглядалася також можливість роботи з нейронними мережами, але величезним мінусом при роботі з ними стали труднощі верифікації результатів дослідження навчальних вибірок.

Для роботи з мережевими елементами підходить адаптивний метод. Він характерний тим, що навіть з низькою обчислювальною складністю буде мати низький рівень помилкових повідомлень. Оскільки цей метод має високу продуктивність і мінімальні обчислювальні потужності, то він підійде не тільки для роботи з базами даних (для яких він спочатку був розроблений), а й для опису поведінки мережевих елементів. Такий метод забезпечить високу надійність при виявленні атак або незапротокольованих дій. Адаптивність в свою чергу допоможе залучити рішення проблеми в одній системі для ряду інших. Саме проблема адаптивності є критичною для більшості готових рішень. Комбінуючи цей метод можна зберегти і вдосконалити саму формулу атак за допомогою мережевих елементів. Актуальність даних робіт полягає у тому, що ми отримаємо продукт, який можна використовувати для будь-яких систем.

З огляду на збільшення загроз отримання несанкціонованого доступу до даних користувача чи компаній, підприємства мають постійно витрачати величезні кошти на захист своїх даних. Малі підприємства знаходяться під загрозою так само, як і великі компанії та холдинги.

Безпека для малого бізнесу може залежати від багатьох аспектів, таких як [17, 18]:

- визначення політики і процедур безпеки;
- IT-інвестиційні рішення;
- питання кадрової безпеки;
- проблема безпеки даних;
- мережева безпека;
- захист від вірусів;
- політика виявлення вторгнень;
- використання карт доступу;
- резервні процедури і плани аварійного відновлення.

Часто несанкціонованим доступом називають слабкість в контрольованій системі, де елементи управління перестають бути ефективними.

Також несанкціонований доступ до системи може бути відкритий через помилку в програмному забезпеченні, яке може бути використане зловмисниками для отримання доступу до системи або мережі.

Виявлення вразливостей і несанкціонованих дій мають важливе значення для підвищення рівня безпеки мережі.

Операційна система піддається несанкціонованому доступу лише тоді, коли є усі передумови для запуску додатку, що не її частиною. Такими додатками можуть бути як різного роду редактори (фото, відео), так і числені програмні комплекси для виконання різних задач (програвачі, інженерні додатки, навігаційні центри тощо). Піддатися несанкціонованому доступу можуть як сама система, так і всі додатки, котрі сумісні з даною системою. Причиною виникнення таких протиправних дій і зловживань є виконання таких умов [19, 20]:

1) широке застосування даної системи;
2) відкритий доступ ядра (opensourse системи);
3) недостатній захист.

Усі ці умови необхідні для отримання несанкціонованого доступу до мережі, програми чи операційної системи. Умова популярності необхідна для того, щоб ціль атаки мала сенс і її вплив здобув видимі наслідки. Немає сенсу зламувати чи отримувати доступ до мереж, про існування яких знає декілька осіб. Якщо система має лише пару копій і використовується вузьким колом спеціалістів, то знижується її привабливість для злому. Іншою умовою є широке поширення системи, що впливає на її привабливість для хакерів.

Безпека системи основана на архітектурних чи програмних рішеннях, які перешкоджають невідомим функціям отримувати доступу до файлів користувача та життєво важливих частин системи. Захищеність блокує несанкціоновану активність, але при цьому накладає певні обмеження на можливості деяких програмних елементів користувача [21].

Більшість присутніх на ринку методів і програмних рішень не в змозі гарантувати стабільно високий рівень захисту системи, локальних і глобальних мереж. Проблема криється в стрімкому рості нових НД. Головна небезпека такого росту в тому, що при інтенсивних потоках вторгнень практично неможливо на 100% забезпечити їх виявлення. Причинами такого росту НД є [18-20]:

Переважно більшу частину НД створюють для ураження комп'ютерів в глобальній мережі, а об'єми написання вірусів зростають кожного дня. При таких темпах розвитку інструментів отримання доступу до компонентів мережі неможливо писати оновлення для баз сигнатур антивірусним компаніям вчасно і з урахуванням кількості та різноманітності НД.

Стрімке зростання НД загрожує тим, що переважна більшість комп'ютерів буде уражена ще до виходу нових вірусних сигнатур. Антивірусним компаніям потрібно постійно випускати оновлення сигнатур щоб конкурувати між собою, а це скорочує час на аналіз шкідливого коду і негативно впливає на якості кінцевого продукту. Працівникам просто не вистачає часу для якісного аналізу кодів вторгнень.

Знешкодження шкідливого коду також непроста задача. Так як написання вірусів стоїть на комерційній основу, то при цьому розробляються і застосовуються технології, що використовують методи приховування від антивірусів, засновані на знайдених вразливостях [11]. Ці технології ускладнюють задачу розпізнавання і видалення вторгнень. НД можуть інкапсулювати в себе процедури самозахисту для перешкоджання видаленню, деактивуючи системні утиліти доступу до реєстру або контролю над процесами. Також застосовується код, який відстежує цілісність файлів НД і необхідні для роботи ключі в системних реєстрах.

Нагально постають проблеми ефективного споживання системних ресурсів. Для відстеження трафіку в режимі реального часу антивіруси повинні мати модулі роботи з системними подіями, що дасть змогу фільтрувати потоки й унеможливлювати НД, які можуть скомпрометувати захист. Більшість системних подій і частота їх появи може сильно уповільнювати роботу при опрацюванні НД.

Існують проблеми несумісності антивірусів. Часто через конфлікти підпрограм перехоплення системних подій неможливо працювати з різними антивірусами одночасно.

Виникає потреба в нових методах боротьби з НД, котрі будуть базуватись на аналізі поведінки та зможуть обходити шифрування. Такі підходи повинні ефективно боротися зі старими й новими модифікаціями вірусів, зберігаючи при цьому високу працездатність і мінімально навантажувати систему. Також важливо навчити систему автономно виявляти несанкціоновані дії без звернення до баз даних.

1.2 Аналіз існуючих інструментальних засобів виявлення несанкціонованих дій і атак в комп'ютерних мережах

Сучасне шкідливе програмного забезпечення містить широкий спектр вірусів, що завдають шкоди не лише інфікованій системі, а іноді й всій локальній чи глобальній мережі. Віруси діляться на класи з загальними характеристиками [18-20, 22, 23]:

1) середовище існування;
2) алгоритми роботи;
3) руйнівна сила.

Середовище існування ділиться на:

1) файлові;
2) завантажувальні;
3) макровіруси;
4) мережеві.

Файлові віруси вмонтовуються в файли. Завантажувальні ховаються в boot-секторах на жорсткому диску. Макровіруси інфікують документи і електронні таблиці текстових редакторів. Мережеві віруси поширюються через пошту чи мережі.

Віруси можуть комбінуватись для ускладнення їх виявлення. Комбінування маскує наявності вірусів у системі поки зловмисники руйнують її чи крадуть дані

користувача. Прикладами таких комбінацій є файлово-завантажувальні та мережеві макровіруси. Вони мають складний алгоритм роботи і використовують стелс і поліморфік-технології для потрапляння до системи [23].

Алгоритми вірусів характеризують [18-20, 22]:

1) резидентність;
2) використання стелс-алгоритмів;
3) самошифрування і поліморфічность;
4) використання нестандартних заходів.

Резидентний вірус інкапсулює свою частину в оперативну пам’ять, яка потім перехоплює звернення операційної системи до об’єктів зараження і записується в них. Резидентні віруси знаходяться в пам’яті та є активними аж до вимикання комп’ютера або перезавантаження ОС. Нерезидентні віруси не інфікують пам’ять комп’ютера і зберігають активність обмежений час.

Стелс-алгоритми [24] приховують віруси в системі. Найпоширеніший стелс алгоритмом – перехоплення запитів ОС на читання/запис заражених об’єктів. Стелс-віруси при цьому «підставляють» замість себе неуражені ділянки інформації. У випадку макровірусів – це є заборона викликів меню перегляду макросів.

Самошифрування і поліморфічность [22, 23] використовуються всіма типами вірусів для того, щоб максимально ускладнити процедуру виявлення вірусу. Поліморфічні віруси [25] не мають сигнатур і не містять жодної постійної ділянки коду. У більшості випадків два зразки того самого поліморфічного вірусу не будуть мати збігів. Це досягається за допомогою шифрування основного тіла вірусу і модифікаціями програми-розшифровувача.

За деструктивними можливостями НД поділяються на [26]:

1) нешкідливі віруси - ніяким чином не впливають на роботу комп’ютера, крім зменшення вільної пам’яті на диску в результаті свого поширення;

2) безпечні віруси - вплив вірусів обмежується зменшенням вільної пам’яті на диску і графічними, звуковими та іншими ефектами;

3) небезпечні віруси - можуть призводити до серйозних збійних ситуацій у роботі комп’ютера;

4) дуже небезпечні віруси - можуть призводити до втрати програми, знищення даних, стирання необхідної для роботи комп’ютера інформації, яка записана в системних областях пам’яті, і навіть сприяти прискореному зносу рухомих частин механізмів, наприклад, головок вінчестера.

Проведений аналіз НД показує, що необхідно розробити методи та моделі розпізнавання як старих, так і нових модифікацій несанкціонованих дій, які дозволять визначати їх тип.

1.2.1 Сигнатурний аналіз

Сигнатури дозволяють виявляти вже відомі віруси [27, 28]. Сигнатура – це набір ознак за допомогою яких можна охарактеризувати об’єкт. Ознаки дають змогу коротко описувати величезні об’єкти. На такому принципі функціонують і хеш-функції, котрі характеризують об’єкт за допомогою коротких ознак. Ознаки типу файлу, дати, адреси та розміру називають «слабкими сигнатурами».

Сигнатура є унікальною ознакою вторгнення, за допомогою якої можна віднести фрагмент що її містить до НД. Якщо вторгнення не є унікальним, то сигнатури що його описують будуть однотипними і являтимуть послідовність розташованих один за одним байтів і адреси в файлі цієї послідовності. Якщо відомо розмір файлу – то це буде додатковим тригером достовірності виявлення НД. Чим більше інформації ми маємо про атаку, тим точніше зможемо її охарактеризувати. Для різних типів вторгнень використовуються різні сигнатури. В найсильнішу сигнатуру входить незмінна частина вірусу (якщо він поліморфний), що значно збільшує розмір сигнатури.

Для мінімізації розміру і довжини сигнатур використовується фрагментація. Фрагментація дає можливість використовувати переривчасті сигнатури, котрі мають дві частини:

1) загальні (характеризують весь тип вторгнень);
2) унікальні (модифіковані вручну).

Також існує метод «половинчастих» сигнатур, що даю можливість використовувати частини різних сигнатур при виявленні поліморфних НД [20, 22, 23]. Метод оперує бітовими полями і може бути використаний не з усіма сигнатурами (все залежатиме від їх типу). Складність розпізнавання поліморфних вторгнень в тому, що вони після розшифровки тіла в пам'ять зберігаються незмінно. Складність в тому, щоб визначити час розшифровки, котрий являється таймером початку роботи вторгнення. Половинчасті сигнатури можуть виявити навіть такі вторгнення за допомогою розшифровки виконуваного і виконаного фрагмента

Недолік методу – величина сигнатур, тому використовуються контрольні коди [28]. Вони формуються з коду вірусу і являються унікальними. Може бути кілька вторгнень з однаковими контрольними кодами (колізіями), але це не впливатиме на їх виявлення.

Контрольними кодами можна заміняти хеш-функції як SHA і MD5 [30], незважаючи на складність їх застосування, вони дуже компактні та вміщаються в одне слово (32 чи 64 байтне) і нівелюють можливість колізій. При використанні хеш-функцій в антивірусній базі маються наступні поля: зміщення, довжина та хеш файлу. При роботі з файлом антивірус перевіряє хеш фрагмента в базі даних з вказаним зміщенням і порівнює з еталонним значенням. Значення рівні – то фрагмент буде шуканим. Цей метод не поступається в точності сигнатурним методам і має високу продуктивність і мінімальні вимоги до системних ресурсів.

Основна перевага сигнатурних методів – точне виявлення типу вірусу. Ця особливість дає змогу вносити в базу як сигнатури, так і методи блокування НД.

Недоліки сигнатурного методу:

1) потрібні семпли НД;
2) необхідність оновлення;
3) ручний аналіз НД при колізіях;
4) виявляє лише відомі НД.

Основним недоліком методу можна назвати мінімальну автономність і залежність від оновлення.Такий метод являється найкращим з точки зору

монетизації – користувач завжди повинен оплачувати можливість оновлення своєї системи.

1.2.2 Евристичний аналіз

Головна мета евристичного аналізу - відстеження невідомих і нових модифікацій несанкціонованих дій [31-34]. ЕА приймає і досліджує програмні файли, а на основі результатів роботи робиться висновок про наявність вторгнень. Для отримання коректного результату потрібно виконати наступні кроки:

Семантичний аналіз. Він дозволяє розпізнати та перетворити до операбельного виду виконувані команди. Після цього проводиться аналіз даних команд, щоб знайти послідовності в коді програм, що реалізовують небезпечні дії;

Інтерпретація. Цей крок допомагає знаходити поліморфні програми (коли дія починає виконувати вторгнення не відразу, а по закінченні певного наперед заданого часу чи циклу). Цей крок вимагає запуску програми. Для виявлення використовується потік команд, що може мати негативні наслідки для інформації користувача, тому виконання даного коду на комп’ютері не бажаний. Для цього емулятором моделюються апаратні і програмні функції, котрі фіксують активність виконуваного коду;

Прагматичний аналіз. Дозволяє за змістом команд і їх груп визначати призначення алгоритму атак.

Семантичний аналіз. Програма впливає на багато факторів (значення регістрів, прапорів процесора, областей пам’яті). Більшість цих параметрів не буде враховуватись при виявленні вторгнень. При виявленні використовуються «дискретні» моделі вірусів [22, 25], і далеко не кожна програмна дія набуває статусу «події», котрі являються програмними діями пов’язаними з системними викликами, що приводять до змін в системі. При семантичному аналізі проводиться пошук і розпізнавання в лістингу дизасемблера послідовностей команд які реалізують «події», що належать «дискретним» моделям. Розпізнавання відбувається наступним чином: множина будь-яких елементів (бітів, байтів, слів або їх послідовностей), що можна представити в якості

«алфавіту», а самі ці елементи «літерами алфавіту». Комбінуючи елементи та складаючи з них різні послідовності ми отримуємо різні «фрази». Описана множина фраз і буде «мовою».

Для еталонних типів вторгнень створюються кінцеві автомати для розпізнавання послідовності асемблерних команд, що реалізують їх поведінку. Програма спочатку дизасемблюється, а потім розпізнається автоматами. На підставі кількості розпізнаних фрагментів і їх функціоналу – аналізатор виносить вердикт щодо шкідливості виконуваного файлу.

Семантичний аналіз [22, 25, 35, 36] поділяється на: статичний і динамічний. Статичний полягає в дизасемблюванні образу виконуваного файлу з накопичувача і його аналіз кінцевими автоматами. Такий метод неефективний через розповсюдження пакувальників та систем захисту програм від злому. Такі програми архівують/шифрують вміст виконуваних файлів, після чого код програми неможливо дизасемблювати. Проблема вирішується застосуванням бібліотек алгоритмів розпакування, за допомогою яких антивірус може розпаковувати упаковані файли. Ефективність методу залежить від своєчасного оновлення типу пакувальника і підтримки розпакування.

Динамічний підхід використовує семантичний аналіз з відладчиком/емулятором. В такому разі перевіряється не лістинг дизассемблера, а фрагменти коду під час покрокового виконання або інтерпретації, що мають попередню обробку. Команди надходять на вхід автомата послідовно, у міру їх інтерпретації емулятором, що дає перевагу при аналізі упакованих або самомодифікованих об’єктів, бо з’являється можливість досліджувати об’єкти після того як пакувальники/шифрувальники відновлять оригінальні тіла в пам’яті та передадуть їм управління. Такий метод працює без використання бібліотек пакувальників, але являється ресурсоємким.

Динамічний семантичний аналіз широко використовується практично у більшості антивірусах. Його перевагою є низький рівень можливих помилок, а недоліком є невисоку ефективність при роботі з нестандартними кодами, бо автомат частіше всього налаштований на розпізнавання заданої послідовності

певних символів. Досить знайти еквівалентну по функціоналу послідовність, на розпізнавання якої налаштований автомат, і метод втратить ефективність [22, 35]. Наприклад, заміна деяких команд еквівалентними блоками. Можна використовувати API функції замість InternetOpenURL, а замість InternetReadFile - бібліотеки wsock32.dll, socket, send, recv, для виконання аналогічних функцій. Це призведе до того, що автомат не зможе перейти в свій кінцевий стан і розпізнати НД. Щоб автомат продовжував розпізнавати фрагмент, необхідно його модифікувати ураховуючи всі можливі варіанти еквівалентного коду, а це практично неможливо.

Інтерпретація шкідливого коду емулятором. Емулятор створює штучне оточення для моделювання необхідного функціоналу для досліджень при високому рівні захисту системи користувача. Навіть якщо запущена шкідлива програма чи вірус, вона не зможе зашкодити системі чи мережі з емулятора.

Необхідність емуляції в тому, що програма не статичний об’єкт, і розпізнавати її статичними методами не завжди можна. Прикладами таких програм є поліморфні віруси і пакувальники [20, 24, 25, 36]. Статична сигнатура запрограмована на появу після виконання певної кількості команд.

Команди можливо виконувати як і відладчики - послідовно, роблячи зупинку після виконання команди, встановивши процесор в режим покрокового трасування. Таким чином використовується трасування коду в реальних умовах. Умовою буде необхідність спостереження за роботою відладчика людини-оператора, для керування процесом і його зупинкою. Також оператор зможе визначати будь-які звернення програм до зовнішнього середовища.

Антивіруси при спостереженні за діями працюючих програм також використовують відладчики, які повністю підконтрольні своїй внутрішній керуючій системі - модулю проактивного захисту [34, 36]. Цей модуль може розпізнавати та блокувати стандартні типи несанкціонованих дій, працюючи як своєрідний фільтр. Слід зазначити, що такий аналіз буде проходити в режимі реального виконання програм, а це дасть змогу отримувати актуальні дані про наявність НД. Аналізатор досліджує взаємодію програми з системою (перевіряє

аргументи викликів API функцій), даючи програмі доступ до реальних ресурсів комп’ютера, що гарантує коректність результатів. Недоліком такого методу є можливість проникнення НД в систему і збою роботи.

В цілях безпеки програми запускаються в емуляторі [36-39] для проактивного захисту від НД. При емуляції програма виконується на інтерпретаторі, який відтворює зовнішнє середовище: пристрої, пам’ять, системні виклики. Логіка розпізнання НД залишається аналогічною проактивним методам. Недоліками емуляції є:

1. Необхідність моделювання апаратних вузлів комп’ютера і частин ОС. Це складний процес і вимагає детального вивчення систем на котрих будуть працювати дані методи й програмні продукти. На ринку є багато програмних засобів котрі емулюють основні при роботі з вторгненнями модулі системи, але і розробники вірусів не стоять на місці і навчились обходити навіть емулятори. Важливим при емуляції є моделювання роботи в мережі інтернет. Тут необхідно моделювати функції завантаження і прийому файлів з подальшим їх збереженням на диску.

2. Низька швидкодія. Емулятор описує процеси працюючої системи, а це вимагає використання величезних обчислювальних ресурсів. Швидкість програмної моделі набагато нижча чим в апаратного аналога навіть враховуючи максимально можливу її оптимізацію. Деякі розробники створюють спеціальні системи емуляції використовуючи фізичні процесори, що дозволяє проводити дослідження з максимальними потужностями.

3. Обмеженість «глибини емуляції». Частіше за все використовується обмежений набір команд при емуляції. Емулюється лише частина програми, а емулятор припиняє роботу або після виконання кожної інструкції, або емулюючи роботу блоків, і лише після цього проводить тести на наявність НД. Після закінчення роботи програми її емуляція втрачає сенс, оскільки вона переходить цикл очікування нових подій. Такими діями можуть бути нові вхідні дані чи команди. При переході в режим очікування емулятор повинен завершувати

роботу і переходити до наступної програми. Визначити настання такого циклу можна двома підходами:

1. Заданням кроків емуляції. Наприклад, після виконання заданої кількості різних кроків чи операцій завершувати емуляцію. Завершувати емуляцію виконавши деяке число кроків або певну кількість різних команд;

2. Заданням часу на емуляцію. По закінченню відведеного на виконання часу програма завершує емуляцію. Даний спосіб зможе обходити обмеження лічильника команд. Такий спосіб не буде сповільнювати систему, бо порожні цикли навіть при повторі не навантажують систему. Час на виконання даних операцій встановлюється експериментально.

Недоліки евристичних методів:

1. ЕА при роботі продукує багато помилкових виявлень.
2. Складність. Попри складність коректного налаштування метод може сповільнювати систему.

Наведені недоліки, в силу специфіки евристичних методів, складно компенсувати навіть апаратними засобами. При подальшому їх використанні доцільно їх модифікувати та допрацювати на предмет виконання поставленої перед ними задачі виявлення вторгнень. Для ефективного використання методів ЕА необхідно:

1. Підвищити результативність розпізнавання нових модифікацій НД.
2. Знизити рівень помилкових спрацювань.
3. Підвищити швидкість роботи методу.
4. Мінімізувати ступінь використання системних ресурсів.
5. Можливість адаптивності.

Сучасні ЕА на основі поведінкового аналізу й емуляторів дають можливість ефективно виявляти несанкціоновані дії відомого типу і невідомі системі.

1.3 Порівняння методів і засобів підвищення надійності виявлення несанкціонованих дій і атак в комп’ютерних мережах

Методи штучного інтелекту [40-42] дають змогу реалізувати можливості навчання для діагностики НД. Ці методи дозволяють значно підвищити надійність

і захист комп'ютерних систем та мереж. Численні евристичні методи реалізовані базуючись на використанні НМ, штучних імунних мереж та мультиагентних систем [59, 60, 62-64, 68, 73-78, 82, 86], можуть виявляти вторгнення. Слід зазначити, що наведені методи мають певні недоліки, котрі або сповільнюють загальну продуктивність систем обробки даних, або жертвують точністю. Для вибору оптимальних інструментів розглянемо найпопулярніші методи виявлення НД.

1.3.1 Продукційні системи

Продукційні системи – це системи, які використовують продукційну модель представлення знань [43]. Продукційна модель представлення знань є однією з найпоширеніших. Представлення знань за допомогою правил-продукцій має в деяких відносинах подібність із правилами виводу логічних моделей [44, 45]. Це дозволяє за допомогою продукцій виконувати ефективний вивід і, крім того, завдяки природній аналогії процесу міркувань людини дані моделі наочніше відбивають знання. У системах продукций знання представляються за допомогою наборів правил виду: "якщо А, то В". Тут А і В можуть розумітися як "ситуація – дія", "причина – наслідок" і "умова – вивід" [46].

Однак не слід ототожнювати правило-продукцію і відношення логічного проходження. Справа в тім, що інтерпретація продукції залежить від того, що розташовано ліворуч і праворуч від знака логічного проходження. Часто під А розуміється деяка інформаційна структура (наприклад, фрейм), а під У – деяка дія, що полягає в її трансформації (перетворенні). Логічна інтерпретація розглянутого вираження накладає обмеження на А і В.

Узагальнено така модель має вид [46, 47]:

$$P = (K, U, A \rightarrow B, E),$$

де, K – клас даної ситуації;

U – умова активації;

$A \rightarrow B$ –Суть продукції;

E – умова закінчення.

Продукційна модель може бути спрощена порядком чи пріоритетом, що може бути введений всю множину продукцій [45]. Порядок означає, що кожна наступна продукція використовується лише тоді, коли попередня продукція не підходить. При пріоритетах спочатку використовується продукція з найвищим пріоритетом. Для боротьби з суперечностями при розширенні баз даних (наприклад однаковий пріоритет) можливе використання повернень [41]. Компоненти такої системи – це база знань, робоча пам’ять і механізм виведення [46]. Структура системи представлена на рис. 1.1.

База знань, використовуючи продукції, описує предметну область. Робоча пам’ять містить факти про поточний етап логічного висновку. Механізм виведення підбирає правила, які можуть виконати дані продукції.

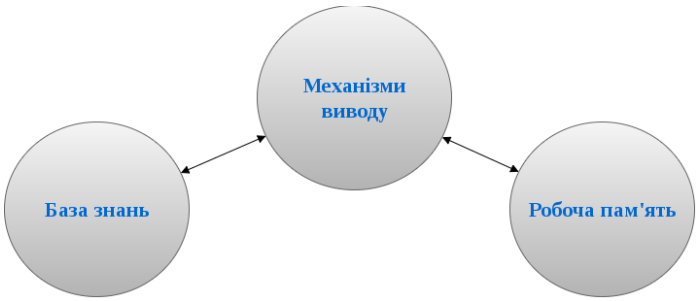


Рис. 1.1. Структура продукційної системи

Продукційні системи використовуються при сигнатурному методі аналізу [27]. Такий аналіз дієвий лише тоді, коли вторгнення відбуваються за однаковими алгоритмами роботи (сигнатурами). Якщо відомий сценарій роботи атаки, то він порівнюється з діями користувача і при виявленні схожих з вторгненнями дій вони блокуються чи видаляються. Якщо сигнатура не повністю відповідає дії користувача, а частково, то можливий варіант сповіщення оператора чи антивірусної системи про можливість НД. Найчастіше даний метод використовується системами виявлення мережевих атак [36, 48, 49] (Snort, RealSecure, eTrustID, AVZ, KasperskyLab, MSE). Сигнатурні методи широко використовуються в наш час, і потребують постійних оновлень для збереження ефективності й конкурентоздатності.

Продукційні моделі можуть підтверджувати вторгнення, опрацьовуючи файли аудиту, роботу виконуваних процесів і мережевих портів.

Переваги продукційних систем такі:

1) модульність – кожне правило описує невеликий, відносно незалежний фрагмент знань;

2) інкрементність – можливість додання нових правил незалежно від інших правил;

3) зручність модифікації як наслідок модульності та інкрементності;

4) прозорість системи (легкість простеження логіки та пояснювання виведення).

Недоліки продукційні моделі:

1) процес виводу має низьку ефективність, тому що при великому числі продукцій значна частина часу затрачується на невиробничу перевірку умов застосування правил;

2) перевірка несуперечності системи продукцій стає дуже складною через недетермінованості вибору виконуваної продукції з конфліктної множини.

Більшість недоліків можна виправити оптимізацією під конкретну виробничу систему.

1.3.2 Статистичний метод

Статистичний метод [50-52] при роботі веде обліку появи характерних ознак, по яким робиться висновок про наявність несанкціонованої активності. Даний метод продукує імовірнісні висновки про наявність вторгнень (коли поведінка системи перестала протікати заведеним чином, то статистичний метод зможе це виявити).

При підрахунку частоти звернення до команд процесора будується таблиця їх активності, на основі якої приймається рішення про наявність чи відсутність вторгнення. Цей метод чудово виявляє поліморфні віруси, які використовують мінімальний набір команд в дескрипторі.

Популярним є метод, що базується на операторах теорії ймовірності і описаний наступним рівнянням Баєса [52-54]:

$$P(D_i|S_j)= P(D_i) * P(S_i|D_j)| \ (P(D_1) * P(S_j|D_1) + P(D_2) * P(S_j|D_2) + (...),$$

де S_j – події;

D_i – діагноз;

$P(D_i|S_j)$ – ймовірність коректності і-го діагнозу виявлення j-ї події;

$P(D_i)$ – ймовірність і-го діагнозу;

$P(S_j|D_i)$ – умовна ймовірність появи і-ї ознаки події j.

Будується вибірка вірусів і програм, де кінцеві дані будуть позначені $P(D_i)$. Потім беруться віруси і визначаються рейтинги $P(S_K\ D_i)$. На результатах будуються дані про наявність НД. Відбувається сканування вторгнень в системі для формування вибірки подій K, $S = \{ S_1, S_2, ..., S_K\}$. Потім знаходиться ймовірність того, що дана активність буде діагностована D_i. Для діагностування вираховуємо ймовірності $P(D_i|S_j)$ для S_j з множини S. Далі знаходиться ймовірність:

$$\mathrm{P}(D_i\ |\mathrm{S}\ ) = \mathrm{P}(D_i \mid S_1) * \mathrm{P}(D_i \mid S_1) * \mathrm{P}(D_i|\ S_2) *\cdots* \mathrm{P}(D_i \mid S_K).$$

Потім підраховуються ймовірності вторгнень і вибирається найбільша.

Даний підхід не ефективний при роботі з почтовими клієнтами, оскільки не всі ознаки можливо описати даним методом.

1.3.3 Штучні нейронні системи

Штучні нейронні системи - це моделі нейронної структури мозку, який, головним чином, навчається з досвіду. Природній аналог доводить, що множина проблем, які поки що не підвладні розв'язуванню машинами, можуть бути успішно вирішені блоками нейромереж [54-58]. ШНС являється мережею процесорів (нейронів). Вони оброблюють і передають інформацію наступним нейронам. ШНС покроково вирішують все складніші задачі.

Особливість ШНС – навчання. Під час навчання виявляються зв’язки нейронів, що визначають співвідношення вхідних і вихідних сигналів [57]. При навчанні ШНС починає виявляти додаткові залежності між вхідними та вихідними даними. Коли мережа навчена, то вона може отримати правильну відповідь і для нових даних, що не знаходились в початковій вибірці. Також

правильне рішення буде отримано, якщо вихідні дані неповні. ШНС дає можливість автономно отримувати й обробляти дані і вміння узагальнювати дані.

НС комбінується з різними типами архітектур (рис.1.2). Найпопулярнішим застосуванням НМ є рекурентні та прямого поширення архітектури.

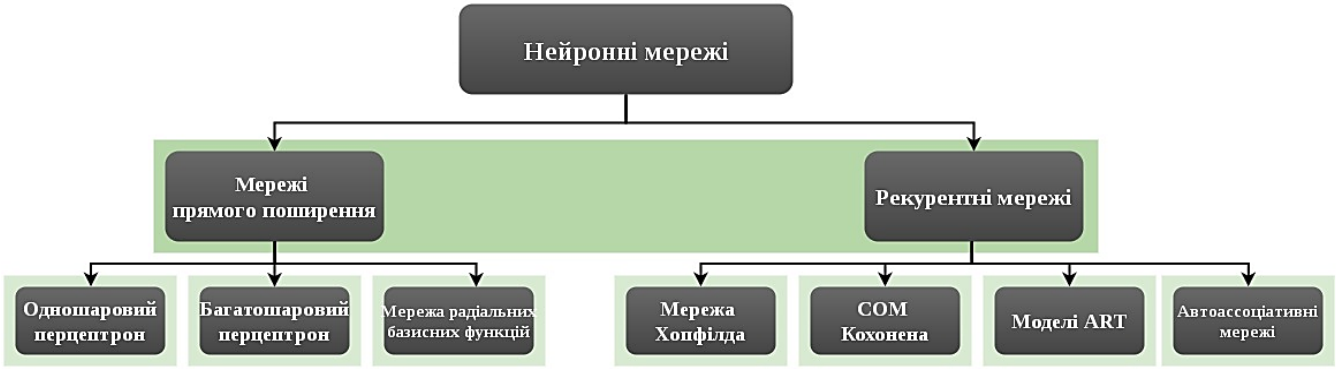


Рис. 1.2. Класифікація ШНС

Навчання ШНС може проводитись по вихідним даним, без вихідних даних (ШНС продукує рішення з вхідних даних) і з підкріпленням (використовуючи штрафи). ШНС можна реалізувати існуючими комплексами (Neuro solutions [104], MATLAB [103]) або відомими алгоритмами. Для побудови такої мережі необхідно мати навчальні зразки [58]. Формування навчальної вибірки залежить від процесу і мети впровадження.

ШНС може бути впроваджена і при побудові систем захисту мереж. На основі ШНС можливо організувати виявлення несанкціонованої поведінки в мережах [56, 58].

Головним при формуванні ШНС є налаштування початкової вибірки, бо при помилці процес навчання буде проходити багато часу і споживатиме багато ресурсів [59-61]. Після задання вибірки процес незворотній, оскільки повністю буде контролюватися закладеним алгоритмом. Він може лише подавати сигнали на вхід і перевіряти на виході. Через це відстежування роботи ШНС досить складний. Топологія мережі підбирається відповідно до середовища впровадження [62].

Метод має ряд недоліків, оскільки неможливо з першого разу коректно підібрати всі функції і їх описати. Але якщо налаштувати всю систему і протестувати її до впровадження в мережу, то можна отримати чудові показники

як швидкості, так і надійності. Найголовнішим буде описати алгоритм з врахуванням всіх тонкостей процесу.

1.3.4 Мультиагентні системи

МАС [63] складаються з агентів, що виконують поставлені перед ними задачі. Вони опрацьовують тільки ті задачі, на які розраховані. Між собою такі агенти незалежні і між ними не відбувається суперечок. Основна задача розбивається на підпункти, опрацюванням яких відповідають відповідні агенти. Правильна організація агентів дасть змогу правильно, швидко і коректно завершити задачу, яка виконується через розподілення всієї задачі на під задачі [64]. Для вирішення завдання агенти групуються під керівництвом центру операцій.

МАС добре проявили себе при захисті від атак відмови в обслуговуванні (DDoS) [65, 66, 98], результатом яких є вихід з ладу хостів, служб чи зупинка DNS-серверів і порушення функціонування мережі.

В роботах [67, 68] запропоновано даний метод при опрацюванні DDoS атак. Програмуються агенти користувачі, порушники та захисники, котрі можуть взаємодіяти. Далі відбувається поділ агентів порушників на «демони» (власне атака) і «майстри» (управляють атаками) (рис. 1.3). Поділ захисників йде на: «семплери», «детектори», «фільтри» і «агенти розслідування». «Семплер» накопичує інформацію і передаю в «детектори», які відповідають за реагування на початок атаки. «Фільтри» моніторять вхідну інформацію, а «агенти розслідування» протидіють агентам атак.

Рис. 1.3. Класи агентів атак і захисту

В [67] створено агентно-орієнтована система, в якій показано роботу агентів для виявлення вторгнень в мережі. Результати тестування показали задовільну ефективність методу в лабораторних умовах.

Даний метод [68] має достатньо позитивних сторін як для самостійного використання, так і для комбінування його з іншими методами для підвищення продуктивності. На даний час всі модифікації застосовуються взаємно з методами як ШІМ так і ШНС, оскільки цьому сприяє достатньо висока адаптивність МАС.

1.3.5 Штучні імунні системи

Штучні імунні системи - один з інструментів виявлення НД. Вони тісно взаємодіють з НМ, генетичними алгоритмами і ШІМ [40-42, 59]. ШІМ, що працюють на основі імунологічних методів, котрі вперше були виявлені в медицині і базувались на роботі лейкоцитів [69-72]. Природна імунна система складається з багатьох функціональних комплексів. Функція імунної системи в класифікації клітин організму для організації імунної відповіді.

Аналіз робіт і моделей [32, 73-78] дає змогу виділити характеристики імунних систем котрі можуть бути пристосовані до виявлення НД в комп'ютерних мережах:

–використання антитіл для опрацювання антигенів (по аналогії формування відповіді на вторгнення до комп'ютерної мережі);

–використання антигенів як вторгнень і формування відповідних імунних відповідей за допомогою антитіл;

–можливість навчання антигенів на роботу по виявленню антигенів. Клонування, селекція та видалення дають змогу швидко і максимально коректно за допомогою антитіл виявляти антигени;

–можливість формування імунної відповіді на будь-яке вторгнення (за допомогою збереження антитіл що відповідають антигенам);

–регулювання режиму відбору потрібних антитіл шляхом роботи з подібністю антитіл (поріг відповідності антитіла антигену). Подібність дає можливість протидії антигену максимально близьким йому антитілом;

–імунна система дає можливість запам’ятовувати і зберігати всі антитіла, що відповідають відомим системі антигеном (формується пам'ять імунної системи, що функціонує як база даних сигнатур);

–використання антитіл з найвищою подібністю дає змогу реагувати на нові невідомі вторгнення;

–можливість масштабування і адаптивності.

За допомогою опису функцій паратопів і епітопів антитіл [72, 77] можна створювати математичні моделі ШІМ і застосувати їх при впровадженні в інформаційні системи. Саме за допомогою паратопів і епітопів відбувається з’єднання антитіл і антигенів. Такі системи мають широкий спектр застосування, починаючи від інструментів оптимізації, систем сканування та розпізнавання образів, побудова комп’ютерних та мережевих систем та закінчуючи класифікацією інформації [73, 74]. ШІМ можуть бути застосовані при вирішенні наступних задач: комп'ютерна безпека, оптимізація чисельних функцій, комбінаторна оптимізація, навчання, біоінформатика, робототехніка, адаптивна система управління, виведення даних, виявлення аномалій або діагностика помилок [73, 79].

Імунні системи являються універсальною моделлю, за допомогою яких можна вирішувати різноманітні задачі. На даний момент найбільшого інтересу ШІМ представляє для застосування в комп’ютерних системах. Саме в роботах [80, 81] принципи ШІМ використані для виявлення вторгнень і атак в мережах. Робота [82] описує створення ШІМ з елементами моніторингу процесів на принципах негативної селекції (видаленням непотрібних антитіл з матриці пам’яті) при виявленні відмінностей у діях користувача і атаках на систему.

Для описання роботи ШІМ використовуються різні методи на основі диференціальних рівнянь затримки чи похідних, а також моделях на основі агентів та стохастичних диференціальних рівняннях.

В [83] розроблено модель виявлення несанкціонованих дій за допомогою елементів імунних моделей. В [84] описано роботу аномалій, котрі застосовуються при побудові гібридних елементів мережевої безпеки і здатні

виявляти атаки з низьким рівнем помилкових спрацьовувань. В таких системах є недолік, котрий полягає в генерації великої кількості трафіку атак перед початком роботи методу. Таким чином дані методи не можуть працювати в умовах реального часу. Розроблений адаптивний підхід на імунних механізмах. ШІМ повторює поведінку захисту імунних систем живих організмів. Наведені роботи демонструють перспективність ШІМ і при опрацюванні помилок. В [85] описано застосування ШІМ для опрацювання помилок.

В [86] описано роботу агентів, при реалізації виявлення несанкціонованих дій. Дані методи також базуються на алгоритмах імунних систем живих організмів. Агенти виступають в якості моніторів один для одного при виконанні загального для них завдання. Кожен агент контролює інших агентів на відповідність їх виконуваній задачі. Якщо агенти не справляються з задачею – то вони видаляються і заміняються іншими. Частіше за все агенти працюють на алгоритмах дендритних клітин імунної системи. Такі агенти (ДК-агенти) і штучні Т-агенти можуть будувати адаптивні імунні підсистеми. Антигени описують роботу вторгнень і дають змогу на основі антитіл формувати необхідну відповідь на поставлену задачу. Система підагентів заснована на збудженні ДК-агентів для сигналів і ТС-агентів для антигенів. Агентний метод дає можливість для реалізації навчання мережі. Імунна відповідь на несанкціоновані дії активується з центру безпеки. В такій системі комп’ютерні хости розглядаються як вхідні сигнали, а тимчасові вихідні сигнали будуть характеризувати небезпечні чи безпечні сигнали.

Робота [35] описує метод комбінованого використання ШНС і ШІМ при роботі з трафіком. Така система самостійно навчається на вхідних даних для можливості самостійного виявлення вторгнень. Головними перевагами такої системи є автономність і висока точність виявлення НД. Низька швидкодія обумовлена тривалим процесом навчання.

Робота [64] описує створення мультиагентної ШІМ з можливістю ведення статистики уразливих елементів мережі. Підхід дозволяє не лише уважніше стежити за такими вузлами, а й тестувати їх на предмет вторгнення. Враховано

можливість передбачення значень часових рядів за допомогою ШІМ. В роботі також розроблено метод клонального відбору при оцінці коректності даних. Робота [97] основана на застосуванні методів діагностики захворювань в медицині для побудови систем виявлення комп’ютерних атак застосуванням теорії ШНС. Пошук несанкціонованих дій виконано на основі об’єднання методів імунних і нейронних мереж.

Наведений аналіз сучасних методів виявлення вторгнень показує, що основний напрямок підвищення ефективності систем лежить у комбінуванні різних методів і технологій для вирішення задач [146]. Також постає необхідність в створенні програмних систем, що допоможуть підвищити надійність та ефективність існуючих систем виявлення несанкціонованих дій в комп’ютерних мережах.

1.3.6 Діагностика

В області комп'ютерної безпеки багато ресурсів спрямовано на підвищення ефективності захисту користувачів від несанкціонованих дій в кіберпросторі. Одним зі способів підвищення безпеки комп'ютерної системи є використання системи виявлення вторгнень [4, 87, 88]. Метод діагностування пов'язаний з системами виявлення вторгнень, але оснований на принципах розвинутих в медицині. За допомогою інформатизації в сфері медицини стає можливим і розвиток нових віх комп'ютерних технологій. На жаль такі методи часто являються суто теоретичними, тобто такими які працюють лише на папері та математичних моделях [89-92]. Основна складність при їх проектуванні - це підбір правильної технологічної та методологічної бази, що дасть змогу застосувати такі методи в реальних умовах. Такі методи базуються на діагностуванні вторгнень [93, 94].

При діагностуванні вторгнень моніторяться відповідні функції системи, таким же чином як в медицині проводиться обстеження і виявляються симптоми хвороби. Симптоми використовують значення обстежень, щоб обчислити ймовірність виявлення певної хвороби в організмі людини. Бінарні симптоми, як

особливий клас симптомів, використовують алгоритми виявлення зміни: порогові або CSM, щоб вираховувати і формувати сигнатури атак чи несанкціонованих дій. Теорія Демпстера-Шафера застосовується для характеристики таких переконань, оперуючи основними поняттями по комбінуванню та використанню операторів злиття [95-97].

Діагноз ставиться шляхом аналізу ймовірності достовірності оцінки різних наборів станів структури проникнення комбінованих доказів симптомів для дерева діагностики, представленого деревом специфікації, щоб максимально точно визначити захищеність системи чи організму. За допомогою такого процесу, система виявлення вторгнень може поєднувати в собі характеристики як сигнатур так і систем виявлення аномалій, і може виявляти відомі чи нові атаки та несанкціоновані дії системи.

Система здатна виявляти раніше відомі й нові атаки, що подібні до типів попередньо задекларованих системою вторгнень. Побудовані таким чином системи показували хорошу ефективність при роботі з наперед заданими атаками типу відмови в обслуговуванні (DoS) [65, 67, 98]. Система виявлення вторгнень була побудована і налаштована проти кількох відомих випадків TCP та DoS атак, які метод був здатний правильно діагностувати під час виконання. Були також задекларовані псевдонові вторгнення, котрі не були заздалегідь відомі системі. Такі типи вторгнень були побудовані з характеристик відомих атак для тестувальної системи, і були цілком коректно продіагностовані та виявлені.

Існуючі програмні рішення мають певну підтримку з точки зору запобігання і виявлення вторгнень, але в них відсутня можливість діагностування. Є передумови для застосування методів діагностування з медицини в комп’ютерних системах. Наявні методи не в змозі повністю виявляти всі НД в мережах, тому комбінування нових підходів з наявними методами можуть давати хороші результати і підвищити надійність. Дані системи мають недоліки в своєчасності детектування атак, але використання методів діагностування може позитивно вплинути на швидкодію відомих методів, мінімізуючи витрати на проведення моніторингу мережі й потоків. Основна ідея полягає в зборі інформації на кількох

архітектурних рівнях, з використанням декількох фільтрів безпеки для виконання кореляційного аналізу симптомів вторгнення. Це дозволяє виявляти спочатку симптоми вторгнення, а потім і їх причини. За їх допомогою можна проводити оцінку збитків в окремих компонентах системи. Попередні теоретичні методи показали ефективність, але не були реалізованими в реальних умовах і системах.

Говорячи про діагностику, ми маємо на увазі здатність чітко визначити причини вторгнень і оцінювати їх наслідки на індивідуальну систему компонентів [99]. За допомогою діагностування можна не лише виявити вторгнення, а й завчасно його попередити, використовуючи методи побудови дерева діагнозів (схожих на сигнатури, але працюючих без оновлень і виявляючих нові форми атак і модифікації старих).

Дана технологія може розширити можливості систем виявлення вторгнень піднімаючи їх на якісно новий рівень. Основна ідея полягає в тому, щоб зібрати інформацію на декількох архітектурних рівнях (мережі, операційної системи, баз даних та додатків) за допомогою фільтрів безпеки та використанням технології обробки складних подій, для виконання кореляційного аналізу симптомів вторгнення [100, 101].

Ідею збору інформації від джерел теоретично описано в роботі [102]. Представлені в роботі методи використовують поняття кореляції та мультианазілу, але не вирішують проблему діагностики аномалій в системі. У запропонованому підході розглядається процес ескалації від симптомів вторгнення до визначення причин вторгнення та оцінки завданих збитків за допомогою онтологій. Два набори працюють попарно: перший дозволяє нам спостерігати за симптомами, а другий дає можливість виносити вердикти про наявність атак чи аномалій. Вихідні дані цього процесу потім можуть бути використані для відновних процесів, і в кінцевому підсумку для забезпечення надійності виявлення несанкціонованих дій.

Теоретичні випробування показали, що такий підхід поліпшує результати виявлення несанкціонованих дій з точки зору підвищення достовірності та надійності процесу прийняття рішень. Даний метод може виявляти нові атаки не

маючи ніякої інформації про них [136, 145]. Передбачена можливість інкапсуляціїї сигнатур, що дасть змогу не лише виявляти вторгнення, а й визначати клас і тип атак.

1.4. Висновки до першого розділу

Таким чином, у першому розділі дисертації проведено аналіз сучасних підходів до виявлення НД в комп'ютерних мережах. У результаті аналізу виявлено недоліки існуючих методів, моделей та систем, які не дозволяють ефективно реалізовувати процедури виявлення та вчасного реагування на вторгнення.

Проведений аналіз проблеми розпізнавання НД показав, що існує необхідність у розробці нових підходів до виявлення та аналізу вторгнень, які повинні бути засновані на аналізі їх поведінки та діяти в обхід шифрування. Нові підходи повинні ефективно виявляти відомі й нові модифікації НД та мінімально завантажувати систему. Система також повинна захищати комп'ютерні мережі без необхідності постійного оновлення антивірусного програмного забезпечення. На основі досліджень встановлено, що доцільною є розробка методів на основі комбінування різних технологій обробки інформації, в яких взаємно компенсуються їх недоліки та об'єднуються сильні сторони. Це дозволить створювати методи пошуку й аналізу альтернативних рішень, на основі принципів навчання та адаптації баз знань до роботи в зовнішньому середовищі для отримання ефективних рішень.

РОЗДІЛ 2

ОРГАНІЗАЦІЯ РОЗПІЗНАВАННЯ НД ЗАСОБАМИ ШТУЧНИХ ІМУННИХ МЕРЕЖ

2.1. Модель аналізатора несанкціонованих дій

Вторгнення в комп’ютерну мережу може загрожувати безпеці як даних, так і самій системі. Сьогодні кіберзлочинці використовують будь-які способи — починаючи від шпигунства і злодійства особистих даних і закінчуючи вимаганням грошей за доступ до інформації.

У більшості випадків атаку провокують самі користувачі і компанії, в яких вони працюють, тому що економлять на захисті інформаційної безпеки або не знають, які антивірусні програми слід встановити на комп'ютер, смартфон або планшет.

Найбільш поширеним є шахрайство у фінансовій сфері, коли злочинці отримують несанкціонований доступ до великих грошових активів. Не менш небезпечні спамние розсилки з різними повідомленнями і звітами. Користувач відкриває лист, і на його комп'ютер встановлюється небезпечна шкідлива програма — троян, шифрувальник і т.д. Вона сприяє сливу конфіденційної інформації, витік даних і втрати доступу.

Для захисту інформації необхідно ретельно відбирати сайти, які відвідуєте в інтернеті. Аналіз інформації записів у мережевих процесах являється важливим способом вирішення проблем в області мережевої безпеки [24, 36, 133].

Використання інструментів виявлення НД залежить від сферами застосування. Чим вужчою є сфера застосування, тим простіше застосовувати до неї ті чи інші інструменти дослідження.

Хорошу продуктивність показують системи, які базуються на роботі штучних імунних мереж. Відмінною особливістю ШІМ є те, що вони не програмуються, а навчаються. Це одна з головних їх переваг перед традиційними алгоритмами. Навчання складається з взаємодій між антитілами і антигенами, які виконують роль системи виявлення вторгнень (антитіла) і несанкціонованих дій

(антигени). ШІМ основною характеристикою має можливість навчання, що дає змогу застосовувати її для будь яких задач (при можливості доступу до навчальної вибірки). До недоліків таких систем входить складність налаштування, що не завжди може підходити користувачам без відповідного рівня знань даної тематики.

Деякі види НД можна виявити за допомогою систем виявлення вторгнень [87, 89]. Вони використовуються для виявлення деяких типів шкідливої активності, яка може негативно вплинути на працездатність обчислювальної машини чи привести до втрати даних. Такими діями можуть виступати атаки на вразливі сервіси, несанкціонований доступ до чутливої інформації і віруси.

Популярне застосування ШІМ для опрацювання трафіку і на його аналізі виявлення вторгнень[135, 137, 146].

Розвиток поліморфізму сучасних вірусів і ускладнення їх коду еволюціонують настільки швидко, що навіть комбінування емуляції з різними аналізаторами коду не завжди є ефективним. Поясненням цьому бувають:

- високий рівень поліморфізму;
- використання пакувальниками різних віртуальних машин з закритою архітектурою. Через це розшифроване тіло атаки може виконуватись на емуляторах частково, а дане НД не буде виявлене;
- використання алгоритмів відкладеного запуску, що дає змогу запускати вторгнення після закінчення роботи емулятора.

Виходить, що проводити пошук для кожної його модифікації необхідно вручну, а це складний та повільний процес.

Більшість алгоритмів маскування кодів НД працює з популярними аналізаторами коду, котрі в роботі застосовують поведінкові методи аналізу. Робота поведінкових методів основана на використанні системних викликів для виявлення несанкціонованої діяльності. Популярними є системи, що відстежують час на виконання задач, а при виникненні затримок чи відхилень від усереднених значень – продукують сигнали тривоги [138, 142, 143]. При моніторингу пакетів мережі важливо вибирати лише параметри які будуть впливати на кінцевий

результат, відкидаючи ті, що можуть негативно вплинути на швидкодію як методу, так і самої системи [144]. Також є можливість спостереження за відхиленнями в роботі мережевих об’єктів (проведення моніторингу) [141, 143]. Аналізатори коду можливо замінити менш громіздкими аналізаторами даних, отриманих за допомогою НМ, ШІМ, мультиагентних систем тощо [41, 45, 51, 55, 56, 73, 75, 82, 88, 94, 97, 102, 134, 139].

Для обробки НД евристичним аналізатором (ЕА), доцільно використовувати функціональні можливості ШІМ. ЕА несанкціонованих дій, заснований на такому методі матиме вигляд (рис. 2.1):

Рис. 2. 1. Модель ЕА вторгнень

1. Блок 1. Виконується спостереження за поведінкою всіх об’єктів системи (як шкідливих, так і не шкідливих) для отримання їх протоколів, що будуть містити як дані застосованих функцій, так і використовані ними аргументи).

2. Блок 2. За вхідну інформацію приймаємо вихідні дані з попереднього кроку і виконуємо їх порівняння. Метою кроку буде отримання фрагментів одного типу.

3. Блок 3. Відповідає за збереження ознак отриманих даних. Додатково ведеться облік збережених фрагментів, на основі якого ознака отримує відповідний рейтинг, який характеризуватиме частоту появи даної ознаки. Кінцевою метою даного кроку буде присвоєння рейтингу всім ознакам, на основі

яких буде вирахувана частота їх появи.

4. Блок 4. Відносить дії в системі до шкідливих чи нешкідливих.

Вимоги моніторингу:

1. Обробка команд CPU.

2. Робота з файловою системою та реєстрами.

Головною метою цього блоку буде отримання імен, до яких відбувається звернення функцій WinAPI, даних про виклики і покрокову роботу інструкцій.

До блоку потрапляє файл, що виконується в штучному середовищі. При опрацюванні запущених програм формуються протоколи роботи. Після завершення роботи отримана інформація передається в наступні блоки. Описаний вище модуль можна реалізувати такими способами:

1 Емуляція. Використовуючи емуляцію, можливо використовувати вже готові програмні рішення, котрі в повному обсязі зможуть гарантувати дотримання вище описаних вимог до блоку моніторингу. Головними функціями емулятора при реалізації блоку будуть: інтерпретатор, завантажувач і бібліотека функції. Далі інтерпретатором буде реалізована модель процесора, котра стане моделювати весь набір використовуваних команд. Від інтерпретатора буде залежати швидкість обробки операцій. Чим швидше будуть опрацьовані відправлені команди, тим менше часу знадобиться аналізатору для видачі результатів роботи.

Завантажувач запущених файлів дозволить зчитувати дані про використання пам’яті виконуваних файлів і побудувати таблиці переходів для задіяних функцій. Також завантажувач опрацьовує шляхи запуску програм для обробки інтерпретатором.

Бібліотека функцій містить моделі системних бібліотек, котрі не мають ніякого впливу на реальну систему. Змодельовані функції записують звернення програм користувача до системних функцій. Потім з них формується статистика відповідних звернень. Взаємодія описаних компонентів зображена на рис. 2.2.

Для побудови модуля моніторингу доречно використовувати емулятор [36]. Він задовольняє всі вище зазначені вимоги.

Рис. 2.2. Виконання моніторингу

2. Моніторинг. При відсутності емулятора відпаде необхідність використання вищезгаданих компонентів крім завантажувача, так як вони відповідають за створення відповідного середовища для експерименту. Особливістю підходу буде підвищена швидкодія системи виявлення несанкціонованих дій. Негативною стороною – зменшення ступеня захищеності, оскільки деякі вторгнення можуть негативно вплинути на реальну систему чи пошкодити її.

При роботі системи для збору даних буде застосовуватися зчитування API функцій за допомогою інкапсуляції інструкцій, що можуть реалізувати передачу управління заданому обробнику.

Блок порівняння кінцевою ціллю має реалізацію виявлення спільних особливостей запушених процесів. Даний блок опрацьовує дані з емулятора. На виході отримуємо загальні фрагменти вхідних протоколів, обробка яких відбувається наступним чином:

1. Видалення непотрібних даних з вхідних протоколів (тих, що не мають корисної інформації про вплив на файлову систему, системні реєстр і процеси чи мережу). З усіх вхідних протоколів усуваються дані про:

- файлову систему;
- Internet;
- роботу вікон;
- реєстр;
- мережу;
- процеси.

Даний етап можливо реалізувати інструментами Deterlab [124], котрі дають можливість поділяти протокол на складові, що оброблюються різними потоками. Таким чином при видалення непотрібних даних дозволить зменшити час на проходження всього процесу по виявленню вторгнень.

2. Після видалення лишньої інформації починається пошук спільних фрагментів, шляхом порівняння протоколів.

Спочатку порівнюються фрагменти максимального розміру, поступово зменшуючи розмір фрагментів на одиницю. При зменшенні розміру фрагменти одного масиву порівнюються з фрагментами іншого. Таким чином, кожен елемент одного масиву буде порівняний з усіма елементами другого. Після порівняння всіх фрагментів другого масиву з першим, фрагмент першого масиву зміщується на одиницю, й операція повторюється. Порівнюються лише елементи однакового розміру (як було зазначено вище) на предмет наявності в них подібних частин (розташування яких не зіграє ніякої ролі).

Наступний етап виконується порівнянням вхідних протоколів однакового розміру з емулятора. В результаті будуть отримані ідентифікатори використовуваних потоків, з яких сформується список однакових фрагментів в різних протоколах з зазначенням їх розміру і зміщення в масивах (це дозволить виявити їх місце розташування).

Бібліотека ознак забезпечує зберігання фрагментів протоколів. У фрагментах зберігається інформація про API функції (тип і значення аргументів). Для збереження фрагментів в бібліотеці їм буде призначатись ознака *O* і рейтинг появи *R* за формулою:

$$R = O/O_{all},$$

де *O* – це об’єкти, що міститимуть дану ознаку;

O_{all} – це сумарна кількість об’єктів.

Прийняття рішень – забезпечує розпізнавання НД певного сімейства і не НД та реалізовується різними технологіями інтелектуальної обробки інформації на основі ШІМ.

2.2. Виявлення ознак для аналізу дій в мережі

Для виявлення НД потрібно визначити ті ознаки, котрі будуть характеризувати несанкціоновані дії. Для розпізнавання НД потрібно мати бібліотеку подій, що впливають на стан системи. Такі бібліотеки можна сформувати за допомогою аналізу НД і виділенні з них ознак з високим рейтингом появи. Головною задачею буде збір набору таких ознак.

Запущена програма впливає на ЦП, регістри, вміст пам’яті тощо. Однак не всі ці ознаки потрібні для виявлення НД. При роботі НД і не НД за допомогою різних аналізаторів можна вирахувати рейтинги появи подій, з яких найбільший інтерес представлятимуть саме події, котрі частіше виникають в НД. Можуть проявитися хибні спрацювання, тобто ті, що не будуть характеризувати НД, а це може знизити ефективність методу. Все це свідчить про те, що кінцевий варіант виділених ознак краще все-таки висилати на перевірку адміністратору.

При поведінковому аналізу частіше за все приймаються до уваги загальні ознаки, що характерні більшості НД. Перевага методу в тому, що при коректній обробці загальних ознак вони будуть успішно виявляти поширені НД, що використовують схожі алгоритми.

Недоліком буде те, що для обходу такого захисту потрібно визначити типи подій, які опрацьовує аналізатор для заміни їх подібними.

Інший спосіб дозволяє виявляти НД, котрі властиві певним типам вторгнень. Для коректності виявлення таких унікальних подій необхідно зробити якомога більше їх перевірок. Виходить, що чим унікальніші події, тим точніше будуть виявлятися вторгнення і нижчим буде поріг помилок. Типи подій, котрі матимуть високу унікальність, найчастіше пов’язані з:

- керуванням файловою системою;
- керуванням процесами;
- керуванням роботою мережі;
- керуванням системним реєстром;
- роботою системних подій.

Також негативний вплив матимуть спільні для НД і не НД події. Необхідно аналізувати виникнення подій у шкідливих і не шкідливих об’єктах для мінімізації помилок. Проведений аналіз допоможе розділити системні події на віруси відомих типів та віруси нових типів і не НД.

При правильному налаштуванні вище описаний метод поведінкового аналізатора зможе самостійно виявляти вторгнення без застосування сигнатур.

2.3. Вибір типу імунної системи

Специфікою роботи з імунологічними механізмами є використання різноманітних інстументів, головним з яких виступає взвємодія антитіл (A) і антигенів (G). Взаємодія антигенів пояснюється в числених роботах [77, 84]. В ШІМ зміна вибірки антитіл A^{gen} в заданому поколінні (gen) описується виразом [86]:

$$A^{gen+1} = Edit(Mut(Cl(Sel(A^{gen}))), A^{gen}),$$

де приведено наступні оператори опису системи (у порядку виконання):

Sel – відбір антитіл;

Cl – клонування відібраних антитіл;

Mut – мутація клонованих антитіл і створення вибірки клонів антитіл;

$Edit$ – редагування антитіл початкової вибірки і вибірки створеної через мутацію для отримання нових антитіл.

Дане рівняння описує загальний вид моделі, в яку на кожному кроці проектування можливо вносити додаткові елементи обробки задіяних тіл.

Роботи [32, 73-78, 84, 86] присвячено проектуванню таких систем, але більшість з них мають теоретичний характер, що пояснюється складністю їх опису через недостатню конкретизацію основних механізмів роботи. Загальновживаним являється поділ всіх типів моделей на п’ять загальних:

1. Клонального відбору.
2. ШІМ.
3. Позитивного і негативного відбору.
4. Дендритні.

5. Моделі теорії небезпеки.

Нижче наведені приклади використання всіх цих моделей при виявленні вторгнень. Описано характеристики моделей, їх переваги і недоліки, а також розглянуті питання реалізації процусу виявлення НД.

2.3.1. Модель клонального відбору

Дана модель використовує властивості природнього імунітету протидіяти хворобі. В живому організмі для подолання недугу формується група антитіл (A) певного призначення для протидії чужорідним антигенам (G). При хворобі система залишає ті A, що можуть протидіяти G (хворобі). A мають рівень протидії G (він називається подібністю), і чим вона вища – тим сильнішим є A при витисненні G. Подібність дозволяє продукувати адекватну відповідь на хворобу, завдяки чому антитіла з високою подібністю мають можливість клонуватися в організмі для подолання хвороб. Саме введення функції подібності дозволяє порівнювати і відбирати леше ті антитіла і в необхідній кількості для цілковитого подавлення чужорідних елементів. Антитіла з найвищою подібністю формують імунну пам’ять організму, що здатна боротися з антигенами (рис. 2.3).

Головні імунні аспекти при описі клонального алгоритму: зберігання певного набору клітин пам’яті, відбір та клонування найбільш придатних антитіл, загибель нестимулюючих антитіл, дозрівання подібності та повторний набір клонів пропорційно їх антигенної подібності та генерація і забезпечення різноманітності [29].

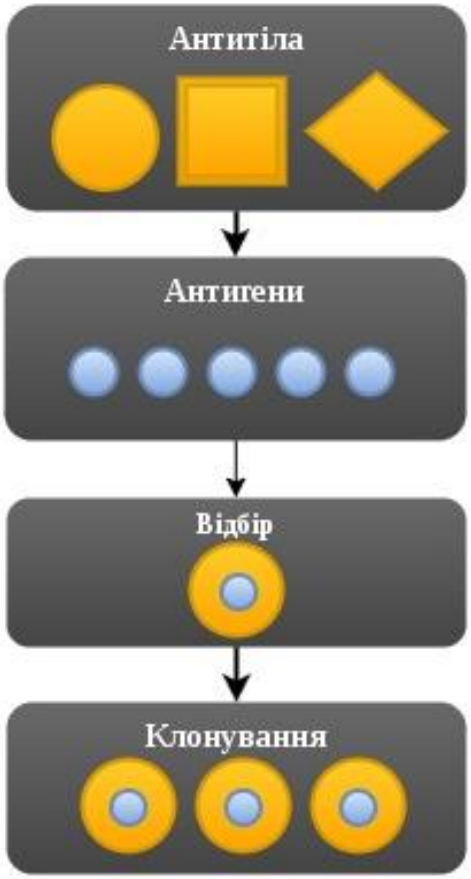


Рис. 2.3. Схема роботи клонального відбору

Оператори клонального відбору несуть відповідальність за:

- клонування Антитіла;
- мутацію клонованих Антитіл (для досягнення максимального рівня відповідності вторгненню);
- видалення слабких Антитіл.

Головними методами клональної селекції є CLONALG і BCA [106, 107], що допомогають при виявленні графічних об’єктів (дають змогу їх обробляти і знаходити потрібні для користувача рішення).

Метод CLONALG був розроблений для завдань машинного розпізнавання графічних об’єктів. Кастро і Зубен [106] запропонували клональний алгоритм відбору CLONALG для навчання і оптимізації. CLONALG генерує вибірку з *n* А, кожна з яких зводиться до потрібного виду через процес оптимізації. На кожній ітерації деякі з кращих існуючих А відбираються, клонуються і мутують, щоб

побудувати нову популяцію-кандидата. Потім оцінюють нові А і додають певний відсоток кращих А до вихідної вибірки. Нарешті відсоток найгірших А попереднього покоління замінюється на нові, випадково створені. Пізніше цей метод адаптували для задач оптимізації. Метод характеризують: наявність клітин пам’яті, відбір і клонування найбільш корисних А, вмдалення найменш корисних А, вибір клонів пропорційно їх корисності, забезпечення і збереження різноманітності вибірки А. У [107] представлено вдосконалений алгоритм відбору клонів на основі CLONALG, з новим мутаційним методом – самоадаптивною хаотичною мутацією. Основні модифікації полягають в тому, що новий алгоритм використовує логістичну хаотичну послідовність для генерації початкової сукупності А. При цьому гіпермутація приймає самоадаптивну хаотичну мутацію.

Метод BCA [108] виконує пошук глобального оптимуму цільової функції та може забезпечити можливість пошуку при будь-яких параметрах заданої початкової вибірки. Особливістю методу є робота оператора суміжної гіпермутації, яка передбачає мутацію не окремих випадкових компонентів клітини, а цілої області декількох суміжних (сусідніх за номерами) компонентів.

Особливості клонального методу:

- взаємодія А і клонів з G;
- відсіювання клонів;
- заміна мутованих клонів;
- заміна клонованих А.

Взаємодія можлива при представленні G і А чи клонам, або А і клонам. Такий процес називається навчанням, і його метою є відсіювання слабких і непотрібних А. Навчання можливе через вирахування значень подібностей.

Алгоритм такого відбору працює наступним чином [29]:

Крок 1. Ініціалізація. Створення (звичайною випадковою генерацією) початкової вибірки антитіл).

Крок 2. Обчислення подібності. Для антитіла обчислити його подібність антигену, а результати записати в матрицю подібностей.

Крок 3. Клональний відбір. Вибрати з вибірки по *n* найкращих антитіл для кожного елементу матриці *D* і помістити їх створену популяцію клонів. Згенерувати клони елементів популяції A пропорційно до їх подібності; тобто, чим вища подібність, тим більше створюється клонів і навпаки.

Крок 4. Дозрівання. Піддати мутації всі клони популяції A з імовірністю, обернено пропорційною їх подібностям (чим нижча подібність індивідуума, тим вища ймовірність його мутації). Обчислити нову подібність кожного антитіла j аналогічно до кроку 2, одержавши нову матрицю подібностей *CD*. Вибрати з вибірки An антитіл, для яких відповідний вектор-стовпчик матриці *CD* дає кращий узагальнений результат подібності, і перенести їх в вибірку клітин пам’яті.

Крок 5. Метадинаміка. Замінити d гірших антитіл популяції A новими.

Крок 6. Замінити *n* антитіл популяції A клітками пам’яті і переходити до кроку 2 до досягнення критерія зупинки.

Перевагою клональної селекції є підтримка постійного розміру вибірки антитіл [29].

Мінімізація кількості створених клонів забезпечує спрощення обчислювальних операцій та дозволяє пришвидшити процедуру навчання, клонування та відбору. Мінімізація дає змогу з мутованих клонів обирати об’єкти з найвищими подібностями до антигенів. Також при клональному відборі визначаються клітини імунної пам’яті з тих об’єктів, котрі знаходяться в стані специфічності у представленому антигені. В такому стані подібність антигена і клона буде максимальною.

Редагування вибірки за допомогою оператора старіння можливе лише для клонованих антитіл. Таке редагування відбувається шляхом порівняння подібностей вибраного клону і клонованого антитіла з антигенами. Якщо подібність клону відповідає антитілу, то цей клон передається до вибірки антитіл. Якщо подібність клона нижча необхідного рівня, то клон видаляється.

2.3.2. Модель ШІМ

ШІМ описана в роботах [79, 83, 84] і дає можливості по вдосконаленню методик роботи антигенів і антитіл на основі формалізації зв'язків епітопів з паратопами. Дані зв'язки і процес роботи показано на рис. 2.4.

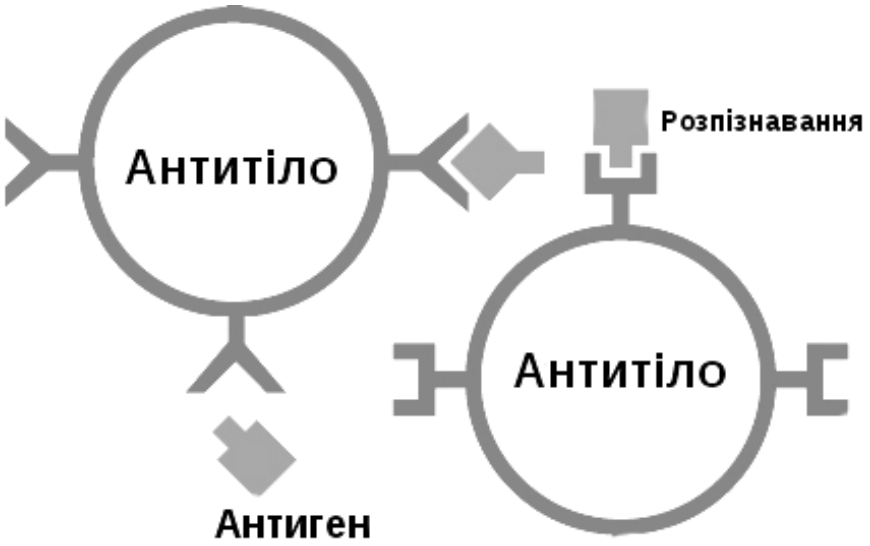


Рис. 2.4. Виявлення антигена

Зв'язки показують, що клони об'єднуються в мережу клітин, що можуть взаємодіяти. Виходить, що відсутність антигенів не впливає на архітектуру мережі, бо в ній лімфоцити об'єднуються через антитіла і можуть швидко реагувати на вторгнення в систему.

Влаштована через лімфоцити мережа антитіл може бути описана математичними рівняннями через опис антитіл і їх подібностей. Для характеризиції подібності (F) застосовуються рівняння взаємодії паратопів антитіл і епітопів антигенів:

$$F = \left[\sum_{\mathrm{K}=1}^{N} \mathrm{P} - (\sum_{N=1}^{L} (\mathrm{e(a + k)} + \mathrm{p((a)} - s_1))\right],$$

де, k - відхилення зарядів паратопа $\mathrm{p(a)}$ і епітопа $\mathrm{e(a)}$ антитіла;

P – кількість поколінь;

s – поріг проходження поколінь.

Епітоп – частина макромолекули антигену, яка розпізнається імунною системою (антитілами, В-лімфоцитами, Т-лімфоцитами). Частина антитіла, що розпізнає епітоп, називається паратопом. Щоб паратоп міг зв'язатися зі своїм епітопом, взаємодіючі ділянки повинні бути комплементарними по конформації, розподілу заряду і гідрофобності. При дотриманні цих умов формуються сполучення епітопів з паратопами. При перекривання електронних оболонок можуть виникати сили відштовхування в результаті тісного контакту поверхонь білкових молекул. Співвідношення сил тяжіння і відштовхування грає вирішальну роль у визначенні специфічності молекули антитіла і її здатності розрізняти структурно подібні молекули.

За допомогою подібності, описаної паратопами і епітопами, можна моделювати динаміку вироблення схожих антитіл. Для N антитіл $\{x_1, \ldots, x_n\}$ і антигенів $\{y_1, \ldots, y_n\}$, зміна x_i антитіл набуває виду:

$$\frac{dxi}{dt} = c\left[\sum_{j=1}^{N} F_{ji}x_i x_j - k_1 \sum_{j=1}^{N} F_{ji}x_i x_j + \sum_{j=1}^{N} F_{ji}x_i y_j\right] - k_2 x_i,$$

де, k_1 –нормування придушення;

$k_2 x_i$ – видалення x_i антитіл;

с – величина вироблення антитіл.

Суми членів послідовностей антитіл $\sum_{j=1}^{N} F_{ji}x_i x_j$ описують зв'язки з антитілами, а $\sum_{j=1}^{N} F_{ji}x_i y_j$ зв'язки з антигенами;

При виявленні антигена вся мережа налаштовується на продукування антитіл з необхідною подібністю. Саме з таких антитіл будуть робитись клонів. Створені клони мутуються і предоставляють всю анформацію про батьківські антигени, з якими потрібно боротися. Подальша робота з вибірками клонів і антитіл відбувається за допомогою стиснення мережевими інструментами відбору тіл з високою подібністю (а з низькою – видалення).

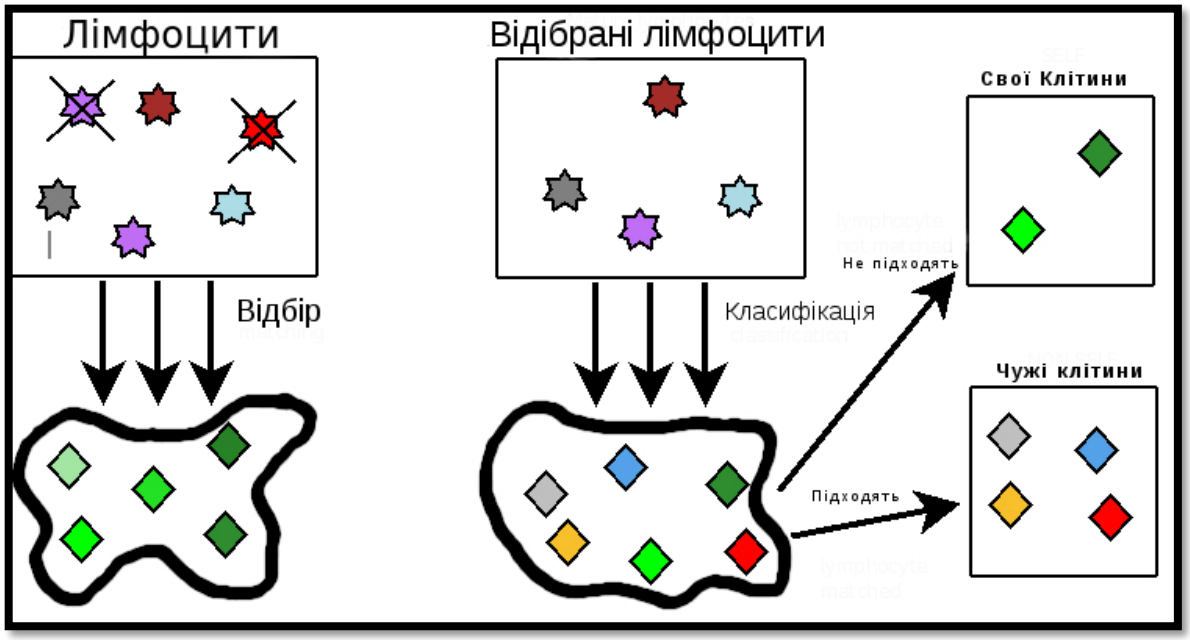
Налаштування вибірок антитіл і клонів йде через стиснення (визначення подібностей тіл і їх клонів). Стиснення малоефективне при відборі за ознаками

старіння, оскільки такі подібності міняють значення, а супресія відразу відкидає низькі значення.

Популярними інструментами побудови ШІМ є aiNET, opt-aiNET, RLAIS, MRLAIS [75-77, 109, 110].

2.3.3. Модель негативного/позитивного відбору

Моделі негативного/позитивного відбору найчастіше використовуються в ШІМ. Моделі негативного відбору переважають над позитивними. Такі методи класифікують всі клітини як «свої» і «чужі», що дає змогу системі коректно на них реагувати (рис. 2.5).



Мал. 2.5. Принцип негативного відбору

При негативному відборі застосовуються лімфоцити з рецепторами, що реагують на антигени занесені в організм. Антитіла формуються на поверхні лімфоцитів. Видаляються антитіла, котрі визначають «свої» клітини, а антитіла, що розпізнають антигени залишаються. Формується вибірка лімфоцитів, що виявляють антитіла.

Псевдокод негативного відбору може бути представлений у вигляді [110]:

input : S = set of patterns to be recognised, n the number of worst elements to select for removal

output : M = set of memory detectors capable of classifying unseen patterns

begin

Create an initial random set of antibodies, A

forall patterns in S do

Determine the affinity with each antibody in A

Generate clones of a subset of the antibodies in A with the highest affinity.

The number of clones for an antibody is proportional to its affinity

Mutate attributes of these clones to the set A , and place a copy of the highest affinity antibodies in A into the memory set, M

Replace the n lowest affinity antibodies in A with new randomly generated antibodies

end

end

Метод [111] на основі негативного відбору став основою створення алгоритмів по виявленню мережевих аномалій і вірусів. Базуючись на даному способі створені одні з перших імунних методів негативного відбору.

2.3.4. Модель теорії небезпеки

Дана модель основана на тому, що імунна система відрізняє ті вхідні сигнали, які завдають шкоду системі від тих, що не завдають. Дана теорія [112] базується на визначенні небезпеки для системи не лише новими вторгненнями, а й вже існуючими в системі вразливостями. Теорії небезпеки описує клітини в якості антигенів, котрі при наявності певних факторів (тригерів) можуть вказувати на наявність в організмі грибкових пухлин (такий самий метод аналогічно можна застосовувати в інформаційних системах, задаючи антитілам певні алгоритми дій на антигени). Сигнали небезпеки не надсилатимуться здоровими чи старіючими (зі зменшеною подібністю) антитілами.

Будь-яке тіло, що атаковане антигеном, посилає сигнал небезпеки для вироблення необхідної захисної реакції імунною системою (макрофагами чи лімфоцитами у випадку біологічних систем). Такі клітини виділяють антитіла, які

відповідають антигенам. Ці антитіла потім клонуються і використовуються для захисту (вирішення поставленої перед ними задачі). Антитіла, які не відповідають антигенам, будуть перебувати в стані спокою (зберігатися до появи відповідних антигенів).

Дана модель удосконалена у роботі Бретчера [113], яка описує теорію небезпеки необхідністю двох сигналів для активації лімфоцитів. Перший сигнал буде відповідати за розпізнавання антигенів; а другий засвідчуватиме те, що антиген є небезпечним.

Основні недоліки теорії небезпеки [112]:

• необхідність сигналу від антитіла для подання сигналу небезпеки;

• сигнал небезпеки може бути хибним;

• сигнали небезпеки можуть бути позитивними або негативними (свідчити про наявність або відсутність сигналу);

• оцінка близькості може бути розцінена як вторгнення (а також використана для моделювання роботи антитіл і антигенів).

Було запропоновано також концептуальні ідеї щодо теорії небезпеки для виявлення аномалій. Ґрунтуючись на теорії небезпеки, імунна відповідь завжди викликається сигналами небезпеки. Використання низької або високої активності оперативної пам’яті чи часу роботи процесора можуть вказувати на небезпечні сигнали. Імунна система може реагувати на антигени в небезпечній зоні лише тоді, коли з’являється сигнал небезпеки. Після виявлення антигенів вони надсилаються до системи для подальшої перевірки.

Теорія небезпеки може бути використана і для задач виявлення даних [114] в тих випадках, де кожен документ має набір атрибутів. Коли ШІМ реалізується, антитіла в системі використовуються для виявлення атрибутів. Кожен документ, який переглядається користувачем, буде фіксуватися антитілом.

Коли користувач звертається до даного документа, то піднімається сигнал небезпеки, і антигени, що відповідають антигену (атрибуту даного документа), спрацьовують та активуються (рис. 2.6). Атрибути документа реагують на

прикріплені антитіла, після чого навчена ШІМ може фільтрувати атрибути і знаходити вторгнення через звернення до певних файлів.

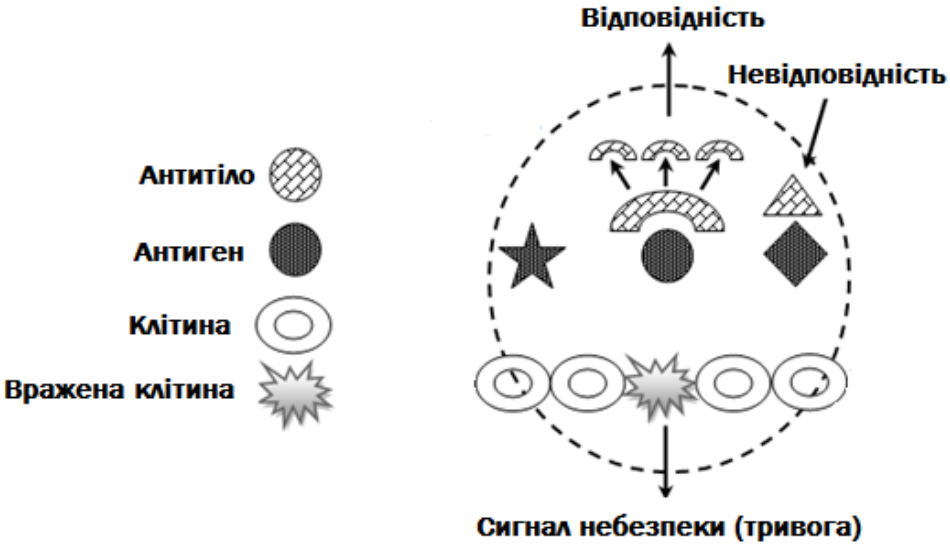


Рис. 2.6. Формування сигналу про небезпеку

В праці [115] використано стратегію захисту разом з алгоритмами теорії небезпеки (DTAL), яка враховує сигнали небезпеки зонально. Ця техніка використовується при створенні комп'ютерних ігор (наприклад, симуляторів гри в футбол), де ігрова зона поділена на дві частини – гравця і противника. Коли антиген (у футболі – м'яч) знаходиться на частині гравця (в зоні небезпеки), то активується алгоритм захисника, і антитіла швидше реагують на всі антигени. Коли приймається сигнал небезпеки разом з сигналом від атакованого антитіла, то лімфоцити починають роботу (будучи заздалегідь в зоні антитіла). Дана стратегія захисника має вищу ефективність на 40-50% залежно від налаштувань антитіл.

2.3.5. Дендритна модель

Ці алгоритми засновані на абстрактній моделі дендритної клітини. Основну роль дендритних клітин, що представляють антиген, описано в працях [116, 117], де ДК складаються з лейкоцитів, які присутні у всіх тканинах. Вони наділені неоднорідною гемопоетичною лінією і функціонують у різних тканинах. Всередині різних тканин ДК відокремлюються і дозрівають. Пізніше вони

переміщуються до вторинних лімфоїдних тканин для представлення антигенів Т-клітинам, щоб викликати імунну відповідь.

Незрілі ДК займають поверхню тіла, зазвичай присутні в незрілому стані та не стимулюють Т-клітини. Після того, як збудники будуть оброблені, вони мігрують до тимуса та селезінки, де незріла ДК дозріває і стимулює імунну відповідь. Як пояснюється в [118], переміщення між різними станами шляхом розпізнавання сигналів включають асоційовані з патогенами молекулярні зразки, сигнали небезпеки та апоптотичні сигнали (безпечні сигнали). Ці сигнали пояснюються наступним чином:

• перетворенням незрілих ДК в зрілі;

• сигнали небезпеки виділяються, коли пошкоджена клітина тканини;

• безпечні сигнали видаються при регуляції загибелі клітин.

Відповідь Т-клітини визначається відповідними концентраціями цих типів сигналів.

Напівзрілі ДК мають супресивний ефект, тоді як зрілі – акцентуючий.

Перший алгоритм дендритних клітин включав поєднання різних сигналів дослідження навколишнього середовища даних (антиген). Нечіткі поля, що виникають відповідно до концентрації костимуляторних молекул, є індикатором для ДК, щоб зупинити збір антигенів та перейти до віртуального лімфатичного вузла. ДК працюють на вхідних сигналах з набором наперед заданих параметрів, щоб видавати вихідні сигнали. Значення «+» призначається, якщо кумулятивний зрілий сигнал є більшим, ніж кумулятивний напівзрілий сигнал, і навпаки. Зрілий контекст представлення цього антигену розраховується відносно загальної кількості антигенів.

ДК розроблений як тканинний сервер в праці [119]. У цьому алгоритмі існує три етапи: ініціалізація, оновлення та агрегація. Ініціалізація стосується встановлення початкових значень, а стадія оновлення підрозділяється на оновлення тканин та клітинний цикл. Сервер тканини включає оновлення тканин і клітинний цикл.

В роботі [115] використано ДК на наборі даних KDD 99 [121] після того, як до оптимізації додавалися дві додаткові функції: мультиплікатор антигену та рухоме вікно. Антигенний мультиплікатор робить кілька копій антигену, щоб подолати проблему «дефіциту антигену». У кожній ітерації нові сигнали розраховуються за допомогою алгоритму рухомого часового вікна.

Р. Оатс [122] розробив підхід алгоритму ДК для проблеми класифікації роботів, де ДК розроблена як окремий фізіологічний модуль для сумісності з бібліотеками і інтерфейсом з двома додатковими модулями: обробки зображень та виконання ДК.

В дослідженні [119] використовується масив, щоб зберегти значення антигену та підраховувати кількість разів, коли ДК зібрав антиген. Існує три параметри в схемі налаштування ДК: маса для обробки сигналів, виведення значень постійного струму та кількості постійних струмів.

Робота [118] вказує на зв’язок ДК з архітектурою та операційними вимогами мереж датчиків. Виходячи з цього варіанту, було запропоновано комплекс ДК для виявлення атак на сенсорні мережі з набором функцій:

1. Сигнали з декількох джерел даних збираються в ДК. Нові вихідні цитокіни накопичуються на етапі дозрівання кожної ДК.
2. Зв’язування антигенів з контекстною інформацією здійснюється комплексом ДК.
3. Обсяг несанкціонованої поведінки вузла визначається комплексом ДК за допомогою згенерованих сигналів.

Алгоритм дендритних клітин є методом фільтрації даних для використання його в задачах виявлення аномалій. Системи розроблені на основі такого методу використовуються для виявлення аномалій в реальному часі.

2.4. Опис імунних операторів

Всі види ШІМ схожі через те, що оперують на спільній теоретичній базі, але в них є відмінності, котрі починаються після вибору глибини проектування. Глибина проектування залежить перш за все від кінцевого призначення

використованої ШІМ (при дослідженні швидкості протікання процесів в мережі достатньо буде і стандартного набору операцій). Наприклад у клональному методі і ШІМ на засадах теорії небезпеки подібна структура, але реалізуються вони по різному. Дендритні моделі також використовують антитіла, але імунні мережі використовують паратопи і епітопи, що значно поглиблює можливості опису розроблених інструментів. Також методи різняться шляхами отримання навчальних вібірок. Частіше за все відмінності починаються при обробці даних, що дозволяє нам використовувати дані методи на різній глибині проектування. Характерною властивістю клонального методу є те, що ми можемо проводити порівняння як антитіл, так і їх клонів, що дає можливість повністю контролювати якість вибірки, а непотрібні антитіла просто видаляти.

Коли антитіло втрачає необхідний рівень подібності, то воно замінюється клоном, що може обробити антиген. Тоді при обробці вибірок використовує пригнічення продукування антитіл з низькою подібністю. Таким чином цінність елементів мережі з низькими показниками відповідності виконанню поставленої задачі не залежатиме від їх типу. Кожен елемент у якого рівень подібності знаходиться на відстані, що перевершує задану границю – буде видалений і не братиме подальшу участь у роботі ШІМ.

Модель негативного/позитивного відбору чудово зарекомендувала себе при побудові систем захисту від вірусів і аномалій. Її відмінністю є принцип обробки вже готових клонів. При негативному відборі відбираються клони з вищою подібністю до антигенів.

Проведений опис дозволив дійти до висновку, що для вирішення задачі доцільно задіяти модель, котра зможе знаходити НД мережею скоординованих антитіл:

$$\mathrm{AIS} = (\mathrm{G}, \mathrm{A}, \mathrm{c}, \mathrm{S}) = [\mathrm{Int}(\mathrm{G}, \mathrm{A}) \rightarrow \mathrm{Rate}(\mathrm{A}) \rightarrow \mathrm{Ext}(\mathrm{A}) \rightarrow$$

$$\rightarrow \mathrm{Mut}(\mathrm{cl}, \mathrm{G}) \rightarrow \mathrm{Membest}(\mathrm{cl}) \rightarrow \mathrm{Del}(\mathrm{A}, \mathrm{cl}, \mathrm{c})] \rightarrow \mathrm{End}(\mathrm{S}),$$

де G –антигени;

A –антитіла;

c – межа близькості;

S – умова закінчення.

Приведена AIS виконується блоками:

Int(G,A) – порівняння антитіл A з антигенами G, через їх подібності (*F*):

$$F_{A-G} = (1 - d_{A-G})^{-1},$$

де d_{A-G} – відстань між *A* і *G*, що обчислюється:

$$\| \quad d_{(A-G)} \| = \sqrt{\sum_{j=1}^{F} (A - G)^2}.$$

Rate(A) –відбір *A* з найвищою подібністю;

Ext(A) – розширення популяції відібраних *A* шляхом клонування їх зразків з найвищою подібністю;

Mut(G, cl) – мутація клонів cl та G ;

Membest(Cl) – утворення пам'яті клонів з найвищою подібністю;

Del(A, cl, c) – видалення A і cl в яких подібність вища межі близькості c;

End(S) – кінець циклу при досягненні умови закінчення S.

Для виявлення типів несанкціонованих дій використаємо клональний відбір та властивості класифікації дій дендритного методу.

Визначаються основні інструкції і оператори опису моделей ШІМ для виявлення НД. Саме оператори дозволять виконувати відбір, розширення, клонування, видалення та мутацію вибірок.

Всі ці оператори мають власні налаштування, котрі необхідно підстроювати під конкретні ситуацію. Правильне налаштування операторів зменшить витрати на виконання інструкцій.

Запорукою швидкості роботи відбору потрібних антитіл буде вирахування динаміки імунних процесів, що розраховується за формулою [123]:

$$\frac{dA_i}{dt} = \left\{ \alpha \sum_{J=1}^{N} F_{ji}\, A_j(t) - \alpha \sum_{K=1}^{N} F_{ik}\, A_k(t) + \beta G_i - k_i) \right\} F_i(\mathrm{t}),$$

де F – подібність антитіла типу (i, j);

N – кількість A;

$\alpha \sum_{J=1}^{N} F_{ji}\, A_j(t)$ – стимуляція утворення A;

$\alpha \sum_{K=1}^{N} F_{ik}\, A_k(t)$ – видалення A;

$(\beta G_i - k_i)$ – стимуляція G;

$F_i(\mathrm{t})$ - видалення A.

Дане рівняння дає можливість прорахувати швидкість відбору необхідного антитіла для побудованої ШІМ.

Основним параметром клонування являється величина вибірки антитіл n і їх кратність N_c.

Для дослідження впливу на швидкість виконання імунного алгоритму (ІА) кількості антитіл у вибірці та їх кратності клонування N_c необхідно буде використати функцію статичного клонування [106, 120]. Він може клонувати кожне з n антитіл наперед задану кількість разів (тобто задану кількість поколінь). Тепер стає можливим визначення для антитіла максимального рівня поколінь клонування, знаючи їх кратність. На рис. 2.7 зображена залежність поколінь для роботи ІА від вибірки антитіл (на 12-ти запусках).

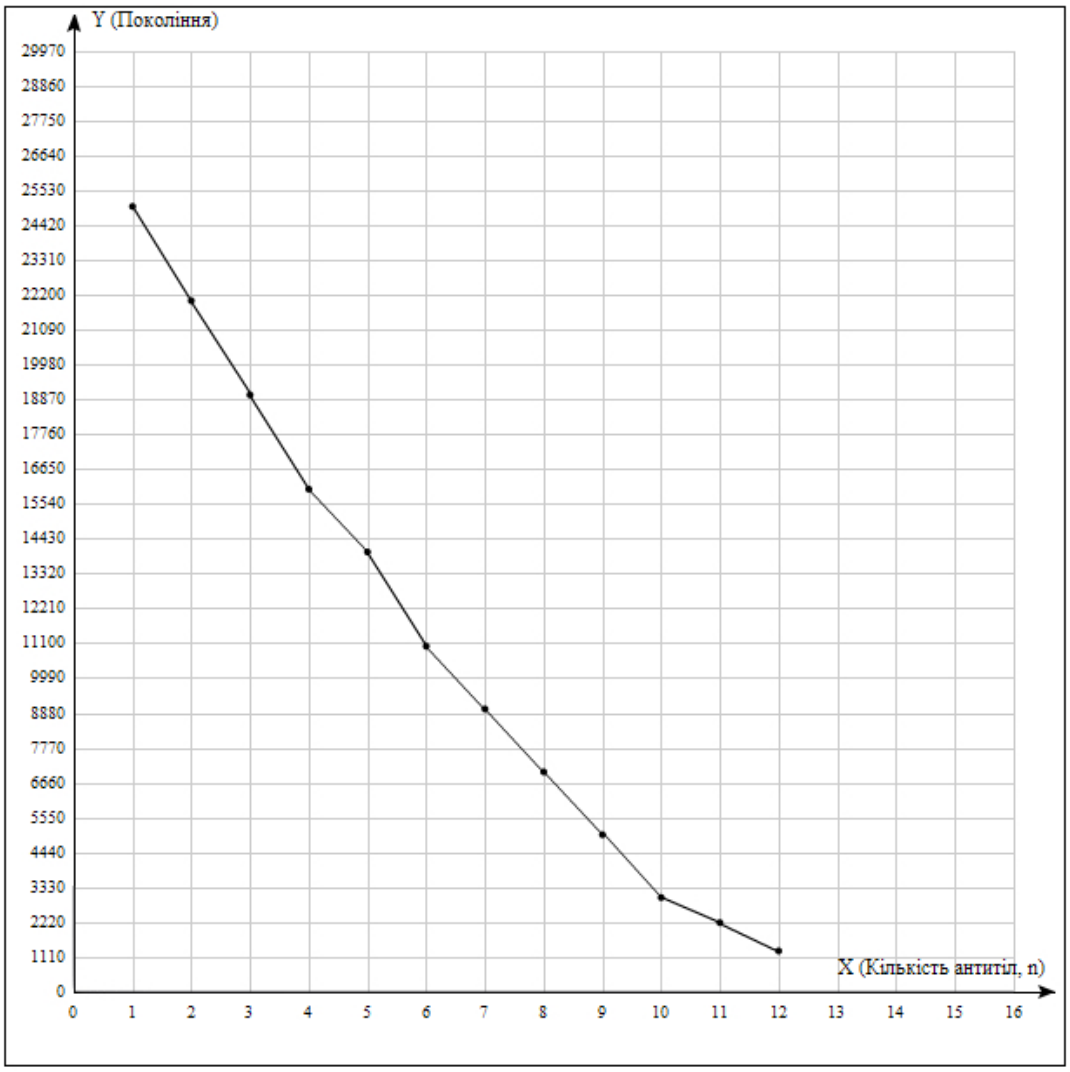


Рис. 2.7. Залежність кількості поколінь від кількості від вибірки антитіл

На діаграмі видно: чим більша вибірка антитіл, тим менше потрібно пройти поколінь для завершення роботи ІА. Логічно припустити, що аналогічно до вибірки антитіл, при зменшенні кратності клонування, зменшиться і необхідність в проходженні зайвих поколінь для досягнення результату ІА. Тобто при збільшенні кількості членів вибірки антитіл ми зменшимо кількість необхідних поколінь, але це так само збільшить витрати ІА на обрахунки подібностей антитіл і клонів.

Тому для зменшення часу на роботу ІА доцільно вибірку антитіл зробити фіксованою, а кратність регулювати за допомогою формули [123]:

$$N_c(A) = \begin{cases} N_{c\_min} \text{ при } F(A) \leq F_{best} * 0.35; \\ N_{c\_max} \text{ при } F(A) \geq F_{best} * 0.65; \\ F(A) / \dfrac{F(A) - F_{best} * 0.25}{F_{best} * 0.4}, \end{cases}$$

де, $\frac{F(A)-F_{best}*0.25}{F_{best}*0.4}$ - середня залежність між подібностями антитіл (для проміжних значень меж кратності);

N_{c_min} і N_{c_max} – межі кратності;

$F(A)$ – подібність антитіла A;

F_{best} – найкраща подібність всього покоління.

Значення межі подібності вираховується рівнянням:

$$F(A) = \frac{\sum_{i=0}^{nl} F(l_i)}{nl},$$

де, l_i – подібність і-го антитіла, а nl – загальна кількість антитіл у вибірці.

Значення 0.35 і 0.65 мінімізують похибку в розрахунках через кореляцію подібності *Nc*. За допомогою формалізації умов роботи кратності буде підвищена швидкість проходження формування вибірок антитіл. Значення 0.35 та 0.65 (на відміну від використаних в роботі [123]) отримано з рівняння обчислення клонів антигену [129]:

$$N_{max} = \sum_{i=1}^{n} round\ (N - D_{i,q}, N),$$

де, N –кількість антитіл;

round – оператор заокруглення результату з дужок до десятинного;

$D_{i,q} = \|A_i - G_q\|$ – відстань від вибраного антитіла *i* до антигена *q* (*i* =1,…, *N*)

Вимоги мутації у ІА:

–визначення схильних до мутації антитіла;

– крок мутації.

При збільшенні цих параметрів буде підтримуватись різноманітність антитіл. Переважно такі зміни будуть надходити від антитіл з низькою подібністю [86]. При зниженні вище зазначених вимог зявиться можливість дослідити область

навколо антитіла і знайти антитіла з найвищою подібністю (а це знизить швидкодію). Для того, щоб не вибирати між точністю і швидкодією, потрібно буде коригувати параметри ІА. Тобто для антитіл з підвищеною мутацією подібності потрібно знижувати значення коефіцієнтів мутації в ІА.

Досягнути такого регулювання мутації можна лише задавши границі мутації⌊P_{mmin}; P_{mmax} ⌋.

Для кожної наступної мутації ймовірність буде залежати від подібності антитіла. Залежність вираховується за формулою:

$$P_m(A) = P_{mmax} * \frac{(F(A) - F_{best})}{(F_{worst} - F_{best})} + \frac{(F_{best} - F(A))}{(F_{worst} - F_{best})} * P_{mmin},$$

де, $P_m(A)$ – мутація антитіла (А);

$F(A)$ – відображає подібність A;

F_{best} – відображає кращу подібності;

F_{worst} – відображає гіршу подібності;

P_{mmin} і P_{mmax} – відображають граничні значення мутації А.

Для знаходження параметрів мутації $P_m(A)$ потрібно задіяти вектор ймовірностей:

$$P(\mathrm{A}_N) = (p_1, p_2 \ \ldots p_N),$$

де p_N – ймовірність мутації N-ї характеристики A ;

N – параметри A.

p_N впливає на ті параметри, котрі йому потрібні для обробки антигена.

Мутація цих параметрів можлива через додавання змінної з нульовим математичним очікуванням:

$$A_{N+1} = A_N + R(0, \sigma_N),$$

де σ_N –відхилення, а R – випадкове значення асоційованого параметра антитіла.

За допомогою кроку мутації відхилення легко отримати амплітуду мутації. Для цього σ_N зазнає мутації, після чого параметри антитіл зміняться. σ_N, в свою чергу, міняється за формулою:

$$\sigma_{N+1} = \sigma_N \frac{F_{best} - F(A_N)}{F_{best} - F_{worst}},$$

де всі F відносяться до коефіцієнтів мутації A;

Саме при роботі імунного алгоритму буде регулюватись як величина, так і можливість мутації параметрів A, що гарантуватиме при максимальній швидкості обробки даних високу точністю. Тобто при підвищенні значення подібності клону після мутації, антитіло буде заміщатися покращеним клоном. Дані дії допоможуть зберігати в вибірках лише антитіла з високою подібністю. Збереження антитіл з низькою подібністю може позитивно вплинути на різноманітність вибірок, але значно погіршить параметри всієї популяції, що однозначно буде мати негативні наслідки. Тому краще буде залишати в вибірках лише антитіла з високою подібністю, що і було теоретично реалізовано.

2.5. Виявлення несанкціонованих дій

Розпізнавання НД реалізовується через ШІМ, де антитіло A відповідатиме за виявлення вторгнення у формі антигену G. За допомогою такої організації стане можливим виявляти НД.

Всі процеси протікатимуть у межах $S^{N \times L}$, де налаштування точки S може корегуватися шляхом налаштування характеристик L. Тобто вся множина властивостей, що характеризуватиме A і G, буде описана L-вимірним вектором, а точка S описуватиме взаємодії типу $(G - A)$ і $(A - G)$.

Коефіцієнти i, q, j, r антитіл, антигенів і подібностей (F), будуть знаходитися в межах: $i = 1, \dots, N$; $q = 1, \dots, Q$; $j = 1, \dots, N$; $r = 1, \dots, R$. Ці межі дозволять пришвидшити рішення задачі, обмеживши спектр виконуваних операцій.

ЕА може знаходитись в режимі навчання чи розпізнавання. Навчання налаштовується нижче вказаними кроками:

1. Введені дані досліджуються під час емуляції для запису їх протоколів.

2. Протоколи піддаються порівнянню для виділення особливостей їх роботи.

3. Готові шаблони поведінки передаються в бібліотеку ознак. Також вираховуються рейтинги появи фрагментів, котрі зберігають дані про всі об’єкти, що містять знайдену інформацію. Складається список з рейтингами ознак НД для обробки в ШІМ.

4. Дослідження не НД. В протоколах роботи не НД відстежуються спільні для не НД фрагменти і формуються списки з рейтингами ознак. Вони будуть задіяні при проведення навчання ШІМ на негативні вердикти (неправильні спрацювання).

5. Використовуючи набори рейтингів появи даних фрагментів будується і навчається ШІМ. Всі дані поділяються на відомі НД і не НД або нові НД.

Розпізнавання налаштовується нижче вказаними кроками:

1. Обробка в емуляторі.

2. Порівняння отриманих даних з НД.

3. Надання інформації про відношення об’єкта до НД.

ШІМ зводиться до виду [73, 86, 129]:

$$AIS = [A, A_{\{v\}}, A_{\{m\}}, v, N, G, Q, L, F, MX, Rate, Ext, N_c, Mut, Del, C, C^*, H, R, Z, M_q, M_q^*, B, \sigma_d, \sigma_s],$$

де A, $A_{\{m\}}$, $A_{\{v\}}$ – вибірка, пам’ять та кількість нових антитіл (($A_{\{v\}} \in S^{v \times L}, v \leq \mathrm{N}$););

N – кількість A;

G – початкова вибірка антигенів;

Q – сумарна кількість G, що використовуються;

L – задіяні ознаки G або A;

F – матриця подібностей $A_{\{i\}}$ до G_q з параметрами $f_{i,q}$;

MX – матриця подібностей антитіл ($A_i - A_j$);

$Rate$ – процес відбору;

Ext –розширення вибірки шляхом клонування;

N_c – кратність клонування;

Mut – процес мутації елементів;

Del –видалення елементів;

C – популяція A із N_c клонів, що згенерована з A($C \in S^{Nc \times L}$);

C^* – очікувана вибірка A із C з необхідним рівнем подібності;

H – матриця, в якій містяться подібності вирахувані між c_r^* (взятих з C^*) і кожним антигеном G_q з елементами $h_{r,q}$;

R – рейтинг появи;

Z –зрілі A;

M_q –пам'ять клонованих тіл G_q (після відбору);

M_q^*– кінцеве значення M_q ;

B – матриця подібностей M_q^* з $b_{i,q}$;

σ_d –умова видалення ($\sigma_d = 0.4$);

σ_s– умова стиснення ($\sigma_s = 0.25$);

τ – тригер кінця роботи.

Правильний відклик гарантується тим, що A конкурують за розпізнавання антигенів. Всі створені дублікати A при видаленні зникають і мережа може стискатися (стиснення забезпечує значення σ_s). Кожна пара $A-G$ може контактувати в рамках заданого кордону $S^{N\times L}$ лише при збігу значень подібностей $f_{i,q}$, що відповідають за їх зв'язки між собою. Саме ці зв'язки і будуть гарантією коректного виявлення вторгнення. Значення подібності $A_{i,j}$ вказує на схожість значень двох різних матриць МХ, з подібностями антитіл $(A_i - A_j)$.

Навчання ШІМ починається з того, що для всіх $G_q(G_q \in G)$ вираховується подібність $f_{i,q}$ до кожного A_i :

$$f_{i,q} = 1/(1 + (A_i - G_q)),$$

З A, котрі мають необхідну подібність, формується підмножина $A_{\{n\}}$. Далі антитіла копіюються (інструкцією Ext) відповідно да їхньої антигенної подібності $f_{i,q}$, формуючи вибірку клонів C. Параметри N_c встановлюються в ІА відповідним рівнем подібності (для фіксованого набору A).

Такі клони складуть вибірку C^*, де всі антитіла c_r^* підлягають процесу мутації з індивідуальною інтенсивністю α_r, котра буде вираховуватись на основі подібності. Значення подібності впливатиме на швидкість мутації і залежатиме від перебору необхідних значень подібності A).

Далі вираховуються подібності $\mathrm{H}_{r,q} = 1/(1 + (c_r^* - G_q))$ елементів з множини C^* стосовно антигенів G_q, при $r = 1, \dots, N_c;\ q = 1, \dots, Q$. Потім з C^* відбирається Z A з найвищим значенням $\mathrm{H}_{r,q}$ і вони надходять до M_q . При цьому з M_q витискаються елементи у яких $\mathrm{H}_{r,q} < \sigma_d$.

Для отриманих клонів M_q повторно вираховується $\mathrm{H}_{r,p} = 1/(1 + (c_r^* - c_p^*))$, де $\mathrm{r,p} = 1, \dots, N_c$, і відбувається витиснення клонів для яких виконується $\mathrm{H}_{r,q} < \sigma_s$.

В свою чергу антитіла $A_{\{m\}}$ об’єднується з M_q^*, де для G_q: $A_{\{m\}} \leftarrow \left(A_{\{m\}}. M_q^*\right)$.

Вираховуються подібності $\mathrm{H}_{i,q} = 1/(1 + (A_{\{m\}}^{\,i} - A_{\{m\}}^{\,q}))$ антитіл $A_{\{m\}}$ і видаляються ті антитіла, для яких $b_{i,q} > \sigma_s$. Стають відомими антитіла: $A \leftarrow \left\{A_{\{m\}}, A_{\{v\}}\right\}$ і ШІМ закінчує роботу.

ШІМ являтиме множину $A_{\{m\}}$ з матрицею B. $A_{\{m\}}$ характеризуватиме внутрішній стан G до потрапляння в мережу. Матриця B характеризуватиме зв'язки A, що дасть змогу зобразити структуру ШІМ. Опис матриці подібності B дає можливість доступу до структури ШІМ і визначати належність A до певних класів. Значення подібностей $A_{\{m\}}$ до G дає необхідну інформацію для визначення антитіл що розпізнають антигени вторгнень.

При виявленні вторгнень на вхід ШІМ подаються нові G, для яких розраховуються подібності з $A_{\{m\}}$. Після вирахування подібності визначається клас антигену і приймається рішення про наявність чи відсутність НД.

Найбільш енергозатратним кроком буде визначення подібності між усіма A, але через використання елементів кратності затрати часу для обробки даної дії будуть мінімальними.

2.6. Висновки розділу 2

1. Представлено модель ЕА несанкціонованих дій, котра за допомогою поведінкового аналізу даних може виявляти нові вторгнення в комп’ютерній мережі, без оновлення сигнатур. До моделі додано розширення котрі мінімізують помилкові спрацювання без зниження швидкодії системи.

2. При виявленні НД використано ті ознаки, які найкраще характеризують події пов'язані з роботою вторгнень.

3. Проаналізовано сучасні математичні моделі ШІМ. В ході аналізу найбільш придатною системою, котра відповідатиме поставленим перед нею вимогам виявилась модель штучної імунної мережі, функціонал якої дає змогу виявляти вторгнення за допомогою взаємодії антитіл з різними ступенями подібності.

4. Розглянуто основні імунні оператори, що можливо використати для опису моделі виявлення НД. Введено додаткові параметри до описуваних моделей, котрі дають змогу пришвидшити роботу і підвищити загальну продуктивність.

5. Всі етапи моделювання, навчання, виявлення і розпізнавання виконано на базі ШІМ. Навчання мережі проводиться на основі отриманих рейтингів появи ознак, після чого дані ознаки відносяться до НД чи дій користувача (або до раніше не відомих вторгнень).

РОЗДІЛ 3
ДІАГНОСТИКА

3.1. Вибір інструментальної бази для реалізації діагностування НД і опис основних операторів

Deterlab - це віртуальна установка підтримки експериментів, яка характеризується тим, що з її допомогою можна запускати будь-які, в тому числі широко поширені в даний час, програми маршрутизації [124]. Deterlab емулює специфічні умови реальних мереж, дозволяючи формувати необхідну топологію і схеми маршрутизації. В процесі експерименту можна імітувати події, характерні для реальних мереж (наприклад, обрив лінії зв'язку або її перевантаження, атаки і спроби злому вузлів) та аналізувати їх наслідки. Також, при необхідності можна забезпечити проходження через віртуальну мережу трафіку між реальними взаємодіючими вузлами.

3.1.1. Дослідження операторів

Мета даного розділу полягає в наданні необхідної теоретичної бази для використання приведених концепцій і теорій, які можуть комбінуватися з сучасними напрацюваннями для підвищення ефективності виявлення вторгнень в комп'ютерній мережі. Розглянуто основні алгоритми виявлення зміни, бо вони будуть використовуватися при діагностуванні для контролю часових фрагментів і формування даних для визначення симптомів і сигнатур вторгнень. Теорія Демпстер-Шафера, розроблена Гленном Шафером і Артуром Демпстером, описана в роботах [95, 96]. Теорія опрацьовувалась і розвивалась в роботах [93, 94, 96, 97, 125]. ТДШ також може бути використана в системах виявлення вторгнень для структурування отриманих в ході постійного моніторингу й аналізу функцій системи даних [136, 145]. ТДШ може об'єднати окремі частини доказів для отримання кінцевого результату і встановлення висновку про наявність чи відсутність вторгнень (діагноз). ТДШ має всі властивості, котрі можуть допомогти виявляти вторгнення і несанкціоновані дії в комп'ютерній мережі.

Для вирішення задачі розпізнавання НД методом діагностування пропонується, основуючись на роботі операторів ТДШ, використати наступну модель виявлення вторгнень (рис 3.1):

Рис. 3.1. Модель діагностування несанкціонованих дій

На основі спостережень і зібраних доказів будуються симптоми, котрі після процесу відбору дають можливість відносити певні типи активностей в мережі до вторгнень чи звичайних дій системи.

Узагальнено процес діагностування зведений до наступного виду:

$$DRT = (Obs, Symp, Sel) = [Present\,(Obs) \rightarrow Check(Symp) \rightarrow Sel(Symp) \rightarrow Get\,(SymptA, SymptP)] \rightarrow Diagn.$$

Де, Obs – спостережувані;

$Symp$ – симптоми;

Sel – критерій відбору.

При роботі виконуються наступні оператори:

Present (Obs) – оператор представлення спостережуваних (O);

Check (Symp) – перевірка спостережуваних на наявність в них симптомів, які будуть служити для діагностики стану системи;

Sel (Symp) – відбір симптомів з набору спостережуваних;

Get (SymptA, SymptP) – після операції відбору отримуємо на виході інформацію про наявність симптому (SymptP) чи його відсутність (SymptA);

Diagn - винесення діагнозу на основі сукупності отриманих симптомів з набору спостережуваних.

3.1.2. Визначення структури проникливості

В ТДШ всі можливі стани, в яких може бути процес, можна описати за допомогою структури проникливості.

Структура проникливості визначається масивом $SRT = \{SRT_1, SRT_2, \dots, SRTi\}$, де для $1 \le i \le N$, SRT_i визначає конкретний стан системи. ТДШ застосовується при визначенні «стану процесу», бо він дає змогу встановити діагноз і на його основі зробити висновок про наявність аномалій у системі.

3.1.3. Основне переконання

На основі спостережень і зібраних доказів головною метою ТДШ є визначення ймовірності того, наскільки даний стан SRT_i є фактичним станом процесу [95, 96]. Діагноз формується на основі доказів ТДШ. Коли маються докази надання інформації про поточний стан процесу, то він повинен або підтримувати групування можливих станів, або не допускати їх, якщо не досягнуто певного ступеня впевненості в цих доказах (якщо вони не переконливі). ТДШ формує докази після досягнення рівня впевненості в них, і такий результат називається основним переконанням (ОП). Формально ОП є функцією f_{srt}, де SRT являє собою пов'язану структуру проникнення. f_{srt} відображає будь-яку підмножину структури проникнення для реального значення, $f_{srt}: P(SRT) \rightarrow R$. Значення області являє собою набір підтримуваних станів, а відповідне значення діапазону представляє силу цієї підтримки. Значення в діапазоні позначається як маса переконання (або просто маса). Всі маси переконання обмежені значеннями 0 і 1 включно, $\forall x \in P(STR), 0 \le f srt(\mathrm{x}) \le 1$. Маса переконання 0 являє собою повну відсутність підтримки відповідного набору станів (тобто не містить фактичного стану). Маса переконання 1 означає що поточний стан входить до даного набору. ОП ділить рівно 1 одиницю маси переконання на всі можливі набори станів. Розділення досягається рівнянням $\sum_{\mathrm{x} \in P(srt)} f_{srt}(x) = 1$. Маса переконання ніколи не повинна розподілятися на порожні стани $f_{srt}(\emptyset) = 0$. Отже, виходить, що ОП - це сукупність всіх можливих «претензій», які можуть виникнути в процесі в межах конкретної структури проникнення. Будь-який набір станів, якому присвоюється

ненульова маса, свідчить, що процес може бути в одному з цих станів. Присвоєння нульової маси означає, що процес відсутній.

Можна назначити масу переконання проти набору станів *A*, через структуру проникливості *SRT*. Це робиться шляхом побудови нового набору станів *B*, що містить всі елементи в структурі проникливості не в *A*,*B* = *SRT/A*, і присвоєнні їй маси переконання. Маса переконання в порожніх множинах дорівнює нулю. Це відбувається тому, що маса в порожньому наборі доказує, що жоден стан зі структури проникливості не описує поточний стан процесу, а це суперечить тому, що наша структура проникливості охоплює всі можливі стани процесу, в яких може бути система. В ОП маса може бути приписана до його структури проникливості. Даний набір має особливе значення, бо будь-яка присвоєна до нього маса не знає нічого про справжній стан процесу (він забезпечує рівну підтримку для всіх станів, не надаючи корисної інформації про них). ОП зі всією своєю масою, призначеною до структури розрізнення, буде повністю пустою.

3.1.4. Правдоподібність переконання

Використовуючи описані вище конструкції, можна перейти до застосування ТДШ для визначення правдоподібності переконання. Нижня межа суб'єктивної ймовірності (переконання) обчислюється за формулою [125]:

$$bl(f_{srt}, A) = \sum_{\mathrm{B}\subseteq\mathrm{A}} f_{srt}(\mathrm{B}),$$

де *SRT* - структура проникливості, *fsrt* - являється ОП, та *A* буде набором станів. Формула підсумовує, що маса переконання міститься безпосередньо у всіх підмножинах наборів опитуваних і звичайних станів.

Верхня межа суб'єктивної ймовірності (правдоподібність) обчислюється формулою:

$$pl(f_{srt}, A) = \sum_{\mathrm{B}\cap\mathrm{A}\neq\emptyset} f_{srt}(\mathrm{B}),$$

де змінні мають такі ж значення, як і в станах переконання. Формула підсумовує, що маси переконання будь-яких станів мають принаймні один спільний з опитуваним стан.

Правдоподібність і переконання пов'язані через рівняння $pl(f_{srt}, A) = 1 - bl(f_{srt}, SRT/A)$, яке відображає альтернативну інтерпретацію правдоподібності, де правдоподібність в A являється залишками впевненості в A, після того, як до A була додана правдоподібність. Виходить, що немає можливості для відхилення поточного стану від структури проникливості, і що ця структура гарантовано буде містити даний стан.

3.1.5. Оператори злиття

Оператори злиття в ТДШ можуть скомпонувати разом кілька доказів з ОП для формування кінцевих ОП. Оператор злиття - це будь-яка функція, яка приймає на вході дві ОП, а в якості вихідного сигналу видає одну (з умовою, що всі ОП знаходяться в одній структурі проникливості).

Оператори злиття в ТДШ описуються рівняннями [125]:

$$n(SRT, A_{srt}, B_{srt}) = \sum_{x,y \in P(SRT) | x \cap y = 0} A_{srt}(x) B_{srt}(y);$$

$$C_{srt}(\emptyset) = dfo(SRT, A_s, B_s, \emptyset) = 0;$$

$$C_{srt}(z) = dfo(SRT, A_{srt}, B_{srt}, z) = \frac{1}{1 - n(S, A_{srt}, B_{srt})} * \sum_{x,y \in P(SRT) | x \cap y = z} A_{srt}(x) B_{srt}(y).$$

Вхідні ОП (A_{srt}, B_{srt}) і вихідні (C_{srt}) знаходяться в одній структурі проникливості SRT. Допоміжна функція n (коли задані два ОП однієї структури проникливості) обчислює загальну кількість мас в суперечливих одна одній частинах доказів (коли множини переходів не мають загальних станів) A_{srt}, і B_{srt}. Для конкретної множини станів $z \in P(SRT)$, за допомогою змінної Демпстера *dfo* обчислюємр загальну масу, яка підтримує z і додає масу з будь-якої множини станів A_{srt}, і B_{srt}, що поділяють z як загальний стан. Слід зазначити, що повністю суперечливі дані не будуть мати загального стану, тому така маса не буде враховуватися (бо в разі $z = \emptyset$, де ці значення будуть враховуватися, $C_{srt}(\emptyset) = dfo(SRT, A_{srt}, B_{srt}, \emptyset) = 0$ маса «зникає»). Компенсується це тим, що маси в

результаті ОП C_{srt}, вирівнюється множенням на $\frac{1}{1-n(SRT,A_{srt},B_{srt})}$. Виходить, що правило Демпстера працює в наборах станів з не конфліктними доказами.

Кумулятивні оператори злиття опрацьовані в роботі [128] обчислюються за допомогою рівнянь:

$$C_{srt}(s) = dfo(SRT, A_{srt}, B_{srt}, SRT)\frac{A_{srt}(SRT)B_{srt}(SRT)}{A_{srt}(SRT)+B_{srt}(SRT) - A_{srt}(SRT)B_{srt}(SRT)};$$

$$C_{srt}(z) = dfo(SRT, A_{srt}, B_{srt}, z)\frac{A_{srt}(z)B_{srt}(SRT) + A_{srt}(SRT)B_{srt}(Sz)}{A_{srt}(SRT)+B_{srt}(SRT) - A_{srt}(SRT)B_{srt}(SRT)}.$$

Обмеження $A_{srt}(SRT) + B_{srt}(SRT) > 0$, потрібне для виконання приведеного вище рівняння. Воно дозволяє з'єднати кілька частин доказів, що приведе до утворення одного рішення по кільком різним подіям. Якщо спостережувані не надійні або упереджені, то таке злиття може допомогти об'єднати їх твердження в одне більш надійне. Оператори кумулятивного злиття також збільшують впевненість ОП в результатах.

3.1.6. Застосування фрагментів часу

Фрагменти часу - це послідовність значень, записаних з фіксованими інтервалами часу. Типом цього значення може бути те, що можна записати або виміряти. У роботі будуть розглянуті лише цілі типи. Тип значення не буде змінюватися між вимірами конкретного часового ряду. Фрагменти можуть знаходитися в будь-якій частині до тих пір, поки вона залишається незмінною протягом усього часового ряду. Також час між двома послідовними значеннями є фіксованою величиною. Будь-яка послідовність значень може розглядатися в якості фрагментів часу, використовуючи індекси значень як його час. Фрагменти часу завжди містять кінцеве число значень, хоча фрагменти можуть бути додані для нескінченної кількості часу.

Визначимо T часовим фрагментом з відповідним типом значення. Нехай послідовність з n значень в межах T індексується цілим числом. Перше значення має індекс 0, а кінцеве значення має індекс $\mathrm{T}_n - 1$, де T_n являє собою загальне число значень, що містяться в часовому ряді T. Значення фрагменту часу T з

індексом i, де $0 \leq i$ $\mathrm{T}_n<$, визначається як $\mathrm{T}_n(i)$. Час виникнення значення визначається $\mathrm{T}_t(i)$, використовуючи ті ж обмеження на i. $T(i)$ визначається як універсальна (по всім фрагментам часу) константа NM, коли значення в момент часу i не виміряне. Зверніть увагу, що якщо $T(i) = NM$, то $\mathrm{T}_t(i)$ містить час вимірювання. Нехай *Tp* відноситься до постійної тривалості часу між будь-якими двома послідовними значеннями в *T*, тобто $\forall_i|0 < i < \mathrm{T}_n, \mathrm{T}_p = \mathrm{T}_t(i) - \mathrm{T}_t(i-1)$.

Перетворення часових рядів застосовується на фрагменти часу, що призводить до появи нового екземпляру часових рядів. Визначимо *rel* як функцію вибору і вироблення часових фрагментів. Застосування *rel* до часового ряду *T* може бути описане як «відносна серія *T*». Функція rel приймає вихідні часові фрагменти T і віднімає їх від константних фрагментів d так, що *T(0) – d = 0*, або просто *d = T(0)*. Значення нових фрагментів відносно *rel(T) = T′* обчислюється як *T′(i)* = T(i) - d, $T_t'(i)$ = $\mathrm{T}_t(i)$ для всіх індексів *i*, де *0 ≤i*<T_n. Оригінальні й вираховані часові фрагменти мають однакову кількість значень. Ця операція змушує бути відносними до початкових значень нові фрагменти. Визначимо ще одну операцію *der*, з *T* часових фрагментів, бути «похідною від T». Значення der(T)=*T′*, визначаються як *T′(i) = T(i) – T(i-1)*, $T_t'(i) = T_t\ (i)$, для всіх індексів *i* ,котрі *0 < i* $\mathrm{T}_n<$. Змінені граничні умови призводять до появи нових часових фрагментів, що містять на один елемент менше $T_n' = \mathrm{T}_n - 1$, який вимагає присутності хоча б двох значень. Ця операція перетворює вихідні фрагменти, які інтерпретуються як послідовність абсолютних значень, до його відповідної послідовності значень дельти, забезпечуючи наближення похідної.

3.1.7. Методика виявлення змін

Важливо визначати коли, поведінка процесу відхиляється від попередньо визначеної специфікації (тобто вона зазнала «змін»). Виявлення змін є більш загальною задачею, ніж виявлення вторгнень, оскільки виявлення відхилення в процесі не обов'язково означає вторгнення [96, 97].

Позначимо алгоритм виявлення змін функцією *C*, для подання отриманих за допомогою моніторингу значень. Функція працює так само як і перетворення

часових рядів, приймаючи часовий ряд *T* в якості вхідних даних, і *C(T) = T′* в якості вихідного сигналу. Базовий тип часових фрагментів не обмежується, проте C повинено бути двійковим. Дозволені тільки значення {0, 1, NM}. Введення і виведення часових фрагментів повинно містити рівну кількість значень $\mathrm{T}_n = T_n^{'}$, так само інтервали часу $\mathrm{T}_p = T_p^{'}$, а також однаковий час виникнення для всіх відповідних індексів, $\forall_i | 0 \leq i < \mathrm{T}_n, \mathrm{T}_t(i) = T_t^{'}(i)$. При індексі *i* з часових рядів *T* вихід C, *C(T)(i) = T′(i)*, слід оцінити в 0, якщо ніяких змін не виявлено. *T′(i)* слід оцінювати в 1, якщо зміну виявлено і процес відображає поведінку, що відхиляється від певної моделі (тобто сигнал активний).

NM може бути повернутий, якщо алгоритм виявлення змін не може прийти до висновку через невиміряні значення в фрагментах часу введення.

Найчастіше використовуються порогові алгоритми. Пороги - один з найпростіших алгоритмів виявлення змін. Алгоритм порогового відхилення найкраще застосовувати в тих випадках, коли основні часові фрагменти мають девіантну поведінку, яка набагато перевищує нормальні значення (наприклад, мають місце перевищення активності в десятки й сотні разів). При роботі з НД відхилення будуть мінімальними, оскільки при роботі вони повинні бути непоміченими системою та оператором, тому використання таких алгоритмів малоефективне [93].

Альтернативою пороговим відхиленням є алгоритм csm (загальна сума), запропонований Е.С. Пейджем [126]. CSM - складніший ніж пороговий алгоритм, бо він не застосовується до кожного значення індивідуально. Це дозволяє своєчасно і точно виявляти навіть невеликі зміни в процесах.

Алгоритм виявлення зміни CSM працює наступним чином. Нехай *csm(T, ul, μ, σ, a)* функція виявлення зміни CSM, де T буде функцією вводу часових фрагментів, ul буде верхнім/нижнім селектором CSM, μ буде середнім відхиленням, σ буде стандартним відхиленням і a буде порогом тривоги. Тип введення часових фрагментів T повинен мати строгий повний порядок, а також підтримку додавання і віднімання операцій. ul - це двійкове значення *ul* ∈ {0, 1}, що представляє верхній селектор при ul = 1, або нижній при *ul* = 0 в CSM

алгоритмі. μ і σ повинні мати той же тип, що і часові фрагменти і являють собою «нормальне» середнє і стандартне відхилення основного процесу. *a* - поріг сигналізації, що також розділяє тип часових фрагментів. Для даного індексу *i|0 ≤ i* < T_n часових рядів *T, T(i),* значення CSM за індексом *i* задається наступним чином [127]:

$$S_i = \begin{cases} \max(0, S_{i-1} + \frac{T(i) - \mu}{\sigma}) & ul = 1 \\ \min(0, S_{i-1} + \frac{T(i) - \mu}{\sigma}) & ul = 0 \end{cases},$$

$$csm(T, ul, \mu, \sigma, a)(i) = \begin{cases} 1 & S_i \geq a \wedge ul = 1 \\ 0 & S_i < a \wedge ul = 1 \\ 1 & S_i \leq -a \wedge ul = 0 \\ 0 & S_i > -a \wedge ul = 0 \end{cases}'$$

$$csm(T, ul, \mu, \sigma, a)_t(i) = T_t(i).$$

Змінна *Si* представляє «значення» в момент часу $T_t(i)$. CSM значення *Si* є сумою до індексу *i*, різниці між послідовними значеннями часових рядів *T(i),* і їх очікуваного значення μ, $T(i) - \mu$. Якщо припустити, що за умови середнього значення μ буде фактичним очікуваним значенням основного процесу тимчасових рядів, то потім значення CSM *Si* повинні залишатися наближеними до нуля. В іншому випадку значення *Si* буде збільшуватися, якщо середнє значення для процесу буде більшим за даних умов (ul= 1), а *Si* буде зменшуватись, коли ul=0. З цього випливає, що значення параметра ul визначає, коли алгоритм відстежує аномальну поведінку в обох напрямках. Фактичний вихід алгоритму виявлення зміни визначається в кожен момент часу $T_t(i)$, шляхом порівняння поточного значення CSM, *Si* з порогом тривоги a. Якщо поріг перевищено, $|Si| \geq a$, то повертається 1, і це значить, що процес проявляє девіантну поведінку. В іншому випадку повертається 0, і процес показує нормальну поведінку. При визначенні різниці між значеннями часових рядів і очікуваних значень $T(i) - \mu$ - це стандартизована величина, що в якості умови використовує значення відхилення σ, $\frac{T(i)-\mu}{\sigma}$ Це дозволяє визначити поріг тривоги (a) та значення CSM |*Si*| для інтерпретації всього числа стандартних відхилень від очікуваного значення

процесу, використовуючи всього лише одне значення з порогу тривоги a, для викликів функції CSM з різними значеннями μ та σ.

3.2. Використання алгоритму затримки

Функція затримки дієва тоді, коли відхилення відбувається за дуже короткий час. Коли виникає аварійний сигнал, то спрацювання сигналізації повинно бути затримане, щоб встигнути правильно синхронізувати інші алгоритми виявлення змін. Таке розширення додає функцію «Невстановлене значення покарання» (UVP), котра може доповнювати алгоритм CSM. Такий алгоритм працює лише з цілими даними (на відміну від часових фрагментів, які за вхідні дані можуть брати «невимірні» значення NM). Кожне значення *Si* в CSM обчислюється відповідною формулою [127]:

$$S_i = \begin{cases} \text{UVP} & \text{T(i)} = \text{NM} \\ \max(0, S_{i-1} + \frac{\text{T(i)} - \mu}{\sigma}) & \text{T(i)} \neq \text{NM} \wedge \text{ul} = 1 \\ \min(0, S_{i-1} + \frac{\text{T(i)} - \mu}{\sigma}) & \text{T(i)} \neq \text{NM} \wedge \text{ul} = 0 \end{cases}.$$

UVP матиме однаковий тип значень з часовими фрагментами T і буде вимірюватися в стандартних відхиленнях.

При реалізації цього розширення CSM приймає значення $S_i^{'}$, обчислене даним алгоритмом, а для обчислення значення модифікованого CSM $S_i^{'}$застосовується функція:

$$S_i^{'} \begin{cases} \max(a, S_i) & \text{ul} = 0 \\ \min(a, S_i) & \text{ul} = 1 \end{cases}.$$

Змінене значення CSM в $S_i^{'}$ потім використовують як вхідні дані для наступного етапу алгоритму, в якому застосовується сигнал порогового значення.

Також в CSM використовується розширення «чутливості» [127], яке дозволяє ігнорувати незначні відхилення від очікуваної поведінки часових фрагментів. Це забезпечує безперервне наближення значення CSM до нуля. Розширення

чутливості додає новий параметр до визначення CSM – параметр К (чутливість), який також має однаковий з часовими фрагментами тип і формує нове визначення CSM (T, ul, μ, σ, a, k). Модифікована функція CSM містить зміну визначення Si, яка приймає вигляд:

$$S_i = \begin{cases} \max(0, S_{i-1} + \dfrac{T(i) - \mu}{\sigma} - k) & ul = 1 \\ \min(0, S_{i-1} + \dfrac{T(i) - \mu}{\sigma} + k) & ul = 0 \end{cases}.$$

К завжди вважається додатним числом, тим самим змушуючи CSM прямувати до нуля. Щоб збільшити значення CSM і перевищити поріг спрацьовування тривоги a, значення часових фрагментів повинні перевершувати (по значенню) чутливість *k*. Збільшення *k* зменшить загальну кількість виявлених вторгнень. Зменшення *k* підвищить загальну кількість вторгнень.

3.3. Організація моделі діагностування

Для побудови моделі необхідно почати з основних визначень, на яких буде базуватися подальша робота. Нехай *S* буде системою, що захищена методом діагностування. Система *S* є вузлом мережі (комп'ютером).

Система (*S*) в будь-який момент часу може бути в одному зі своїх станів. Кожен стан системи визначається унікальним поєднанням значень його даних (від мережі, користувача, датчиків тощо) і пам'яті. Оскільки існує обмежена (хоча і досить велика) кількість можливих комбінацій вмісту даних і пам'яті, то кількість станів системи кінечна. Нехай $S = \{S_0, S_1, \dots, S_{S_{n-1}}\}$ - це кількість (*Sn*) станів системи *S*, в яких вона може бути. Значення змінної S є абстрактним визначенням системи й буде використовуватись не в математичних контекстах. Тобто, коли мова йде про *S*, то це відноситься до всього, що відбувається в системі. Можна спрощено описати будь-який стан в *S* як $s \in S$ (в ТДШ *Si*, $0 \le i < S_n$). Оскільки стан системи може змінюватися щодо часу t, то визначимо поточний стан *S* в момент часу t, як *S(t)* ($S(t) = s \in S$). Тип одиниць вимірювання t немає значення, бо

кількість станів системи *S* кінечна. Також необхідною умовою буде те, що *S* не змінюватиме стану частіше заздалегідь заданої частоти.

3.3.1. Дерево специфікацій

Стани в S можна організовувати в групи за допомогою алгоритмів і схем групування. Будується «Дерево специфікацій», котре допоможе організовувати стани чи процеси в системі на різній глибині деталізації. Таким чином, буде легше виділяти певні характеристики (специфікації) деяких станів для їх зберігання в пам'яті чи організації роботи.

Дерево специфікацій T складається з асоційованого дерева спрямування T_g = *(V, E)* (яке буде згадуватися як дерево специфікацій певного компонента дерева), а також вершин стану функцій відображення T_v, з їх обмеженнями. Починаючи з асоційованого дерева *T*, T_g, *V* буде множиною *Vn* вершин, V = $\{V_0, V_1, \dots, V_{V_{n-1}}\}$, та *E* позначатиме множину *En* ребер *E* = $\{E_0, E_1, \dots, E_{E_{n-1}}\}$.

Для конкретного краюe_i, $0 \le i < E_n$, нехай $e_{i,s} \in V$ являтиме собою «джерело» вершини e_i та $e_{i,s} \in V$, представлятиме вершину «призначення». Всі ребра *E* спрямовані з вихідної вершини в вершини призначення. Оскільки *Tg* є деревом, то буде одна вершина без вхідних ребер, а всі інші вершини будуть мати рівно одне вхідне ребро. Нехай вершина $v_0 \in V$ є вершиною без вхідних ребер і буде називатися «кореневою» вершиною. Корінь матиме тільки вихідні ребра і буде вершиною, що спрямована до будь-якої іншої вершини в *Tg*.

Визначимо «глибину» будь-якої вершини, $v_i \in V$, де $0 \le i < V_n$, як число ребер в унікальному (виходячи з того, що *Tg* є деревом) спрямованому шляху від кореневої вершини v_0до v (відстанню між v_0 і v). Глибина вершини v*i* може бути позначена змінною v*i,d* .

Стани системи *S* пов'язані з вершинами дерева специфічності *T*, T_g, через вершину стану функції T_v, що відображає вершини *V* з T_g до підмножин *S*, T_v *: V* → *P(S)*. Стани, пов'язані з вершиною $v \in V$, можуть перебувати у зв'язаному стані набору. Функція відображення T_v повинна відповідати наступним критеріям:

• Коренева вершина дерева специфічності, T_g , v_0, повинна бути пов'язана з усіма станами в S, T_v $(v_0) = S$.

• Для всіх ребер в T_g, $e_i \in E$, де $0 \leq i < En$, вершина призначення пов'язана з $e_i, e_{i,d}$, повинна мати пов'язані з ними стани відображення в T_v, $T_v(e_{i,d})$ і містити підмножину станів з вихідної вершини $e_{i,s}, T_v(e_{i,s})$, тобто повинна виконуватись відповідність: $T_v(e_{i,d}) \subset T_v(e_{i,s})$.

• Всі вершини в Tg, які мають однакову глибину, $\mathrm{d} \in \mathrm{N}, \forall_{v_i} \in V \mid 0 \leq i < V_n \wedge v_{i,d} = d$, повинні взаємно виключати пов'язані набори станів. А саме, $\forall_{v_i}, v_j \in V$, де $v_{i,d} = v_{j,d}$ отримуємо, що $Tv(v_i) \cap Tv(v_j) = \emptyset$.

• Всі вершини Tg повинні мати принаймні один стан в їх наборі станів $\forall_v \in VT_v(v) \neq \emptyset$.

Оскільки глибина вершин збільшується, то розмір вершин пов'язаних наборами станів будуть меншими за розміром, $\forall_{e_i} \in E$, де $0 \leq i < E_i$ маємо $|T_v(e_{i,d})| < |T_v(e_{i,s})|$, а це означає, що класифікація набуває «конкретності». Асоційовані стани з вершини на глибині $d+1$ не потрібні для обліку всіх пов'язаних з ними станів вершини в шарах d. Таким чином, для кожного конкретного стану S, $s \in S$ на глибині C, де $C \leq d$, буде існувати рівно один пов'язаний набір станів вершини $v \in V$, де існує кожний стан для s, $\exists v \in V \mid s \in T_v(v)$. Тобто матимемо унікальний шлях вершин, що походить від кореня v_0, і закінчується в вершині глибини d, що має відповідний набір станів і містить конкретний стан s. Всі пов'язані набори станів вершин глибини, що більші за d, не містять конкретний стан s, тобто, $\forall_{v_i} \in V$, $де\ 0 \leq i < Vn$ та $v_{i,d} > d$, і ми маємо $s \notin T_v(v_i)$.

3.3.2. Дерево діагностики

Основою діагностування буде дерево специфікацій, що використовується для представлення всіх діагнозів у яких може перебувати система S. Таку систему можна назвати деревом діагностування. Її структуру можна побачити на рис. 3.2.

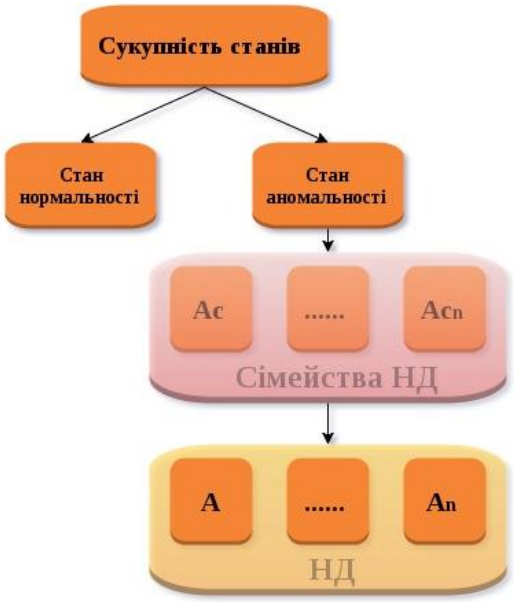


Рис. 3.2. Структура дерева діагностики

Починаючи з верхньої частини дерева, кореневий вузол містить всі можливі стани, які можуть бути в *S*.

У наступному слої стани пов'язуються з однією з двох вершин (нормальності або аномальності). Всі стани з вершини нормальності відображають нормальну поведінку. Нормальна поведінка визначається очікуваннями адміністратора системи *S*. Всі стани, які не описуються нормальною поведінкою, будуть пов'язані з вершиною аномалій.

Вершини *An* в кінцевому шарі $A = \{a_0, a_1, \dots, a_{A_{n-1}}\}$ пов'язані з конкретною атакою проти системи *S*. Атаки являються використаннями вразливостей в *S*. Використання певного входу або набору входів, що ведуть від нормального до аномального стану, будуть вразливістю. Кожна вершина *A*, $a \in A$, що пов'язана з конкретною атакою, має пов'язаний з нею набір станів $T_v(a)$, який містить всі аномальні стани, котрі доступні після виконання цієї атаки.

Вершини між аномаліями і атаками відповідають за класифікацію атак AC_n і представлені масивом AC = $\{ac_0, ac_1, \dots, ac_{AC_{n-1}}\}$, де ac∈AC представляє яку-небудь конкретну вершину. Кожна вершина в AC являє собою конкретну вказану користувачем групу атак, що відноситься до підмножини вершин *A*. Прикладом

несанкціонованого доступу в An можуть бути атаки відмови в обслуговуванні. Будь-яка атака a∈A, що вважається відмовою в обслуговуванні, буде мати свою вершину в класифікаторі атак AC. Дії пов'язані з кожною вершиною атак мають унікальні характеристики.

Говорячи про систему S в час t, і дерево діагностики DT, яке представляє простір станів S, функція $DT_s(S,t)$ повертає найглибшу вершину v від DT_g, пов'язані стани якої містять поточний стан S, S(t)∈ $DT_v(v)$. Оскільки кожна вершина v в DT_g набуває особливого значення, виходячи з набору станів дерева діагностики, то повернене значення $DT_s(S,t)$ буде діагнозом для S в момент часу t.

Першим кроком моделювання дерева діагностики в ТДШ буде пов'язання всіх станів системи з конкретним ідентифікатором. Тобто формується набір станів SSn з простору ідентифікаторів $SS = \{ns, as_0, as_1, \dots, as_{A_{n-1}}\}$ (SS означає «простір станів»). Ідентифікатор простору станів $ns \in SS$ відповідає всім станам, пов'язаним з вершиною нормальності дерева діагностики (ns означає «нормальний стан»). Ідентифікатори простору станів $\{as_0, as_1, \dots, as_{A_{n-1}}\} \in SS$ (as означає «стан атак»), відповідають станам відповідних вершин атаки $A = \{a_0, a_1, \dots, a_{A_{n-1}}\}$. Утворюється простір станів SS, що являє собою структуру проникнення в ТДШ. Щоб застосувати ТДШ на будь-якій вершині v дерева діагностики, використовується підмножина ss структури проникнення $ss \subseteq SS$, де ss містить весь простір ідентифікаторів станів, чиї відповідні вершини є віхами v (якщо v включається в цей набір). Щоб привести ТДШ в форму дерева діагностики DT, необхідно DT віднести до структури проникнення SRT описаним вище способом (отримавши DT_{srt}).

За рахунок введення нових елементів і зв'язків досягається підвищення достовірності виявлення атак і мінімізація можливих помилкових визначень вторгнень у мережі.

Поставлена задача вирішується завдяки інспектуванню стану системи на виникнення аномальної чи неправомірної активності, використовуючи дерево

діагностування (*DT*), що дає змогу відстежити активність системи користувача. Діагностування системи проводиться у реальному часі і дозволяє вчасно реагувати на порушення. Як вихідні характеристики для діагностування беруться дані про роботу системи (*S*) на рівних проміжках часу (*t*). Дані використовуються в *DT* для генерації звітів про відмінність від еталонних значень в *S* та для формування повідомлень про несанкціоновані дії або підозрілу активність. При нормальній роботі система продовжує функціонувати в звичайному режимі.

Суть ілюструється схемою (рис.3.3):

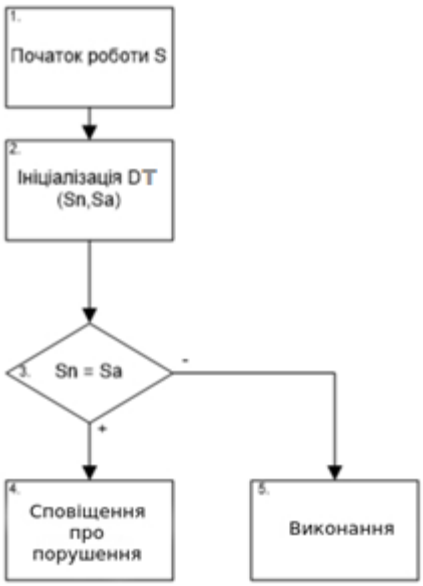


Рис. 3.3. Схема роботи

1. Початок роботи системи S.

2. Створення дерева діагностування *DT* та формування масивів з основними станами *Sn* та *Sa*. *Sn* зберігає стани нормального функціонування системи. *Sa* містить сигнатури несанкціонованих дій і атак.

3. Порівнюються стани в сітці діагностування на виявлення відхилень від нормального стану активності системи в сторону відхилень.

4. При виявленні сигнатур, котрі відповідають відхиленням (*Sa*), *DT* сповіщує про порушення.

5. При виявленні сигнатур, котрі відповідають станам нормального функціонування системи (*Sn*), *DT* не перешкоджає роботі системи.

3.3.3. Можливість спостереження

Основною вимогою до контрольованої системи *S* при діагностуванні є можливість спостереження за нею. Для *S* всі стани *S, S ∈ Sn*, повинні піддаватись спостереженню. Для спостереження за системою потрібно мати можливість отримати зразки значень вхідних даних і пам'яті S в будь-яку мить часу *t, S(t) = s*. Можливість спостереження *S* є необхідною при діагностуванні, оскільки це дозволяє робити висновки про поточний стан *S* навіть коли невідомо *S(t).*

Коли система знаходиться в множині станів s ⊆ S, то зразки поведінки можуть бути записані й створено профіль поведінки. Можна знати множину станів системи, якщо вона працює в контрольованому середовищі, яке задовольняє умови роботи всієї множини цих станів. Наприклад, щоб створити профіль поведінки для всіх станів *S* при зчитуванні з диска *s ⊆ S,* де *S* може працювати в контрольованому середовищі, яке може оперувати інформацією на диску з різними способами й методами. Неможливо працювати відразу з усіма методами зчитування інформації, бо це призведе до перевантаження пам'яті та сповільнення або зависання системи.

Під час виконання, коли *S* фактично використовується для виконання своїх функцій, неможливо знати достовірний стан *S(t),* оскільки середовище *S* не контролюється. Однак, якщо профілі поведінки були побудовані для різних наборів станів *S*, а потім, після значень вибірок з поточного стану *S(t),* можна порівняти поточну поведінку з відомими зразками поведінки системи, то таким чином можна виявити стан, в якому знаходиться *S(t).*

При діагностуванні профілі поведінки будуються тільки для відповідних множин станів вершин дерева діагностики DT_g. Цей набір станів являється підмножиною пов'язаних множин вершин станів, які є віхами в DT_g, і називається DT_{srt}. Слід зазначити, що ця змінна також відноситься до структури проникливості для дерева діагностики *DT*. Профілі поведінки будуються для станів DT_{srt}. Стан інших вершин DT_g може бути апроксимований у вигляді об'єднання вершин пов'язаних з просторами цих станів. Поки відомі профілі

поведінки для станів DT_{srt}, інші профілі поведінки можна оцінити, навіть якщо вони безпосередньо не спостерігаються.

3.3.4. Задача діагностування

Мета діагностування - визначити повністю спостережувану систему S і дерево діагностики DT, яке класифікує стани S, щоб визначити, чи є ефективним діагноз S за допомогою DT, $DT_s(S,t)$ в будь-який момент часу t. Вище було описано засоби, за допомогою котрих можна досягнути поставленої задачі.

Отримання діагнозу S, $DT_s(S,t)$ буде рішенням поставленої задачі. Якщо $DT_s(S,t)$ повертає значення, відмінне від всієї сукупності вершини DT_g, то відомо, що система поводиться нормально, тобто $DT_s(S,t)$ повернула вершину нормальності. Якщо система поводиться аномально, то $DT_s(S,t)$ повернула вершину аномалії. Діагностування працює як класичний детектор аномалій, інформуючи коли S перестає працювати в звичайному режимі. Причина аномального стану може бути пов'язана з помилками чи несанкціонованою діяльністю в системі. Якщо трапилася відома раніше атака (тобто вона міститься в наборі діагнозів дерева атак A), то $DT_s(S,t)$ повертає відповідний елемент A, $DT_s(S,t) \in$ A. Це забезпечує можливість точного виявлення не лише атаки, а й її типу. Якщо стався напад, який був того ж класу, як клас атаки в $ac \in$ AC, то $DT_s(S,t) =$ ac буде повернений в якості діагнозу. Ця функція має потенціал виявлення нових атак, які поводяться так само, як й ті, що містяться в певному класі ac.

3.3.5. Процедура обчислення діагнозу

Метою діагностування є обчислення діагнозу S в момент часу t, $DT_s(S,t)$. Візуальне уявлення потоку операцій під час діагностики приведена на рис. 3.4.

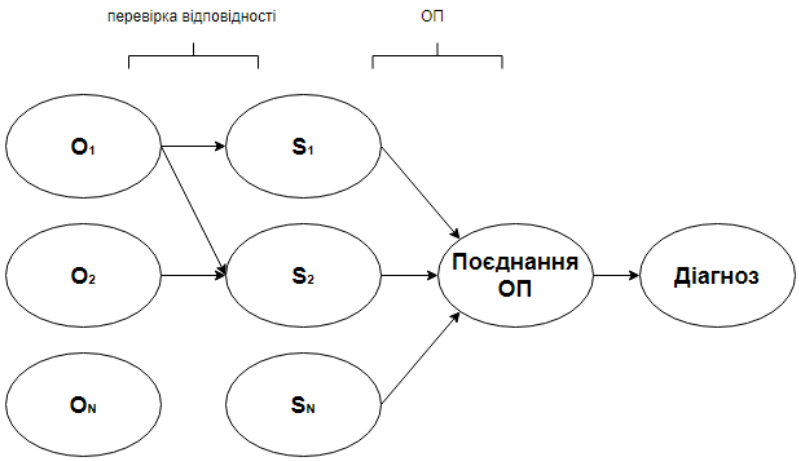


Рис. 3.4. Схема обчислення діагнозу

3.3.6. Методика відбору спостережуваних

S повинна бути відкритою для спостереження, а змінна в поточному стані *S*, *S(t)* повинна бути доступною для записування. Нехай спостережувані мають можливість відношення до часових фрагментів шляхом запису значення цього параметра стану в кожен момент часу *t*. Необхідно зібрати й визначити набір спостережуваних, які будуть служити для діагностики стану *S*. Визначимо цей набір спостережуваних O_n як $O = \{o_0, o_1, \dots, o_{O_{n-1}}\}$ і $o \in O$, або $o_i \in O$, де $0 \leq i < O_n$, що буде спостережуваною. Оскільки *S* може бути в будь-якому просторі станів DT_{srt}, то необхідна можливість перевірки актуальності кожної спостережуваної. Якщо спостережувана не є актуальною, вона не сприятиме діагностиці стану *S* і не буде включена в набір спостережуваних *O*.

3.3.7. Процедура побудови симптомів

Симптом приймає значення спостережуваних в момент часу t, і видає на виході ОП в порівнянні з DT_{srt}, що забезпечує можливість діагностики *S*. Симптоми отримують нові дані про спостережуваних як тільки вони стають доступними, а за визначенням часових фрагментів вони доступні через одинакові проміжки часу. Симптоми підтримують і приймають рішення на основі внутрішніх станів, таким чином, одинакові значення можуть подаватися в симптом кілька разів, а це може привести до різних ОП на виході. Нехай набір ST_n симптомів подається як $ST = \{st_0, st_1, \dots, st_{ST_{n-1}}\}$, де $st \in ST$, або $st_i \in ST$, де $0 \leq$

$i < ST_n$, відноситься до будь-якого конкретного симптому. Спостережуваний, що пов'язаний з кожним симптомом, визначається функцією STO, яка приймає симптом і повертає відповідне значення спостереження на виході. Значення симптому st в момент часу *t*, визначається через *st(STO(st)(t)),* припускаючи, що *st* раніше міг вже використовуватись. Таким же чином визначаються послідовні значення базової спостережуваної *STO(st)* для t', що являється передумовою до $\forall t' < t$, *STO(st)(* t'*)* (оскільки симптом *st* підтримує внутрішній стан). Виходить, що *st(t)* відноситься до значення симптомів *st* в момент часу *t*.

Клас симптому визначимо за допомогою бінарних симптомів. Бінарне визначення симптому починається з визначення поточної спостережуваної величини *o(t),* яка подається в бінарний симптом *BS*. Це значення подається безпосередньо в алгоритм виявлення зміни BS_{cd} (cd - change detection). Алгоритм виявлення змін приймає дійсні значення в якості вхідних даних і видає значення BS_{cd}:R→{0,1}. Спостережувані значення можуть мати не вимірювані значення *o(t) = NM*, і повинні бути опрацьовані алгоритмом виявлення до наведеного вище виду. Алгоритм виявлення зміни є компонентом симптому, який може містити внутрішній стан. Якщо алгоритм виявлення зміни симптомів не має внутрішнього стану, то і симптом не матиме його. Якщо результат алгоритму виявлення змін $BS_{cd}(o(t))$ буде коректним, то попередньо обчислене ОП, BS_p буде повернене як ознака симптому. В іншому випадку буде повернене попередньо обчислене ОП, BS_a. Обидва ОП будуть належати до однієї структури проникнення BS_f, таким чином, до цих змінних можна посилатись як $BS_{PBS_f} = BS_p$ та $BS_{aBS_f} = BS_a$. Якщо BS_{cd} істинні, то це говорить про наявність симптому, якщо ні - симптом відсутній.

Щоб побудувати двійковий симптом, необхідно вибрати пов'язані з ним спостережувані й визначити характеристику його часових фрагментів. Алгоритм рішення визначає, чи будуть ці характеристики присутніми або відсутніми. Значення, отримані за допомогою цього алгоритму, повинні змінюватися кожного разу, коли S знаходиться в кожному зі станів DT_{srt} .

Цей сценарій можна продемонструвати наступним прикладом. Маємо двійковий симптом BS зі структури проникнення $BS_f = DT_{str} = \{n, a_0, a_1\}$. Алгоритм вирішення BS_{cd} завжди має значення станів DT_{srt}, в яких може бути система *S*. Під час роботи системи визначається, в яких станах *S* може бути в DT_{srt}. Кожний простір станів DT_{srt}, котрий раніше впливав на алгоритм рішення, виробляє в ньому лише достовірні вихідні дані. Таким чином, якщо алгоритм рішення дає позитивні дані, то маємо дію котра міститься в просторі станів. Якщо алгоритм видає негативні дані, то це значить, що в нас немає ніякої інформації, з якою можна працювати, і алгоритм видає один і той же висновок. Також можливий інший сценарій. Якщо BS_{cd} реагує на позитивні стани, коли поточний стан S знаходиться в просторі станів n∈BS_f. Під час виконання, якщо BS_{cd} реагує на позитивні стани, то можна стверджувати, що нинішній стан S знаходиться в n, та якщо BS_{cd} набуває негативного значення, то можна сказати, що S знаходиться в a0, a1 ∈ BS_f. Незалежно від результату алгоритму рішення, ступінь інформації, що ми отримали в просторі станів BS_f, знаходиться в поточному стані *S(t)*. Алгоритм виявлення змін CSM [127] використовується при побудові симптомів.

Логічний висновок алгоритму рішення BS_{cd} використовується в BS_p або BS_a при подальшій роботі. Обидва ці твердження приймають форму ОП і мають різні методи побудови. BS_p і BS_a повинні мати два записи з ненульовими масами. BSp призначає відповідну масу в масиві станів DT_{srt}, що спрямовує BS_{cd} до позитивного значення. BS_{cd} повинен містити всі стани в DT_{srt}, що спрямовує BS_{cd} до невірного (негативного) значення. Якщо BS_{cd} не може продукувати істинні або хибні значення при впливі станів f ∈ DT_{srt}, то F видаляється з BS_p і BS_a, усуваючи можливість симптому впливати на f. Інша частина маси BS_p і BSa повинна бути призначена до структури проникнення. Якщо x - побудований стан симптому в наборі для BS_p або BS_a, то він являє собою призначену для користувача систему мас в діапазоні 0 <y <1, і b_{BS_f} і є симптомом ОП, а b_{BS_f}матиме вигляд b_{BS_f} (x) = y і b_{BS_f} (BS_f) = 1−y. Чим більше значення y, тим

вище ймовірність отримання симптому на виході. Значення у двійковому симптомі *BS* буде позначатись як *BSy*.

3.3.8. Робота з симптомами

При оцінці масиву симптомів ST в момент часу *t* проводиться набір основ переконань еквівалентного розміру [125]. Для постановки діагнозу стану *S* потрібно мати цілісну ОП. Необхідно поєднати набори ОП для формування єдиної ОП, котра буде відображати їхню суть. Поєднання ОП проводиться операторами злиття ТДШ.

Якщо всі симптоми приймаються як незалежні один від одного (тобто лежать в основі спостереження), то кожна змінна відстежується окремо одна від одної та об'єднується незалежними операторами злиття. Якщо симптоми залежать один від одного, то один оператора злиття може об'єднувати всі ОП. Якщо групи симптомів будуть залежати один від одного, то необхідно буде використовувати два оператори злиття. Виходить, що один залежний оператор злиття буде об'єднувати всі залежні групи ОП в одну незалежну. Це дасть змогу сформувати набір незалежних основ переконань, що будуть еквівалентні за розміром до числа залежних груп. Надалі незалежні ОП утворять кінцеву ОП. Приклад такого процесу зображений на рис. 3.5.

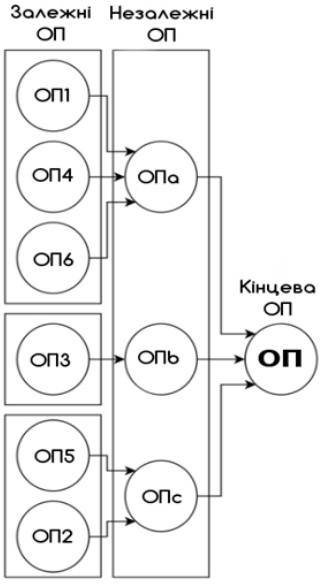


Рис. 3.5. Об’єднання симптомів

3.3.9. Аналіз кінцевого діагнозу

Після того, як всі ОП, що випромінюються від симптомів в *S* були об'єднані за допомогою операторів злиття, залишається одна ймовірність $B_{DT_{srt}}$. Ця ОП являє собою дані, отримані кожним із симптомів в *S*. Ці симптоми випромінюють дані про стани в DT_{srt} системи *S*, котра знаходилась в моменті *t*. Стани простору, використані в якості основи структуру проникнення, представлені в DT_{srt} та відповідають пов'язаним станам вершин в дереві діагностики DT_g. Використовуючи простори станів структури проникнення DT_{srt}, можна оцінити стан простору будь-якої вершини V в DT_g шляхом формування об'єднаного простору станів всіх пов'язаних станів вершин. Виходить, що$B_{DT_{srt}}$ побудовано над структурою проникнення DT_{srt} і містить комбіноване рішення просторів станів контрольованої системи *S*, за допомогою якого можна обчислити ймовірність того, що поточний стан *S* існує в просторі станів, пов'язаних вершинами v в DT_v (*v*). Стає можливою оцінка поточного стану системи простору *S* в будь-який момент часу *t* DT_s *(S,t).*

Тепер можна описати спосіб інтерпретації $B_{DT_{srt}}$ для прийняття рішення про величину DT_s (*S,t*). Спочатку порівнюють суб'єктивні ймовірності нормальності та аномальні вершини. Відповідний масив станів нормальності вершини відповідає масиву $\{n\} \subset DT_{srt}$ структури проникнення дерева діагностики *DT*. Пов'язані масиви станів аномалій вершин відповідають масиву $A \subset DT_{srt}$, де A = $\{a_0, a_1, \dots, a_{A_{n-1}}\}$, тобто масиву всіх станів атак. Переконання і правдоподібність вершин нормальності рівняється *bl*($B_{DT_{srt}}$, {*n*}) та *pl*($B_{DT_{srt}}$,*{n}),* а значення цих же вершин для аномолії дорівнює bl($B_{DT_{srt}}$, *A*) та *pl*($B_{DT_{srt}}$, *A*). Якщо імовірність аномалії вища нормальності, то система перебуває в аномальному стані й виконується наступний крок. Якщо імовірність нормальності вища - система перебуває в нормальному стані й процес діагностування завершується.

На наступному етапі, по можливості, визначається тип атаки. Відповідні стани для кожного з класів атак вершин AC = $\{ac_0, ac_1, \dots, ac_{AC_{n-1}}\}$ обчислюються в ТДШ, ∀ ac ∈ AC ac ⊂ DT_{srt}. Обчислюються зв'язки переконання й

правдоподібності, а кожна атака ac ∈ AC пов'язується з відповідними $bl(B_{DT_{srt}}, ac)$ та $pl(B_{DT_{srt}}, ac)$.

Всі суб'єктивно імовірнісні оцінки для вершин класу атаки потім порівнюються з «критеріями рішення», щоб визначити правильність рішення.

Таким критерієм може бути поріг t від значень достовірності. Тільки вершини класу атаки зі значеннями правдоподібності вище цього порогу, $pl(B_{DT_{srt}}, ac) \geq t$, будуть розглянуті в процесі відбору. Якщо жоден клас вершин атаки не залишається після застосування даних критеріїв, то система перебуває в аномальному стані, і процес діагностики завершується. Якщо виявляється хоча б один клас атаки, то вибирається атака з найбільшою ймовірністю появи, і система переходить до наступного кроку процесу діагностики.

На заключному етапі визначається, чи знаходиться S в стані конкретної атаки a з масиву пов'язаних класів атак $a \in ac$. Кожна атака буде містити елемент з масиву атак $\{a\} \subset$ A. Переконання і правдоподібність обчислюється для кожної атаки: $bl(B_{DT_{srt}}, \{a\})$, $pl(B_{DT_{srt}}, \{a\})$, $\forall\{a\} \in$ ac. Якщо жодна з вершин атак не виявляється, то система знаходиться в стані невідомої атаки і процес діагностики закінчується.

Якщо виявлена вершина атаки, то вибирається атака з найбільшою суб'єктивною ймовірністю, і система вважається під загрозою цієї конкретної атаки. Після цього процес діагностики закінчується.

Якщо в процесі роботи виникне необхідність використання двох пар ймовірності і правдоподібності a_b, a_p , b_b, b_p (якщо дані будуть мати максимально наближені значення), то необхідно буде їх порівняти й вибрати максимально ефективні дані:

• Порівнюючи верхню межу правдоподібності кожної ймовірності одна з одною $a_p > b_p$.

• Порівнюючи нижню межу кожної правдоподібності одна з одною $a_b > b_b$.

• Порівнюючи очікувані ймовірності одна з одною (очікувана правдоподібність слідуватиме симетричному розподілу ймовірностей) (a_b + a_p)/2 > (b_b+ b_p)/2.

• Порівнюючи точність суб'єктивних ймовірностей одна з одною $a_p - a_b > b_p - b_b$.

3.3.10. Перевірка діагностування

При наявності набору повністю бінарних симптомів ST для кожного симптому st∈ ST необхідно призначити значення достовірності st_y. Обидві ОП з *st* приймають вид st_aта st_p, кожен з яких матиме два незаповнених вхідних значення в ОП (одне з яких буде структурою проникливості). Нехай BST позначатиме «масив істинних вірувань» і буде BST*($st_p \wedge st_p$) (BST)>0, тобто буде одним фокальним елементом, а не структурою проникнення st_p. Нехай BSF позначатиме «масив хибних вірувань» і буде визначатись BSF $\subset st_a \wedge st_a$ (BSF) > 0, тобто одним координаційним елементом, а не структурою проникнення st_a. Значення достовірності st_y можна налаштувати так, щоб регулювати кількість мас, що виділяються на будь-який з координаційних елементів випромінюваної ОП. Цей принцип застосуємо до двох окремих значень достовірності, котрі регулюють достовірність в st_aта в st_p. Нехай st_y застосовується для налаштування ОП st_p, тоді st_p (BST) = st_y та st_p (st_f) = 1 − st_y. Нехай st_xрегулює ОП st_a, тоді st_a (BSF) = st_xта st_a (st_f) = 1 − st_x. Тепер потрібно визначити значення, котрі необхідно застосовувати для встановлення довіри в st_x і st_yдля комбінації симптомів і процесу аналізу, щоб діагностувати поточний стан системи *S*. Перш за все *S* має бути в одному з просторів станів структури проникнення дерева діагностики DT_{srt}. Позначимо "поточний стан" *S* як *CS*. Виявляється, що коригування значень довіри stx і sty не буде впливати на результати діагнозу *CS* через те, що:

1. Всі симптоми - бінарні.

2. Кожен симптом на виході має масив симптомів BST або BSF, який містить поточний стан S, CS. Кожен симптом правильно передбачає CS з масиву симптомів, що містить CS. Слід зазначити, що за визначенням бінарного симптому CS міститися в BST, BSF або BST і BSF таким чином, симптом завжди здатний зробити правильний прогноз.

3. Кожний стан в структурі проникнення DT_{srt} повинен мати унікальну сигнатуру або набір сигнатур. Позначимо сигнатуру як масив ST, де кожен елемент являє собою логічне значення, що повертається з базового алгоритму виявлення симптому st_{cd}. Впорядкування логічних елементів не має значення до тих пір, поки він залишається послідовним при порівнянні елементів один з одним. Сигнатури конкретного стану простору z∈DT_{srt} позначимо як SIG(z). Стан простору являє собою множину будь-яких елементів, які могли утворитися коли z був у поточному стані CS і всі симптоми в ST правильно передбачили CS. Якщо z з'являється лише в стані BST або BSF, то для кожного симптому ST, SIG (z) матиме одну сигнатуру. Якщо z з'являється в обох станах BST та BSF для n різних симптомів, то можна побачити, що SIG(z) буде містити 2^n різних сигнатур. Головна вимога полягає в тому, що всі стани в DT_{srt} повинні мати унікальний набір сигнатур, тобто $\forall a$, b ∈ DT_{srt} , a ≠ b, SIG(a) ∩ SIG(b) = ∅. Якщо DT_{srt} знаходиться в сценарії з двома станами a та b, які поділяють ту ж сигнатуру SIG(a) ∩ SIG(b) ≠∅, то це виправляється додаванням нових симптомів BST і BSF, з можливістю вибору між ними.

4. Значення оцінки правдоподібності використовується для порівняння величини імовірнісних меж різних станів під час остаточного діагнозу.

Якщо дотримано вище зазначені вимоги, то відбудеться збір симптомів ST, і за допомогою операторів злиття їм буде присвоєний фінальний рівень $f_{DT_{srt}}$. Кожен елемент $f_{DT_{srt}}$ буде містити поточний стан S, CS, бо всі симптоми в ST є двійковими, а BST, BSF, або BST та BSF повинні містити стан z. Виходячи з того, що всі симптоми дають правильні прогнози, то вибрані стани повинні містити CS в якості свого елемента. Також існує симптом, стан котрого міститься в CS, а не в z, бо в іншому випадку і CS і z поділяли б одну сигнатуру на двох. Таким чином,

функція правдоподібності викликається на певному стані в z, $pl(f_{DT_{srt}}, \{z\}$ і просто складає разом усі маси станів, що містяться в z, оскільки CS міститься в усіх поточних станах FDT. Всі інші стани z виникають рідше, тому $pl(f_{DT_{srt}}$, {CS}) завжди буде мати найбільше значення в стані $z \in DT_{srt}$, і правильне передбачення буде зроблено незалежно від призначення фактичної маси. Даний процес описано для окремих станів, але з нього стає ясно, що $pl(f_{DT_{srt}}$, x), де x ∈ P(DT_{srt}) та CS ∈ x буде нижчим ніж $pl(f_{DT_{srt}}$, {CS}) за умови, що третє припущення справедливо при порівнянні SIG(CS) та SIG(x). Поки правдоподібність більша і поки існує двійковий симптом, де CS міститься тільки в одному розділі, а стани x відбуваються в іншому, система буде функціювати.

3.3.11. Методика налаштування параметрів діагностики

Процес виявлення симптомів громіздкий і часто видає багато помилок. Щоб цього уникнути, при присвоєнні довір BST і BSF, в станах бінарного симптому при прогнозуванні актуалізується CS. Як правило, якщо симптом st правильно визначений при прогнозуванні стану BS (де BS буде симптомом, а BST або BSF множиною станів), йому повинно бути призначено більше маси довіри $f_{DT_{srt}}$ (BS) ≈ 1, а якщо симптом st некоректний, то йому буде призначено меншу масу віри $f_{DT_{srt}}$ (BS) ≈ 0. Чим більша маса довіри представлена в BS, тим більше вона буде ураховуватись при злитті, а чим менша маса - тим менше вона буде враховуватись.

Система може бути побудована для призначення мас ймовірності до набору симптомів. Якщо є |ST| бінарні симптоми, кожен з яких має BST і BSF стани, то буде два |ST| стани, яким потрібно призначити маси ймовірності. Спочатку всі маси будуть встановлені в певне початкове значення. Потім система почне працювати під наглядом оператора. Якщо система передбачила CS неправильно, наприклад, c ∈ DT_{srt}, а оператор ідентифікує невідповідність і встановлює справжній CS в стан $d \in DT_{srt}$, то можна моніторити кожен стан з z, що випускає симптом з ST, і занизити масу ймовірності неправильних визначень c ∈ z ∧ d ∈ z. Якщо система не змогла правильно передбачити CS через те, що множини станів

DT_{srt} були назначені після порівнянь достовірності, а оператор не помітив помилку, то з часом система сама по собі завищить масу правильних і занизить масу неправильних станів.

Недоліком такого методу є те, що навіть навчена таким чином система може виводити неправильний результат. Коректність залежить від кількості симптомів, що містяться в ST, кількості симптомів, що виробляють неправильні набори станів, і важливості ролі, яку неправильно вироблені стани відіграють при визначенні CS. Можна навіть навчити систему так, щоб вона завжди видавала хибні значення CS. Але якщо можливість виявлення правильного діагнозу працює коректно, то з часом система зможе перейти в правильний режим виявлення несанкціонованих дій шляхом перерозподілу мас важливості серед своїх станів.

Як вже говорилося раніше, хоч листи дерева діагностики й можуть визначити точний набір станів, інші вузли апроксимуються як об'єднання множин станів їх похідних. У випадках, коли система переходить в стан, який міститься в стані набору вузла, але не в його наближеному стані, діагностика буде повертати вузол, що передує справжньому вузлу в дереві діагностики. Такий результат буде менш точним. Єдиний спосіб забезпечити те, щоб конкретний вузол завжди отримував найбільш точний діагноз, це повністю забезпечити пояснення набору станів об'єднанням множин станів його попередніми вузлами. Це призведе до збільшення розміру і кількості вершин листів, що збільшить структуру проникнення, котра буде вимагати більшої кількості симптомів для правильного виконання діагностики. Підвищення достовірності, втраченої в результаті встановлення станів наближення, вимагає додаткових обчислювальних ресурсів через складність кінцевої моделі. З-поміж величезної кількості можливих станів, в яких система може бути, отримання абсолютно точного діагнозу практично недосяжне.

Додатковою перевагою навчання мас є інформація, яку воно надає про якість симптомів. Якщо симптом має низьку масу ймовірності на обох своїх станах, то він буде автоматично замінений іншим, точнішим симптомом.

3.4. Висновки до розділу 3

1. Представлено модель виявлення НД, що базується на діагностуванні вторгнень в комп’ютерній мережі, без оновлення сигнатур. Дана модель основана на роботі з симптомами вторгнень, котрі за допомогою використання операторів ТДШ формують відповідне рушення про наявність чи відсутність НД в системі.

2. Для виявлення вторгнень описано ознаки НД, які найкраще характеризують дії зловмисника чи атаки.

3. Проаналізовано сучасні математичні моделі діагностування і виявлено ті елементи, котрі можна застосувати для виявлення НД в комп’ютерних мережах. Вирішено опис роботи моделі виконати за допомогою операторів ТДШ, що забезпечить можливість уникнення типових помилок виявлення НД і підвищити ефективність окремих елементів. Розглянуто основні алгоритми виявлення зміни, бо вони будуть використовуватися при діагностуванні для контролю часових фрагментів і формування даних для визначення симптомів і сигнатур вторгнень. ТДШ має всі властивості, котрі можуть допомогти виявляти вторгнення і несанкціоновані дії в комп'ютерній мережі.

4. Було описано всю необхідну інформацію для організації діагностування. Дано визначення системи й станів, в яких перебуває система, а також вимоги до цих станів. Для забезпечення функціонування та організації станів системи на різних рівнях деталізації ведено дерево специфікацій. Воно використовується для побудови дерева діагностики, котре моделює аномальну поведінку системи. Дано пояснення спостережуваності й того, як формуються і використовуються симптоми для діагностики системи. Показано як кілька симптомів можуть бути об'єднані разом для забезпечення точнішого діагнозу. Показано, що діагностика буде працювати коректно як в лабораторних, так і в реальних умовах, за допомогою функції налаштування маси симптомів.

РОЗДІЛ 4
ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

4.1. Опис інструментальної бази

При реалізації роботи виявлення НД засобами ШІМ застосовано інструменти емуляції та аналізу інформації. Для побудови блоку моніторингу використано можливості Deterlab (з вмонтованим емулятором Emulab) для проведення збору даних і отримання протоколів роботи.

Для побудови блоку порівняння в Deterlab розроблено доповнення Compair, яке з даних емулятора Emulab відбирає потрібні для проведення експерименті ідентифікатори НД.

ШІМ організується будь якими інструментами засобами GNU Octave, котрі дають можливість реалізації імунних мереж і також сумісні з Deterlab. У таку ШІМ надходить отримані за допомогою Emulab ідентифікатори використовуваних потоків. Вихідною інформацією будуть матриці пам'яті та подібності антитіл, на основі якої можливо буде графічно зобразити ШІМ за допомогою засобів математичного обчислювання і моделювання (GNU Octave).

Для реалізації можливості виявлення нових НД (не з навчальної вибірки) до навченої ШІМ будуть досилатися нові зразки вторгнень (які будуть виявлені відповідними антитілами). Даний функціонал буде забезпечений використанням інструменту NetSim наявного в Deterlab [124] (NetSim забезпечить досилання нових НД до навченої ШІМ).

Експеримент був побудований з використанням інструментів Detelab [124], які надають доступ до необхідних архітектур, операційних систем і програмного забезпечення. Налаштування Deterlab для проведення експерименту представлено на рис. 4.1.

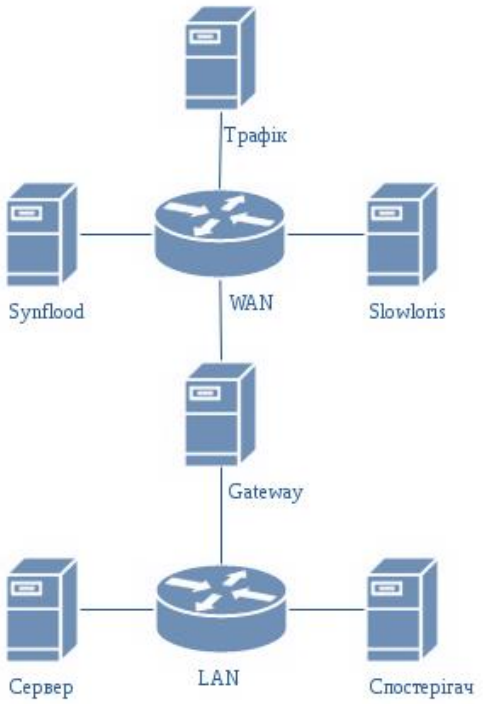


Рис. 4.1. Налаштування Deterlab

Кожен прямокутний вузол являє собою обчислювальну машину, кожне коло - мережеві комутатори, а лінії представляють мережеве підключення. Вузол трафіку призначений для з'єднання потоку http-запитів з сервером. З'єднання виконане за допомогою Apache Benchmarking Tool [130], що також регулює кількість трафіку.

Налаштування даної топології описується наступним чином [124]:

```
set ns [new Simulator]
source tb_compat.tcl
# Кількість вузлів
set NODES 6
set lanstr ""
for {set i 0} {$i < $NODES} {incr i} {
    set node($i) [$ns node]
    append lanstr "$node($i) "
}
# Налаштування затримки
set lan0 [$ns make-lan "$lanstr" 100Mb 0ms]
$ns rtproto Static
$ns run
```

Вузол Synflood по команді атакує серверний вузол атаками типу SynFlood [17, 48]. Принцип атаки полягає в тому, що зловмисник, посилаючи SYN запити,

переповнює на сервері чергу на підключення. При цьому він ігнорує Syn + ACK пакети цілі, не надсилаючи відповідні пакети, або підробляє заголовок пакета таким чином, що відповідь Syn + ACK відправляється на вигадану адресу. У черзі підключень з'являються так звані напіввідкриті з'єднання, що чекають підтвердження від клієнта. Після закінчення тайм-ауту ці підключення відкидаються. Завдання зловмисника полягає в підтримці заповненої черги, щоб не допустити нових підключень. Через це клієнти, що не є зловмисниками, не можуть установити зв'язок, або встановлюють його з суттєвими затримками.

Вузол Slowloris [17, 48] атакує сервер атаками типу Slowloris. Ця атака працює на рівні додатків по протоколу http. Slowloris дозволяє вивести веб-сервер з ладу за допомогою помилки в HTTP. Slowloris намагається встановити багато підключень з веб-сервером та тримати їх відкритими. Це досягається шляхом відкриття з'єднання з веб-сервером та надсиланням спеціальних HTTP-заголовків. Періодично, він буде посилати знову і знову такі HTTP- заголовки не завершуючи попередні запити.

Вузол Gateway виступає шлюзом з "WAN" до внутрішнього вузла сервера "LAN". Весь трафік між мережами повинен проходити через вузол Gateway. По команді, вузол запускає інструмент командного рядка утилітою tcpdump [131] для захоплення всього мережевого трафіку з "WAN" до вузла сервера.

Вузол Спостерігача віддалено контролює вузол сервера, шляхом періодичної відправки http-запитів і перевірки доступності та часу відклику.

Вузол сервера в першу чергу запускає http-сервер Apache [130], котрий матиме доступ до журналу звернень (внутрішня функція, що реалізована в Deterlab для роботи зі зверненнями при проведенні досліджень). Також реалізовується можливість роботи зі статистикою операційної системи через модуль періодичних інтервалів.

Всі вузли запускаються одночасно при виконанні експерименту. Кожен вузол має скрипт, який визначає виконувані ним функції. Всі скрипти вузлів залежать від глобальної задачі й методів її рішення. Робота скриптів синхронізується на

кожному етапі експерименту можливостями Deterlab, що забезпечують можливість реалізації паралельного виконання.

Глобальний скрипт активує вузол сервера та очікує його запуску. Далі запускаються всі механізми відбору на вузлах Сервера, Спостерігача і Gateway.

Вузли чекають запуску своїх механізмів відбору, після чого запускається генератор трафіку і вузли атак (Synflood, Slowloris) з наперед заданою частотою.

Всі вузли чекають поки сервер почне приймати атаки й трафік, а семплери будуть записувати його поведінку. Після закінчення експерименту, вимикаються спочатку вузли атак, а потім вузли генерації трафіку, після чого вимикається сервер і Deterlab дезактивується. Всі записані в результаті експерименту зразки файлів зберігаються в централізованому сховищі. Всі робочі регістри експериментальної установки після збереження необхідної інформації очищаються, і після повторної синхронізації експеримент може бути продовжений.

4.2. Отримання і обробка первинних даних при виявленні несанкціонованих дій

Для отримання протоколів роботи, вторгнення запускаються на емуляторі Emulab. Для тестування взято набір типових для середовища Deterlab НД з з рис 4.2. Дані НД відібрано через можливість зміни інтенсивності їх виконання, що дозволяє їм обходити типові мережеві екрани.

Рис. 4.2. Набір НД

Для приведених вище НД емулятором отримано протоколи роботи, які за допомогою порівняння утилітою Compaire формують вибірку характерних поведінкових ознак.

В результаті порівняння протоколів виявлено 15 ознак для Synflood і Slowloris:

Ознака1 – міститься у: **3; 4; 5; 6; 7; 8; 9; 14; 15.**
Рейтинг появи: 9/15 = 0.6;
Ознака2 – міститься у: **1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15**.
Рейтинг появи: 15/15 = 1;
Ознака3 – міститься у: **1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15**.
Рейтинг появи: 15/15 = 1;
Ознака4 – міститься у: **1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 14; 15**.
Рейтинг появи: 13/15 = 0.86;
Ознака5 – міститься у: **1; 2; 3; 4; 5; 6; 10; 11; 12**.
Рейтинг появи: 9/15 = 0.6;
Ознака6 – міститься у: **1; 2; 3; 4; 5; 10; 11**.
Рейтинг появи: 7/15 = 0.46;

Ознака7 – міститься у: **1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15**.
Рейтинг появи: 15/15 = 1;
Ознака8 – міститься у: **1; 2; 3; 4; 5; 6; 7; 8; 11; 12; 13; 14; 15**.
Рейтинг появи: 13/15 = 0.86;
Ознака9 – міститься у: **1; 2; 3; 4; 5; 6; 7; 10; 11; 12; 13; 14; 15**.
Рейтинг появи: 13/15 = 0.86;
Ознака10 – міститься у: **1; 2; 3; 4; 5; 6; 7; 10; 11; 12; 13; 14; 15**.
Рейтинг появи: 13/15 = 0.86;
Ознака11 – міститься у: **1; 2; 3; 4; 5; 6; 7; 10; 11; 12; 13; 14; 15**.
Рейтинг появи: 13/15 = 0.86;
Ознака12 – міститься у: **1; 2; 3; 4; 5; 6; 7; 10; 11; 12; 13; 14; 15**.
Рейтинг появи: 13/15 = 0.86;
Ознака13 – міститься у: **1; 2; 3; 4; 9; 10; 11.**
Рейтинг появи: 7/15 = 0.46;
Ознака14 – міститься у: **1; 2; 3; 4; 5; 9; 10; 11; 12.**
Рейтинг появи: 9/15 = 0.6;
Ознака15 – міститься у: **1; 2; 3; 4; 5; 6; 7; 8; 9; 12; 13; 14; 15**.
Рейтинг появи: 13/15 = 0.86.

Отримано 15 ознак поведінки НД Synflood і Slowloris та визначено частоту їх появи. На основі навчальної вибірки з вирахуваних рейтингів появи ознак у НД і не НД запущено процес моделювання. Обробка отриманої інформації реалізується в GNU Octave.

Для моделювання експерименту організується дерево діагнозів (*DT*) [145] показане на рис. 4.3. При роботі з DeterLab ми можемо сканувати систему C поки вона знаходиться в станах просторів, котрі пов'язані з кінцевими вузлами (як того вимагає *DT*). Щоб перевести систему в стан простору, де вона знаходиться під загрозою атаки (пов'язаної з відповідним вузлом атаки), Deterlab активує необхідний вузол (Synflood, Slowloris), який дозволить почати атаку на сервер. Механізми відбору з різних вузлів спостерігають за системою поки вона знаходиться в стані простору однієї з атак, а потім фіксують конкретні змінні, що характеризують цей стан. Інтенсивність атак можна налаштовувати власноруч, дозволяючи системі отримувати різні зразки одного і того ж самого стану простору атаки. Система може бути поміщена у нормальний стан простору, котрий пов'язаний з вузлом трафіку. Для цього треба запустити генератор трафіку

і дезактивувати вузли атак. Таким чином можна отримати зразки "нормальності" або трафіку.

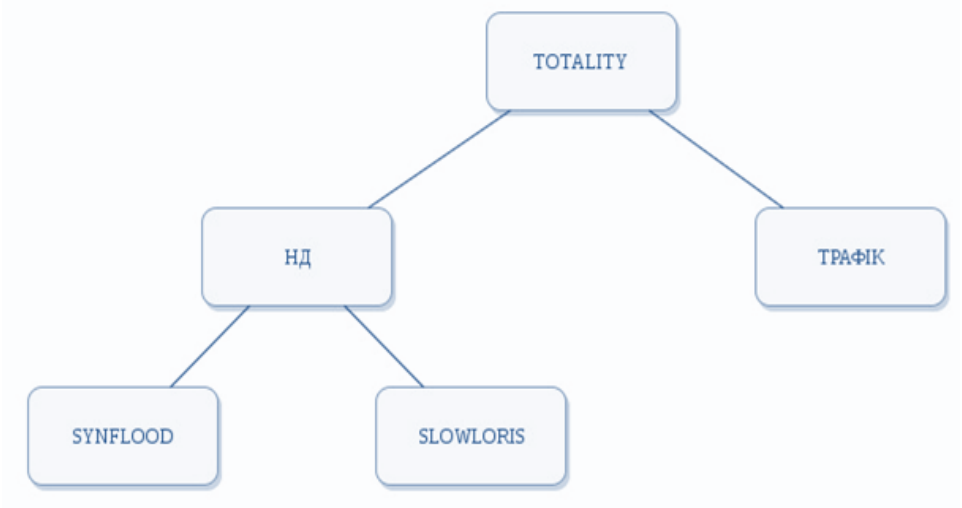


Рис. 4.3. Дерево діагностики

Вузли "Totality", "НД" та "Аномалії" не визначаються по напрямку. Оскільки *DT* базується на визначеннях ТДШ, то стани просторів "НД" та "Аномалії" будуть однаковими (тобто міститись один в одному).

Структура проникнення для *DT* (*DTc*) в ТДШ міститиме наступні елементи *DTc* = {TRAFFIC, SYNFLOOD, SLOWLORIS}. Стани атак Synflood і Slowloris будуть представлені як {SYNFLOOD} ⊆ *DTc* і {SLOWLORIS} ⊆ *DTc*. Стан трафіку буде представлений як {TRAFFIC} ⊆ *DTc*. Клас атак НД міститиме стани атак {SYNFLOOD, SLOWLORIS} ⊆ *DTc*.

Для перевірки властивостей методу діагностування невідомих атак буде сконструйовано невідому системі несанкціоновану діяльність, котра буде входити в стан простору наявних атак. Відомі наперед атаки працюють наступним чином - запущені при відповідній інтенсивності вони спричиняють затримку роботи системи. Тобто при досягненні граничної інтенсивності вони перестають бути трафіком і стають атаками. У нашому випадку доцільно буде тестувати систему на новий тип атаки використовуючи атаки Synflood і Slowloris запущені одночасно, але при граничній до атаки інтенсивності. Тобто негативного впливу на систему при поодинокому спрацюванні вони не будуть мати, але одночасно їх сумарна інтенсивність буде відповідним чином виливатися у відповідний діагноз

"нова атака". Така атака була протестована на різних рівнях інтенсивності й відібрано зразки що викликають перебої в роботі системі (саме вони будуть характеризувати їх стан простору).

З вибірки зібраних протягом експериментів файлів нараховано 919 спостережуваних. Переважна більшість з них не містить необхідних для побудови *DT* станів. З усієї вибірки відібрано лише 17 придатних для моделювання спостережуваних O (табл. 4.1).

Таблиця 4.1

Набір спостережуваних

Ідентифікатор	Ім’я монітора	Ім’я Сервера	Ім’я спостережуваної
1.	MOC	Сервер	LOADAVG - Active Processes
2.	MOC	Сервер	NETSTAT-TcpExt-Delayed ACK Locked 1
3.	MOC	Сервер	NETSTAT - TcpExt - Delayed ACK Lost 1
4.	MOC	Сервер	NETSTAT - TcpExt - Listen Drops
5.	ARM	Сервер	Serve Time
6.	ARM	Сервер	Final HTTP Status
7.	ASM	Сервер	Closing Connection
8.	ASM	Сервер	Logging
9.	ASM	Сервер	Waiting For Connection
10.	SAM	Спостерігач	Response Time 1
11.	SAM	Спостерігач	Wget Return Code
12.	PM	Шлюз	IPv4 Checksum
13.	PM	Шлюз	TCP Flag ACK 1
14.	PM	Шлюз	TCP Flag FIN
15.	PM	Шлюз	TCP Flag PSH
16.	PM	Шлюз	TCP Flag SYN 1
17.	PM	Шлюз	TCP Acknowledgement Number 1

У табл. 4.1 у колонці «Ім’я монітора» MOC означає «Монітор операційної системи», ARM - Apache Request Monitor, ASM - Apache Status Monitor, SAM - Server Availability Monitor і PM - Монітор пакетів (відповідно до рис. 4.1).

Монітори розміщені на вузлах шлюзу, спостерігача, а також Сервера. Вони збирають значення спостережуваних змінних за допомогою сканування з заданим періодичним інтервалом, або отримуючи значення спостережуваних у відповідь на подію (наприклад, прихід пакета). Монітори працюють на різних швидкостях,

оскільки точність кожного сканування різна і залежить від джерела даних. Щоб симптом отримував послідовно вхідні дані, всі спостережувані спочатку об'єднуються у єдине "значення опису" за встановлений проміжок часу (в одну секунду, в даному випадку). Значення опису містить те, скільки разів значення спостережуваної входило в дане вікно часу, її середнє значення і зміну. Якщо не зібрано спостережуваних значень за цей період часу, то встановлюється прапор, який вказує що значення не вимірювалося (NM), і значення досліджуваної приймається за останнє виміряне значення. Коли алгоритм виявлення зміни симптому побудований, він може переймати переваги як інтерпольованих, так і невиміряних прапорів значень під час своєї роботи.

4.3. Виявлення НД

Для перевірки коректності виявлення НД взято Synflood і Slowloris. Для навчання ШІМ використовується вибірка рейтингів появи ознак НД і не НД з табл. 4.2.

Таблиця 4.2

Вибірка для виявлення НД сімейств Synflood і Slowloris

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1		1	1	0.86	0.6	0.46	1	0.86	0.86	0.86	0.86	0.86	0.46	0.6	0.86
2		1	1	0.86	0.6	0.46	1	0.86	0.86	0.86	0.86	0.86	0.46	0.6	0.86
3	0.6	1	1	0.86	0.6	0.46	1	0.86	0.86	0.86	0.86	0.86	0.46	0.6	0.86
4	0.6	1	1	0.86	0.6	0.46	1	0.86	0.86	0.86	0.86	0.86	0.46	0.6	0.86
5	0.6	1	1	0.86	0.6	0.46	1	0.86	0.86	0.86	0.86	0.86		0.6	0.86
6	0.6	1	1	0.86	0.6		1	0.86	0.86	0.86	0.86	0.86			0.86
7	0.6	1	1	0.86			1	0.86	0.86	0.86	0.86	0.86			0.86
8	0.6	1	1	0.86			1	0.86							0.86
9	0.6	1	1	0.86			1						0.46	0.6	0.86
10		1	1	0.86	0.6	0.46	1		0.86	0.86	0.86	0.86	0.46	0.6	
11		1	1	0.86	0.6	0.46	1	0.86	0.86	0.86	0.86	0.86	0.46	0.6	
12		1	1		0.6		1	0.86	0.86	0.86	0.86	0.86		0.6	0.86
13		1	1				1	0.86	0.86	0.86	0.86	0.86			0.86
14	0.6	1	1	0.86			1	0.86	0.86	0.86	0.86	0.86			0.86
15	0.6	1	1	0.86			1	0.86	0.86	0.86	0.86	0.86			0.86

За допомогою GNU Octave організуємо навчання ШІМ на даних з табл. 4.1.

Дендрограмму навченої імунної мережі зображено на рис. 4.4. Судячи по отриманій інформації було утворено два кластери вхідних даних. Перший кластер складається з антитіл: 1, 2, 5, 7, 13, 15; другий – з антитіл: 3, 4, 6, 8, 9, 10, 11, 12, 14. Отримані дані для виявлення нових антигенів оброблюються за допомогою NetSim рис. 4.5.

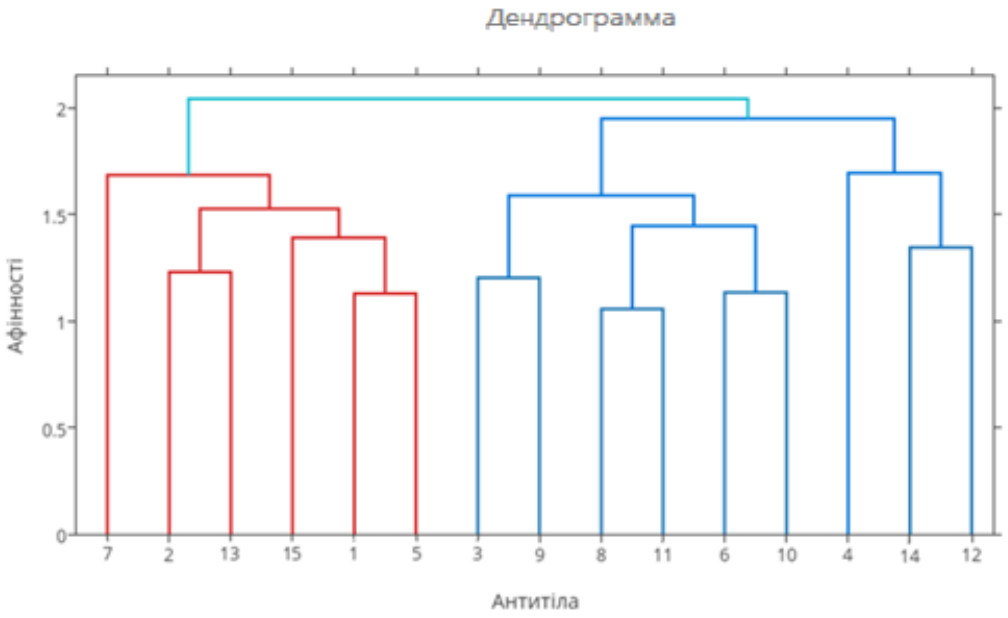


Рис. 4.4. Дендрограма мережі

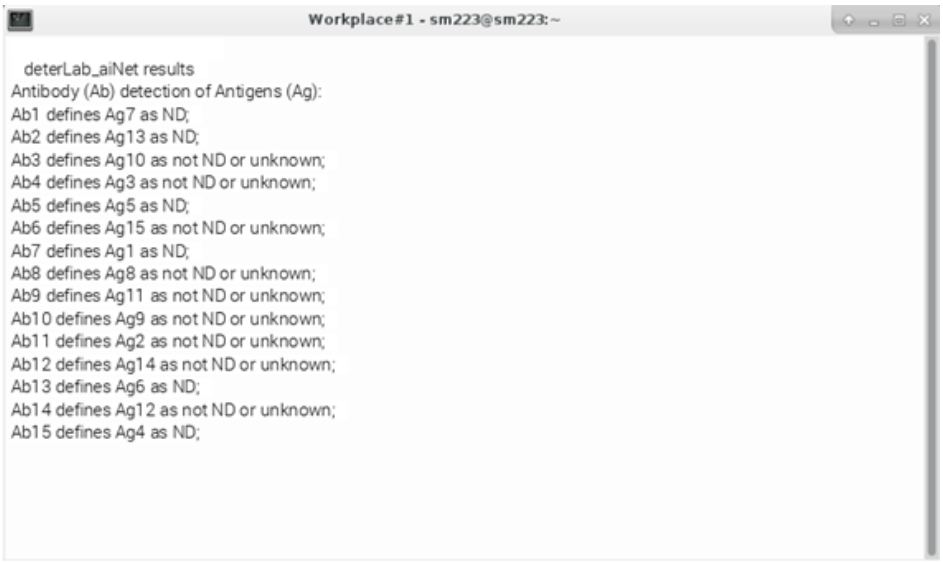


Рис. 4.5. Результат роботи NetSim

Антитіла 1, 2, 5, 7, 13, 15 виявляють антигени НД (1-й кластер); антитіла 3, 4, 6, 8, 9, 10, 11, 12, 14 виявляють антигени не НД або вторгнення інших типів (2-й кластер).

Для виявлення нових антигенів НД використаємо одночасно запущені Synflood та Slowloris атаки з середньою інтенсивністю:

1-й антиген – loggining ;

2-й антиген – Payload Remaining Length;

3-й антиген – MEMINFO – Dirty;

4-й антиген – NETSTAT - TcpExt - Listen Drops.

Характеристики нових антигенів показані у табл. 4.3.

Таблиця 4.3

Характеристики нових антигенів

Ag	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1	0	1	1	0.86	0.6	0.5	1	0.86	0.86	1	0.86	0.86	0.5	0.6	0.86
2	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0.5
3	0	0	0	1	0	0	0	0	0.5	0	0	0	0	1	0
4	0	1	1	0.86	0.6	0.46	1	0.86	0.86	1	0.86	0.86	0.46	0.6	0.86

Результати виявлення нових антигенів показано на рис. 4.6.

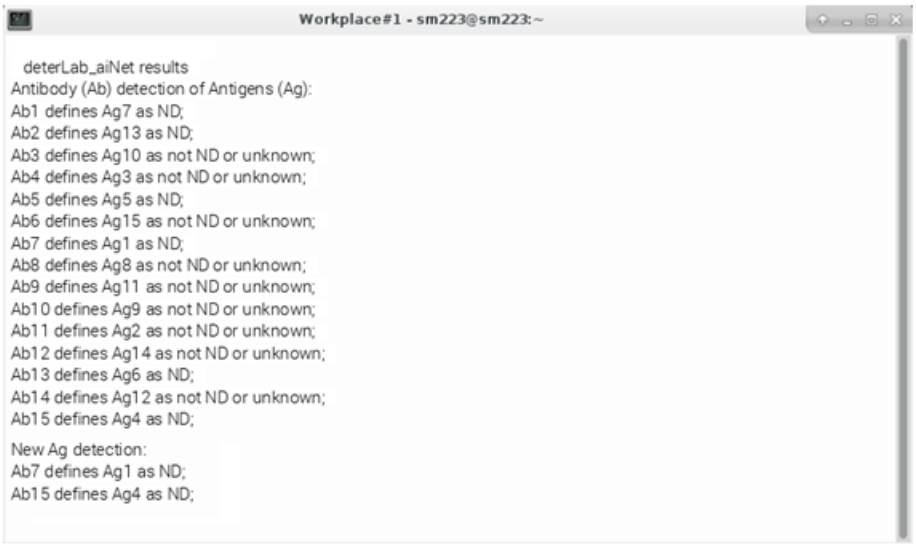


Рис. 4.6 Результати виявлення нових антигенів

З рис. 4.6 зрозуміло, що навчена імунна мережа змогла прийняти коректне рішення: 1-й і 4-й антигени віднесені до НД. 2-й та 3-й антигени не було віднесено до вторгнень, оскільки навіть при середній активності і роботі одночасно вони не завдають ніякої шкоди.

Навчена ШІМ коректно виявила всі вторгнння на роботу з якими була програмована, а також змогла ідентифікувати нові (ті що не входили у навчальну

вибірку) антигени. Таким чином дана модель може працювати не лише з наперед заданими діями, а й виявляти їх модифікації автономно, без участі користувача.

На наступному етапі діагностування береться набір спостережуваних O для побудови симптомів ST. Всі побудовані симптоми бінарні, тобто алгоритм рішення бере значення фрагментів часу серії спостережуваних і видає логічне значення, яке вказує на присутність чи відсутність симптому в даний момент часу.

Побудовано 22 симптоми. Спостережуваних було лише 17, а це значить, що деякі параметри були використані декілька разів. Якщо симптоми мають спільні спостережувані, то алгоритм рішення класифікує їх на основі різних аспектів часових фрагментів. Такі симптоми будуть залежними один від одного при злитті їх переконань (у нашому випадку 5 симптомів являються залежними).

Усі симптоми базуються на CSM алгоритмах рішення. Алгоритм рішення і його параметри підбираються після аналізу часових фрагментів відомих станів.

Результат симптомів побудованих алгоритмом CSM можна побачити у табл. 4.4.

Таблиця 4.4

Симптоми отримані алгоритмом рішення CSM

SID	OID	AVG	STD	DRV	UA	UM	LA	LM	UDP	NSD	XSD	DFT	LGR
1	4	0	0.02	-	4	5	-	-	-	-	-	-	-
2	5	0	20000	-	5	10	-	-	05	-	-	0.32	240
3	6	0	1000	-	5	10	-	-	04	-	-	0.5	-
4	7	102	200	-	-	-	-7	-14	05	-	-	0.2	-
5	8	2	0.2	-	-	-	-20	-40	-	-	-	-	-
6	9	1.7	0.1	-	-	-	-5	-10	-	-	-	0.3	300
7	9	1.7	0.1	-	5	10	-	-	-	-	-	0.3	300
8	10	0.27	0.2	-	-	-	-5	-10	-	-	-	0.04	-
9	8	2	0.2	-	-	-	-20	-40	-	-	-	-	-
10	11	3.22	0.4	-	5	10	-	-	-	-	-	0.02	-
11	12	3.01	0.2	-	-	-	-5	-10	-	-	-	0.1	400
12	12	3.01	0.2	-	5	10	-	-	-	-	-	0.1	-
13	13	$42849*10^5$	$4*10^7$	-	-	-	-20	-40	-	-	4	2.01	-
14	14	9514	1200	-	-	-	-20	-40	-	-	-	1	400
15	14	9514	1200	-	20	40	-	-	-	-	-	1	400
16	1	25.7	100	-	-	-	-13	-20	-	-	-	0.14	-
17	2	1108	822	-	-	-	-15	-20	-	-	-	1.6	-
18	3	4094	28000	+	-	-	-10	-20	-	-	-	0.02	400
19	17	$2*10^{-6}$	2	-	10	20	-	-	-	-	-	$2*10^8$	-
20	15	65500	10000	-	-	-	-10	-20	-	1.5	-	-	100
21	16	$2*10^{-6}$	4	-	10	20	-	-	-	-	-	4	-
22	17	$2*10^{-6}$	2	-	10	20	-	-	-	-	-	$2*10^8$	-

SID позначає ідентифікатор симптому, OID - ідентифікатор спостережуваної. AVG і STD - для середнього і стандартного відхилення CSM. DRV - похідна. UA, UM, LA, LM - верхнє значення тривоги, верхнє граничне значення тривоги, зниження тривоги, і нижнє граничне значення тривоги. UDP - це невизначене значення штрафу. NSD мінімальне стандартне відхилення. XSD - максимальне стандартне відхилення. DFT - дрифт. LGR - затримка.

Алгоритм рішення приймає рішення про наявність відповідного стану симптому. Новий тип атаки (відомі атаки з малою інтенсивністю) входитиме до стану наявності атаки чи вторгнення. Недолік CSM - низька швидкість детектування несанкціонованих дій (але він більш точний). Цей алгоритм "простоює" до 20 секунд перед віднесенням спостережуваної до стану наявності у ній атаки чи НД (цей час витрачається на обрахування всіх даних для винесення вердикту).

Після закінчення роботи алгоритму рішення для симптому, на вихід подаються дані основного переконання по наявності чи відсутності симптому (*Sta* або *Stp*). *Sta* - говорить про відсутність симптому, а *Stp* - про його присутність у даній спостережуваній. У табл. 4.5 наведені основні переконання по кожному окремому симптому (st∈ST), що подавався на вхід алгоритму рішення (де «+» - симптом наявний; «-» - симптом відсутній; & - невизначений (може бути як «+», так і «-»)).

Кожен рядок характеризує симптом st ∈ ST, а кожна колонка характеризує стан структури проникнення дерева діагностики. Для визначення *Stp* основного переконання для певного симптому, дивимось на колонки з відповідними даними («+» або «&» свідчать про наявність симптому). Такий набір станів буде елементом *Stp*, де всі інші елементи будуть його структурою проникнення. Таким же чином будується *Sta*, але про відсутність симптому будуть говорити елементи «-» та «&» у відповідних колонках. Для наведених вище даних використана маса переконання 0.95, а для структури проникнення назначено значення 0.05 (1-0.95 = 0.05). Ці значення вибрані через те, що вони ідеально підходять для побудови бінарних симптомів.

Для трафіку використовується лише значення «-», оскільки він використовується для порівняння зі спостережуваними вузлами НД (трафік характеризується відсутністю симптомів, бо коли він має симптоми вторгнень, то експериментальна установка позиціює його як НД).

Елемент «&» значить, що асоційований стан може бути як у sta так і у stp. (Це пояснює те, що з 17 спостережуваних ми отримали 22 симптомів).

Нова атака продукує сигнатури як присутності, так і відсутності симптомів. Ці сигнатури дають можливість виявити НД і віднести їх до підгрупи "нових" небачених системою атак.

Таблиця 4.5

Основи переконань симптомів

S	**Трафік**	**Synflood**	**SlowLoris**	**New**
1	+	+	&	-
2	-	-	+	+
3	-	-	+	+
4	-	+	+	+
5	-	+	-	+
6	-	+	+	+
7	-	-	+	-
8	-	+	+	+
9	-	+	-	+
10	-	-	+	-
11	-	-	+	-
12	-	+	&	+
13	-	+	-	+
14	-	+	+	-
15	-	-	+	+
16	-	-	+	+
17	-	-	&	+
18	-	-	+	+
19	-	+	-	+
20	-	-	+	-
21	+	&	+	+
22	+	+	&	+

Коли симптоми отримали значення основного переконання, вони об'єднуються у загальне переконання для встановлення діагнозу. Для цього використовуються залежні, незалежні або комбіновані оператори злиття.

Використання "відстані кореляції" Зекеля [132] (коли дано дві змінні як вхідні дані) дозволяє вивести реальне значення у діапазоні [0, 1] (де 0 указує що змінні незалежні, а 1 - залежні). Кожна пара спостережуваних повинна бути розглянута на "залежність" одна від одної за допомогою порогу залежності. Спостережувані з відстанню кореляції меншою за поріг залежності вважаються незалежними. При більшій відстані кореляції - спостережувані будуть залежними. Візуалізація залежностей побудованих таким чином зображена на рис. 4.7. Кожен компонент даного графіку виступає групою спостережуваних, і повинен розглядатися як залежний один від одного. Таким чином залежні симптоми об'єднуються залежними операторами злиття.

$$Z^2(\mathrm{X},\mathrm{Y}) = \frac{V^2(\mathrm{X},\mathrm{Y})}{\sqrt{V^2(\mathrm{X}) * V^2(\mathrm{Y})}},$$

де, Z - відстань кореляції;

V – коваріація відстані;

X,Y – значення ОП для злиття.

Після об'єднання груп основ переконань залежних симптомів операторами злиття, утворені основи переконань будуть незалежними. При їх об'єднанні незалежними операторами злиття вони будуть використовуватись для отримання кінцевих основ переконання для встановлення діагнозу.

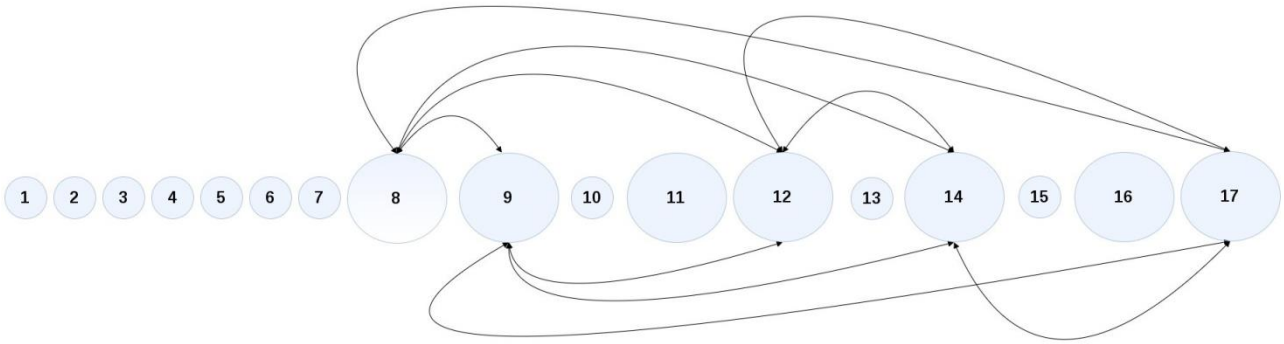


Рис. 4.7. Граф залежності спостережуваних

Числа, які містяться у кожному вузлі, відповідають спостережуваним ідентифікаторам. Лінія, що з'єднує два вузли, представляє залежність. Темні лінії вказують на більше значення кореляції (більш високий рівень залежності між спостережуваними змінними).

Після того, як всі ОП симптому (St) піддалися злиттю, можна переходити до встановлення діагнозу. Порівнюються суб'єктивні межі ймовірності просторів нормального та аномального стану. Для порівняння, найбільші значення правдоподібності визначають обраний простір станів. Якщо значення правдоподібності рівні, то вибирається простір станів з більшим значенням ймовірності. Якщо значення рівні, то діагноз є невизначеним, і повертається стан тотальності.

У разі вибору стану "трафік" система знаходиться у нормальному стані й діагностика переривається. У іншому випадку система в аномальному стані. Процес діагностики переходить на визначення конкретної відомої атаки, якщо вона відповідає поведінці системи S. Кожен стан атак {SYNFLOOD} і {SLOWLORIS} перевіряється на відповідність, якщо жоден з цих станів не підходить (рівень правдоподібності у них нижчий заданого значення (0.8 в нашому випадку)), то вибирається стан "нової" невідомої атаки.

За найбільшим значенням визначається поточний стан системи й діагноз говорить про наявність конкретної атаки чи НД. Якщо і правдоподібність і переконання рівні за значенням, то виконуються оба значення попарно.

4.4. Аналіз ефективності

Для аналізу ефективності розпізнавання НД порівняно існуючі і розроблені методи [140].

У якості критеріїв порівняльної оцінки ефективності запропонованих методів і моделей розпізнавання НД обрані такі системні ресурси комп'ютера:

– час аналізу програм;

– завантаження центрального процесора (ЦП);

– завантаження оперативної пам'яті (ОПП).

Проведемо аналіз витрат системних ресурсів комп'ютера на розпізнавання НД з використанням запропонованих методів і моделей на основі ШІМ та Діагностування. Результати споживання системних ресурсів комп'ютера представлені в табл. 4.6.

Таблиця 4.6

Споживання системних ресурсів

	Навантаження ОПП, %	**Навантаження ЦП, %**	**Час роботи, с**
ШІМ	41	34	14
Діагностика	52	21	12

З наведених у табл. 4.6 результатів можна зробити висновок про те, що по використанню системних ресурсів при розпізнаванні несанкціонованих дій ШІМ і Діагностування суттєво відрізняються. ШІМ дає на 21.2% менше навантаження на оперативну пам’ять в порівнянні з діагностуванням, але на 38.24% і 14.29% поступається при порівнянні використаних ресурсів ЦП і загального часу роботи. Виходить, що додаткові витрати часу на обчислення поколінь при створенні нових клонів уповільнюють ШІМ. За результатами порівняльного аналізу можна зробити висновок, що по швидкодії метод діагностування трохи перевершує ШІМ за рахунок підвищеного використання ресурсів ОПП. Представлені методи

виявили всі вторгнення з навчальної вібірки, тому їх ефективність вирішення поставленої перед ними задачі однакова.

Наступним кроком буде порівняння ефективності розпізнавання несанкціонованих дій за допомогою існуючих рішень [36]. Результати порівняльного аналізу наведено у табл. 4.7.

Таблиця 4.7

Результати дослідження ефективності методів виявлення НД

НД	ШІМ	Діагностика	AVIRA	KASPERSKY
NETSTAT - TcpExt - Delayed ACK Locked1	+	+	+	+
NETSTAT - TcpExt - Delayed ACK Lost 1	+	+	+	-
Response Time 1	+	+	+	+
TCP Flag ACK 1	+	+	+	+
TCP Flag SYN 1	+	+	+	+
TCP Acknowledgement Number 1	+	+	-	+
LOADAVG - Active Processes	+	+	+	+
Serve Time	+	+	+	+
Final HTTP Status	+	+	+	+
Closing Connection	+	+	-	-
Logging	+	+	+	+
Waiting For Connection	+	+	+	-
Wget Return Code	+	+	-	+
TCP Flag FIN	+	+	+	-
TCP Flag PSH	+	+	+	+

Для аналізу ефективності розпізнавання НД в порівнянні з відомими методами обрані програмні рішення Kaspersky і AVIRA, а також розроблені методи на основі ШІМ та Діагностування.

Як видно з наведених у табл. 4.7 результатів, з допомогою розроблених методів були розпізнані всі НД, бо вони були навчені на розпізнавання НД відповідних сімейств.

Існуючі рішення AVIRA і Kaspersky не змогли визначити 3 і 4 НД відповідно, що свідчить про недосконаліть даних програмних рішень при роботі з поведінковим аналізом вторгнень. Ефективність при роботі з навчальною

вибіркою комплексів AVIRA і Kaspersky становить 80% і 73.3%, а запропонованих рішень – 100%. Тобто випробувані методи виявлення несанкціонованих дій у комп’ютерній мережі перевершують існуючі на 20% і 26.6% відповідно.

Результати порівняльного аналізу НД показують, що запропоновані методи перевершують відомі антивірусні продукти, використані в порівняльному тесті і здатні виявити невідомі НД.

За допомогою запропонованих методів розпізнані всі НД, які були присутні у навчальній вибірці (рис. 4.2).

На основі аналізу отриманої інформації зроблено такий висновок: при коректної навчальній вибірці і вірному виборі параметрів навчання ШІМ показує високу точність виявлення нових НД. При тривалому навчанні вибірки виходять більш різноманітні варіанти роботи і антигенів і антитіл, що дозволяє системі гнучко і оперативно реагувати на нові вторгнення. Навчання ШІМ вимагає більше часу на вирішення задачі та додаткових ресурсів ЦП, але при цьому підвищується точність. Діагностування в свою чергу вимагає більше ресурсів оперативної пам’яті для коректного опрацювання вторгнень операторами ТДШ.

ВИСНОВКИ

У дисертаційній роботі на основі проведених досліджень вирішено важливу науково-прикладну проблему – виявлення несанкціонованих дій в комп'ютерній мережі. При цьому отримано такі результати:

1. Виконано системний аналіз принципів й особливостей функціонування засобів виявлення НД в мережах, який дав підстави аргументувати доцільність і можливість створення методів виявлення несанкціонованих дій в комп'ютерних мережах;

2. Проведений огляд існуючих засобів виявлення НД в мережах дозволив визначити основні напрямки дослідження і слабкі сторони сучасних програмних рішень. Подано опис розроблюваних методів виявлення несанкціонованих дій в комп'ютерних мережах, котрі зможуть вчасно і коректно реагувати на вторгнення і працюватимуть автономно;

3. Запропоновано модель виявлення НД в комп'ютерній мережі, в якій розпізнавання вторгнень відбувається через аналіз поведінкових ознак, що дає можливість автономно розпізнавати НД і уникати хибних спрацювань;

4. Отримав подальший розвиток метод виявлення НД в комп'ютерній мережі, який базується на використанні операторів ШІМ для побудови структурованої мережі антитіл. Отже, побудована ШІМ здатна швидше ідентифікувати як відомі, так і нові НД;

5. Представлено модель виявлення НД в комп'ютерній мережі засобами інспектування стану системи на виникнення аномальної поведінки, використовуючи дерево діагностування, що дає змогу відстежувати активність системи та симптомізувати дії користувача, а за рахунок введення нових елементів і діапазонів роботи досягається підвищення достовірності виявлення НД і мінімізація помилкових спрацювань;

6. Запропоновано метод розпізнавання НД в комп'ютерній мережі засобами діагностування на основі операторів ТДШ, де на відміну від існуючих пропонується відстежувати часові фрагменти на заданих діапазонах часу і з них,

за допомогою операторів злиття, формувати діагнози. Все це дозволить автономно виявляти невідомі системі НД;

7. Проведено порівняльний аналіз запропонованих рішень, котрий експериментально підтвердив, що дані методи підвищують достовірність ідентифікації НД і атак в комп'ютерній мережі;

8. Розроблені методи та моделі використані для діагностування НД в комп'ютерній мережі ТОВ «Газбудсервіс», а також для аналізу НД в комп'ютерній мережі ДП «Короп-пласт».

Отримані результати дисертаційної роботи також впроваджено в навчальний процес на кафедрі комп'ютерних систем та мереж Національного авіаційного університету.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Denning D. E. An intrusion-detection model. *In Proc. IEEE Symposium on Security and Privacy*. 1987. Vol. 13, No 2. P. 222-232.

2. Sheyner O. Scenario Graphs and Attack Graphs. PhD thesis, SCS, Pittsburgh : Carnegie Mellon University, 2004. 141 p.

3. Kvarnström H. A survey of commercial tools for intrusion detection. Technical Report, Göteborg : Chalmers University, 1999. 99 p.

4. Edward G. Intrusion Detection. 1st ed., Intrusion.Net Books, New Jersey : Sparta, 1999. 218 p.

5. Eckmann S.T., Vigna G., Kemmerer R. A. STATL: An Attack Language for State-based Intrusion Detection. *Dept. of Computer Science, University of California, Santa Barbara*. 2000. Vol. 12, No 2. P. 71-103.

6. Masayoshi M., Shirahata S. The Design and Implementation of Session Based. *IEICO*. 2005. P.551-562.

7. Vigna G., Kemmerer R. A. NetSTAT: A Network-based Intrusion Detection Approach. *Proceedings of the 14th Annual Computer Security Application Conference*. 2000. P.73-81.

8. Gorodetski V.I., Kotenko V. I. Attacks Against Computer Network: Formal Grammar-Based Framework and Simulation Tool. *Institute for Informatics and Automation. RAID*. 2000. Vol. 2516. P.219-238.

9. Гамаюнов Д.Ю., Смелянский Р.Л. Модель поведения сетевых объектов в распределенных вычислительных системах. *Программирование*. 2007. № 4. С.20–31.

10. Lee W., Stolfo S. Data mining approaches for intrusion detection. *In Proc. of the 7th USENIX Security Symposium*. 1998. No 7. P.79-94.

11. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы. Учебник для вузов. 4-е изд. СПб : Питер, 2010. 944 с.

12. Магницкий Ю.Н. Использование бинарной нейронной сети для обнаружения атак на ресурсы распределенных информационных систем. *Динамика неоднородных систем*. 2008. № 3. С. 200-205.

13. Крыжановский А.В. Применение искусственных нейронных сетей в системах обнаружения атак. *Доклады ТУСУРа*. 2008. – № 2 (18), часть 1. С. 37-41.

14. Гаврилов А.В. Применение постоянно модифицирующихся нейронных сетей для защиты программного обеспечения. *Нейрокомпьютеры: разработка, применение*. 2008. № 1-2. С. 90-101.

15. Lu W., Traore I. Detecting new forms of networkintrusion using genetic programming. *Computational Intelligence.* 2004. Vol.20, Issue 3. P. 475-494.

16. Jeya P. G., Ravichandran M., Ravichandran C. S. Efficient Classifier for R 2 L and U 2 R Attacks. *International Journal of Computer Applications*. 2012. Vol. 45, №21. P. 43-52.

17. Whitman M. E., Mattord H.J. Management of Information. Publisher : Cengage Learning: 4 edition, 2013. 576 p.

18. Debar H., Dacier H., Wespi A. Towards a taxonomy of intrusion detection systems. *Computer Networks*. 1999. № 31. P. 805-822.

19. Оладько В. С., Микова С.Ю., Нестеренко М. А., Садовник Е. А. Причины и источники сетевых аномалий. *Молодой ученый*. 2015. №22. С. 158-161.

20. Басараб М.А., Строганов И.С. Обнаружение аномалий в информационных процессах на основе мультифрактального анализа. *Вопросы кибербезопасности*. 2014. №4(7). С. 30-40.

21. Бил Дж. Обнаружение вторжений. Москва, 2006. 656 с.

22. Гатаулин С. И. Распознавание вирусов с частичной полиморфностью. Москва, 2012. 51 с.

23. Гошко С.В. Технологии борьбы с компьютерными вирусами. СПб.: Солон-Пресс, 2009. 352 с.

24. Корнюшин Н.П., Глушков С.В., Варлатая С.К., Шаханов М.В. Защита информационных процессов в компьютерных сетях. Владивосток, 2015 178 с.

25. Безруков Н.Н. Компьютерная вирусология. Справ. Руководство. Киев

: УРЕ, 1991. 416 с.

26. Абалмазов Э.И. Методы и инженерно-технические средства противодействия информационным угрозам. Москва : Гротек, 1997. 248 с.

27. Медведев Н.В., Марков А.С., Федин А.А. Применение метода статического сигнатурного анализа для выявления дефектов безопасности веб-приложений. *Наука и образование: электронное научно-техническое издание*. 2012. № 9. С. 21-31.

28. Бойцов Л.М. Использование хеширования по сигнатуре для поиска по сходству. *Прикладная математика и информатика*, ВМиК МГУ. 2001. № 8. С. 135-154.

29. Искусственная иммунная система URL: http://info-farm.ru/alphabet_index/i/iskusstvennaya-immunnaya-sistema.html (дата звернення 12.10.2018)

30. Kuznetsov A. A. On the cryptographic security of the "BotikKey" authentication protocol against attacks on MD5 hash function. *Programmnye Sistemy: Teoriya i Prilozheniya*. 2015. № 3(26) P. 135—145

31. Большев А.К., Лисс А.Р. Прототип эвристической системы обнаружения вторжений в компьютерные сети на основе метода главных компонент. *Научно-Технические Ведомости СПбГПУ, Серия «Информатика, Телекоммуникации, Управление*». 2010. Т. 4(103). С. 200-205.

32. Кораблев Н.М., Кушнарев М.В., Ужвий Д.П. Нейросетевой эвристический анализатор вредоносных программ с иммунным обучением. *Радиоэлектроника и информатика: научно-техн.журнал*. 2014. № 2 (65). С. 19-25.

33. Кораблев Н.М., Кушнарев М.В., Ужвий Д.П. Гибридная модель эвристического анализатора вредоносных программ. *Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління*: матеріали п’ятої міжнар. наук.-техн.конференції, м. Полтава, 23-24 квітня 2015 р. Полтава, 2015. С. 27.

34. Проактивные технологии для борьбы с вирусами. URL:

http://citforum.ck.ua/security/virus/proactive_tech (дата звернення 12.12.2018).

35. Головко В.А., Безобразов С.В. Проектирование интеллектуальных систем обнаружения аномалий. *Открытые семантические технологии проектирования интеллектуальных систем:* Матер. междунар. научно-технич. конф. OSTIS-2011. м. Минск, 7 сент. 2011 г. Минск, 2011. С. 185-196.

36. Касперски К. Компьютерные вирусы изнутри и снаружи. Спб.: Питер, 2007. 527 с.

37. Блинов П.А., Андреев А.Н. Программный эмулятор электрокардиосигналов. *Математическое и программное обеспечение вычислительных систем*: *Межвуз. сб. науч. трудов*, 2009. С.117-118.

38. Лягинова О.Ю. Использование моделей аппаратно-программных средств, созданных на базе программ-эмуляторов, в профильном курсе «Информатика и ИКТ». *Сборник «Ученые записки ИИО РАО»*. 2011.Выпуск 34. С. 194-199.

39. Лягинова О.Ю: Использование программ-эмуляторов при изучении-программного обеспечения. *Информатика и образование*. 2010. №12. С. 115-118.

40. Turing A.M. Computing Machinery and Intelligence. *Mind*. 1950. V. 59, № 236. P. 433–460.

41. Дрейфус Х. Чего не могут вычислительные машины: Критика искусственного разума. Пер. с англ. Изд. 2-е. Москва : Книжный дом «ЛИБРОКОМ», 2010. 336 с.

42. Леденева Т.М., Подвальный С.Л., Васильев В.И. Системы искусственного интеллекта и принятия решений: Учеб. Пособие. Уфимск. гос. авиац. техн. ун-т, Воронеж, гос. тех. ун-т. Уфа : УГАТУ, 2004. 206 с.

43. Vere S.A. Relational production systems. *Artificial Intelligence*. 1977. №8. P. 47-68.

44. Яхно Т.М. Системы продукций: структура, технология, применение. Новосибирск : ВЦ СО АН СССР, 1990. 127 с.

45. Жожикашвили А.В., Стефанюк В.Л. Алгебраическая теория продукционных систем. *Восьмая нац. конф. по искусственному интеллекту с*

между-нар. участием. Сб. тр. в 3 томах. Москва: Физматлит. 2002. Т.1. С. 428-436.

46. Иванов А.С. Модель представления продукционных баз знаний на ЭВМ. *Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика*. 2007. Т. 7. № 1. С. 83-88.

47. Тютюнник М.Б. Прототип продукционной системы параллельного программирования. *Открытый дальневосточный конкурс программных средств студентов, аспирантов и молодых специалистов* Тез. докл. 2004. С. 45-49.

48. Лукацкий А.В. Обнаружение атак. СПб.: БХВ-Петербург, 2001. 596 с.

49. Козлов Д.А. Энциклопедия компьютерных вирусов. Москва : «Солон», 2001. 464 с.

50. Дуброва Т.А. Статистические методы прогнозирования: учеб. пособие для вузов. Москва : ЮНИТИ - ДАНА, 2003. 206 с.

51. Дубров А.М., Мхитарян В.С., Трошин Л.И. Многомерные статистические методы: ученик. Москва : Финансы и статистика, 1998. 352 с.

52. Моррис У.Т. Наука об управлении. Баесовский подход. Москва : Мир, 1971. 304 с.

53. Tipping M. Sparse Bayesian Learning. *Journal of Machine Learning Research*. 2001. №1. P. 211-244.

54. Tresp V. A Bayesian Commitee Machine. *Neural Computation*. 2000. P. 2719-2741.

55. Жуков Л. А., П.В. Решетникова. Формализация технологии применения нейронных сетей с учителем и особенности их использования для решения прикладных задач. Красноярск : ИПЦ КГТУ, 2005. 168 с.

56. Golovko V. Neural Networks approaches for Intrusion Detection and Recognition. *Computing*. 2006. Vol. 5. № 3. P. 118-125.

57. Atencia M. A., Joya G, Sandoval F. A formal model for definition and simulation of generic neural networks. *Neural Processing Letters, Kluwer Academic Publishers*. 2000. vol. 11. P. 87-105

58. Shivani S., Himali J., Sathvik S., Kiran B. Virus Detection using Artificial

Neural Networks. *International Journal of Computer Applications*. 2013. Vol. 84. № 5. P. 17-33.

59. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. Москва : Горячая линия-Телеком, 2004. 452 с.

60. Осовский С. Нейронные сети для обработки информации. Москва : Финансы и статистика, 2002. 344 с.

61. Синявский О.Ю., Кобрин А.И. Обучение спайкового нейрона с учителем в задаче детектирования пространственно-временного импульсного партерна. *Нейрокомпьютеры: разработка и применение. М. Радиотехника*. 2010. №8. С. 69-76.

62. Редысов В.Г. Эволюция, нейронные сети, интеллект: Модели и концепции эволюционной кібернетики. Москва : УРСС, 2005. 224 с.

63. Устиненков Е.С. Способ анализа мультиагентных систем на основе нечетких когнитивных карт. *Информационный бюллетень Академии военных наук*. 2010. № 22. С. 110-115.

64. Каляев А.И. Мультиагентная организация облачных вычислений на базе сети компьютеров частных пользователей. «*Высокопроизводительные вычислительные системы» Труды молодых ученых ЮФУ и ЮНЦ РАН*. 2012. С. 68-72.

65. Kotenko I. V., Alexeev A., Mankov E. Formal Framework for Modeling and Simulation of DDoS Attacks Based on Teamwork of Hackers-Agents. *I AT. IEEE Computer Society*, 2003. P. 507-510.

66. Lee H.-W., Kwon T., Kim H. NS-2 Based IP Traceback Simulation Against Reflector Based DDoS Attack. *AIS Ed. by T. G. Kim*. Vol. 3397 of Lecture Notes in Computer Science. Springer. 2004. P. 90-99.

67. Уланов А.В., Котенко И.В. Защита от DDoS-атак: механизмы предупреждения, обнаружения, отслеживания источника и противодействия. *Защита информации. Инсайд*. 2007. № 1. С. 60-67.

68. Котенко И.В. Многоагентные технологии анализа уязвимостей и

обнаружения вторжений в компьютерных сетях. *Новости искусственного интеллекта*. 2004. № 1. С. 56–72.

69. Воробьев А., Быков А., Караулов А. Иммунология и Аллергология. Москва : Практическая медицина, 2006. 282с.

70. Кетлинский С.А., Симбирцев А.С., Воробьёв А.А. Эндогенные иммуномодуляторы. Санкт-Петербург : Гиппократ, 1992. 264с.

71. Корнева Е. А. Введение в иммунофизиологию Санкт-Петербург : ЭЛБИ-СПб., 2003. 310 с

72. Ярилин А. А. Основы иммунологии: Учебник. Москва : Медицина, 1999. 608с.

73. Dasgupta D. Artificial Immune Systems and Their Applications. Berlin : Springer, 1999. 306 p.

74. Atreas N. D., Karanikas C. G., Tarakanov A. O. Signal processing by an immune type tree transform. *Lecture Notes in Computer Science*. 2003. No 2787. P. 111-119.

75. Тараканов А. О., Гончарова Л. Б. Иммунокомпьютинг биочип – биокомпьютер. *Труды СПИИРАН*. 2002. Вып.1, т.2. С. 92-104.

76. Castro L.N., Timmis J.I. An Artificial Immune Network for Multimodal Function Optimization. *Evolutionary Computation: IEEE Congress*, 3-7 April, 2002 proceedings. Hawaii, 2002. Vol. 1. P. 674-699.

77. Castro L.N., Timmis J.I. Artificial Immune Systems: A Novel Paradigm to Pattern Recognition. *Soft Computing*. 2002. P. 67-84.

78. Graaff A.J., Engelbrecht A.P. Optimised coverage of non-self with evolved lymphocytes in an artificialimmune system. *Intl. J. Computational Intelligence Res*. 2006. P. 127-150.

79. Аткина В.С. Применение иммунной сети для анализа катастрофоустойчивости информационных систем. *Известия ЮФУ. Технические науки. Информационная безопасность*. 2011. №12 (125). С. 203-210.

80. Брюхомицкий Ю.А. Мониторинг информационных процессов методами искусственных иммунных систем. *Известия ЮФУ. Технические науки.*

Тематический впуск «Информационная безопасность». 2012. №12 (137). С. 82-90.

81. Котов В.Д., Васильев В.И. Система обнаружения сетевых вторжений на основе механизмов иммунной модели. *Известия ЮФУ. Технические науки. Тематический выпуск «Информационная безопасность»*. 2011. №12 (125). С. 180-190.

82. Гаврилов А.В., Тихомиров А.В. Применение иммунных систем в целях защиты корпоративной информации от нецелевого использования. *Известия Южного федерального университета. Технические науки*. 2010. Т. 108. № 7. С. 154-163.

83. Литвиненко В.И., Бидюк П.И., Фефелов А.А., Баклан И.В. Гибридная иммунная сеть для решения задач структурной идентификации. *Искусственный интеллект.* 2004. № 3. С.89-99.

84. Зайцев С.А., Субботин С.А. Кластерный анализ с использованием гибридной модели на основе искусственной иммунной сети. *Бионика интеллекта.* 2010. №3(74). С.70 - 75.

85. Самигулина Г.А. Разработка интеллектуальных экспертных систем прогнозирования и управления на основе искусственных иммунных систем. *Теоретическая информатика*. 2009. Вып 4. С 15-22

86. Dasgupta D., González F. An immunity-based technique to characterize intrusions in computer networks. *IEEE Transactions on Evolutionary Computation*. 2002. Vol. 6. P. 281–291.

87. Васютин С.В., Лебедев С.В. Построение агентов мониторинга системы обнаружения атак. *Информационная безопасность*: Материалы VI Международной научно-практической конференции, 1-7 июля. г. Таганрог: Изд-во ТРТУ. 2004. С. 181-182.

88. Олдер Р., Бабин Дж., Докстейтер А. Snort 2.1. Обнаружение вторжений. 2-е издание. Пер. с англ. Москва : БИНОМ, 2006. 656 с.

89. Wang K., Stolfo S. Anomalous Payload-Based Network Intrusion Detection. *Lecture Notes in Computer Science*. 2004. Vol. 3224. P. 203-222.

90. Peddabachigari S., Abraham A., Grosan C., Thomas J. Modeling intrusion

detection system using hybrid intelligent. *Netw. Comput. Appl. January*. 2007. Vol. 30. P. 114-132.

91. Дайнеко В. Ю. Эффективные алгоритмы обучения динамических байесовских сетей в системах обнаружения вторжений. *Сборник тезисов докладов конгресса молодых ученых.* 2012. Вып. 1. С. 128-129.

92. Scarfone K., MellGuide P. Guide to Intrusion Detection and Prevention Systems (IDPS).11NIST Special Publication 800-94, 2007. 127 p.

93. Chen Q., Aickelin U. Anomaly Detection Using the Dempster-Shafer Method. *Proceedings of the International Conference on Data Mining* (DMIN2006), Las Vegas, USA, 2006. – P 232-238.

94. Yang B-S., Kim K. J. Application of Dempster-Shafer theory in fault diagnosis of induction motors using vibration and current signals. *Mechanical Systems and Signal Processing.* 2006. Vol. 20 (2). P. 403-420.

95. Dempster A.P. Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Statist*. 1967. Vol. 38, No. 2. P. 325–339.

96. Shafer G. A. Mathematical Theory of Evidence. Princeton and London : Princeton University Press, 1976. 236 p.

97. Voorbraak F. A computationally efficient approximation of Dempster-Shafer theory. *Internat. J. Man-MachineStud*. 1989. Vol. 30(5). P. 525-536.

98. Hussain A. A., Heidemann J., Papadopoulos C. Framework for classifying denial of service attacks. *In Proceedings of the ACM SIGCOMM Conference*, Karlsruhe, Germany, August 2003. P. 99–110.

99. Jakobsson M., XiaoFeng W., Wetzel S. Stealth attacks in vehicular technologies. *In: Proc. of The Vehicular Technology IEEE Conference*, Bloomington, USA, September 26-29, vol. 2, 2004. P. 1218–1222

100. Yu-Sung W., Bagchi S., Garg S., Singh N. SCIDIVE: a stateful and cross protocol intrusion detection architecture for voice-over-IP environments. *In: Proc. Of Dependable Systems and Networks Conference*, June 28, 2004. P. 433–442

101. Majorczyk F., Totel E., Saıdane A. Anomaly Detection with Diagnosisin Diversified Systems using Information Flow Graphs. *In: Proc. of The IFIP 3rd*

International Information Security Conference, July 17, 2008. P. 301–315

102. Chandola V., Banerjee A., Kumar V. Anomaly detection: A survey. *In: ACM Computing Surveys.* 2009. Vol. 41. P. 1-58.

103. Половко А. М., Бутусов П. Н. MATLAB для студента. СПб.: БХВ-Петербург, 2005. 320 с.

104. Neurosolutions User's Manual. Gainesville: NeuroDimension Inc. 1995.

105. Корченко А.О., Гізун А.І., Волянська В.В., Гавриленко О.В. Евристичні правила на основі логіко-лінгвістичних зв'язок для виявлення та ідентифікації порушника інформаційної безпеки. *Захист інформації.* 2013. №3 (60). С. 251-257.

106. Castro D., Von Zuben. Artificial Immune Systems. Part II: A Survey of Applications, Technical Report. RT DCA 02/00. Brazil : FEEC/UNICAMP, 2000. 64 p.

107. Castro D., Von Zuben. Learning andoptimization using the clonal selection principle. *IEEE Trans. on Evolutionary Computation*. 2002. Vol. 6, No. 3. P.239-251.

108. Kelsey J., Timmis J. Immune Inspired SomaticContiguous Hypermutation for Function Optimisation. *Proceedings, Part I Genetic and Evolutionary Computation Conference (GECCO 2003)*. 2003. No 1. P. 207-218.

109. Hart E., Timmis J. Application areas of AIS: the past, present and future. *Journal of Applied Soft Computing*. 2008. Vol. 8. P. 191-201.

110. Harmer K., Williams P. D., Gunsch G. H., Lamont G. B. An artificial immune system architecture for computer security applications. *Transactions on Evolutionary Computation*. 2002. Vol. 6, Issue 3. P 252 – 280.

111. D'haeseleer P., Forrest S., Helman P. An immunological approach to change detection: algorithms, analysis and implications. *In Proceedings of the IEEE Symposium on Security and Privacy*. 1996. P. 10-18.

112. Mazhar N., Farooq M. A Sense of Danger: Dendritic Cells Inspired Artificial Immune System (AIS) for MANET Security. *GECCO'08,* Atlanta, 12-16 July 2008. P. 63-70.

113. Bretscher P., Cohn M. Theory of Self-Non Self. *Science*. 1970. No 169, P. 1042-1049.

114. Aickelin U., Cayzer S. The Danger Theory and Its Application to Artificial Immune Systems. *1st International Conference on Artificial Immune Systems* (ICARIS 2002), Canterbury, 9-11 September 2002. P. 141-148.

115. Prieto C.E., Nino F., Quintana G. A Goalkeeper Strategy in Robot Soccer Based on Danger Theory. *IEEE Congress on Evolutionary Computation*, Hong Kong, 1-6 June 2008. P. 443-447.

116. Mosmann T., Livingstone A. Dendritic cells: the immune information management experts. *Nature Immunology*. 2004. Vol 5(6). P. 564–566.

117. Mahnke K., Johnson T., Ring S., Enk A. Tolerogenic dendritic cells and regulatory t-cells: A two-way relationship. *Journal of Dermatologic Science*. 2007. Vol 46(3). P.159–167.

118. Kim J., Bentley P., Wallenta C., Ahmed M., Hailes S. Danger is ubiquitous: Detecting malicious activities in sensor networks using the dendritic cell algorithm. *In Proc. of the 5th International Conference on Artificial Immune Systems (ICARIS)*, LNCS 4163, London, U.K, 2006. P. 390–403.

119. Greensmith J., Aickelin U., Twycross J. Articulation and Clarification of the Dendritic Cell Algorithm. 5*th International Conference on Artificial Immune Systems* ICARIS 2006, Oeiras, 4-6 September 2006. P. 404-417.

120. Gu F., Greensmith J., Aickelin U. Further Exploration of the Dendritic Cell Algorithm: Antigen Multiplier and Time Windows. *7th International Conference on Artificial Immune Systems*, Phuket, 10-13 August 2008. P. 142-153.

121. KDD CUP99. URL : https://kdd.ics.uci.edu/databases/kddcup99/task.html (дата звернення 12.12.2018)

122. Oates R., Greensmith J., Aickelin U., Garibaldi J., Kendall G. The Application of a Dendritic Cell Algorithm to a Robotic. *6th International Conference on Artificial Immune Systems (ICARIS 2007),* Santos, 26-29 August 2007. P. 204-215.

123. Castro D., Coelho G.P., Caetano M.F. Artificial Immune Systems. *4th International Conference, ICARIS* 2005, Banff, Alberta, Canada, August 14-17, 2005. P. 469-482

124. The DETER Project URL: http://deter-project.org/ (дата звернення

12.12.2018)

125. Zomlot L. Prioritizing intrusion analysis using Dempster-Shafer theory. *In: Proceedings ofthe 4th ACM workshop on Security and articial intelligence*. *ACM.* 2011. P. 59-70.

126. Page E. Continuous Inspection Schemes / E. S. Page // *In: Biometrika*. 1954. Vol 41. P. 100-115.

127. Granjon P. The CUSUM algorithm a small review. GIPSA-lab, 2014. 22 p.

128. Audun J., Diaz J., Rifqi M. Cumulative and averaging fusion of beliefs. *In: Information Fusion*. 2010. Vol. 11(2). P. 192-200.

129. Castro D., Von Zuben. aiNet: An Artificial Immune Network for Data Analysis. USA : Idea Group Publishing, 2001. 40 p.

130. Apache Software Foundation. URL: http://httpd.apache.org (дата звернення 12.12.2018)

131. The Tcpdump Group. URL: http://www.tcpdump.org (дата звернення 12.12.2018)

132. Székely G., Rizzo M., Bakirov N. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*. 2007. Vol. 35, No. 6. P. 2769–2794.

133. Zhukov I. A. Detection of computer attacks using outliner method. *Науковий журнал «Молодий вчений»*. К.: 2016. № 9(36). С. 91-93

134. Балакин С. В. Методы и средства повишения достоверности идентификации несанкционированных воздействий и атак в компьютерной сети. *Комп'ютерні системи та мережні технології :* зб. тез доп. VIII міжнар. наук.-техн. конф., м. Київ, 16-18 квіт. 2015 р. Київ, 2015. С. 11-12.

135. Балакин С. В. Системы предотвращения атак в компьютерной сети на основе сигнатурных методов. *Комп'ютерні системи та мережні технології :* зб. тез доп. IX міжнар. наук.-техн. конф., м. Київ, 21-23 квіт. 2016 р. Київ, 2016. С. 12-13.

136. Балакин С. В. Средства диагностирования несанкционированных воздействий и атак в компьютерной сети. *Комп'ютерні системи та мережні технології :* зб. тез доп. X міжнар. наук.-техн. конф., м. Київ, 20-22 квіт. 2016 р.

Київ, 2017. С. 13-14.

137. Balakin S. V. Traffic analysis for intrusion detection systems in telecommunication networks. *Політ. Сучасні проблеми науки :* зб. тез доп. XIV міжнар. наук.-практ. конф., м. Київ, 2-3 квіт. 2014. С. 59-60.

138. Жуков И. А. Идентификация атак в компьютерной сети методом усредненного времени обращений. *Проблеми інформатизації та управління: зб. наук. праць.* К.: 2015. № 2(50). С.65–69.

139. Балакін С. В. Застосування штучних імунних систем при виявленні шкідливих программ в комп'ютерній мережі. *Проблеми інформатизації та управління: зб. наук. праць.* К.: 2017. № 1-2(57-58). С. 61-68.

140. Жуков И. А. Исследование эффективности метода обнаружения вторжений в компьютерные сети на основе искусственных иммуных систем. *Проблеми інформатизації та управління: зб. наук. Праць.* К.: 2017. № 3(59). С. 65-69.

141. Балакин С. В. Выявление компьютерных атак с помощью мониторинга сетевых объектов. *Технологический аудит и резервы производства*. Харьков, 2015. № 5-6(25). С.36-38.

142. Балакин С. В. Организация пресечения вторжений в компьютерные сети алгоритмами выявления изменений. *Вісник НТУ «ХПІ». Серія: Механіко-технологічні системи та комплекси.* Харків, 2017. № 20(1242). С.3-7.

143. Жуков И. А., С. В. Балакин Обнаружение компьютерних атак с помощью метода отклонений. *Радіоелектронні і комп'ютерні системи: наук. – техн. жур.* Харків, 2016. №5(79). С. 33-37.

144. Спосіб запобігання комп'ютерним атакам у мережі за допомогою фільтрації вхідних пакетів: пат. 110330 Україна: МПК G06F 12/14. №201602196; заявл. 09.03.16; опубл. 10.10.16, Бюл. №19. 4 с.

145. Спосіб діагностування несанкціонованих дій в комп'ютерній мережі: пат. 123634 Україна: МПК G06F 12/14. №201702719; заявл. 23.03.17; опубл. 12.03.18, Бюл. №5. 4 с.

146. Балакин С. В. Оптимизация искусственных иммунных систем при

идентификации несанкционированных сетевых воздействий. *Комп'ютерні системи та мережні технології :* зб. тез доп. XI міжнар. наук.-техн. конф., м. Київ, 19-21 квіт. 2018 р. Київ, 2018. С. 7-8.

ДОДАТОК А. Акти впровадження у виробничий та навчальний процес

ЗАТВЕРДЖУЮ
Директор ДП "Компанія "Короп-пласт"
Мисяк Г.А.
"26" червня 2017 р.

АКТ

про використання результатів науково-дослідної роботи

"Аналізатор несанкціонованих дій в комп'ютерній мережі"

Цим актом підтверджується, що розроблений аспірантом кафедри КСМ Національного Авіаційного університету Балакіним Сергієм В'ячеславовичем під керівництвом доктора технічних наук, професора Жукова Ігоря Анатолійовича продукт "Аналізатор несанкціонованих дій в комп'ютерній мережі" використовується для підвищення ефективності роботи комп'ютерної мережі на підприємстві ДП"Компанія "Короп-пласт".

Програмний продукт "Аналізатор несанкціонованих дій в комп'ютерній мережі" виконано у повній відповідності до технічного завдання з виявлення несанкціонованих дій в комп'ютерній мережі і дозволяє визначати наступні параметри:

1) тип несанкціонованих дій;
2) час початку несанкціонованих дій;
3) вплив на систему і комп'ютерну мережу;
4) реакцію операційної системи на дані дії.

Проведені експериментальні дослідження розробленого продукту для виявлення несанкціонованих дій в комп'ютерній мережі на прикладах конкретних програм показали його ефективність. Технічний ефект від впровадження даного продукту заключається у можливості виявлення різного виду вторгнень з похибкою, що не перевищує 6%.

Економічний ефект залежатиме від дотримання правил інструкції користувача за умов впровадження програмного забезпечення для вирішення задачі підвищення надійності виявлення несанкціонованих дій і атак в комп'ютерній мережі.

Керівник відділу Козачок В.П. (ПІБ) "26" червня 2017 р.

ЗАТВЕРДЖУЮ
Директор ТОВ"Газбудсервіс"
Мисник М.І.
"26" червня 2017 р.

АКТ

про введення в експлуатацію способу

"Діагностування несанкціонованих дій в комп'ютерній мережі"

Цим актом підтверджується, що розроблений аспірантом кафедри КСМ Національного Авіаційного університету Балакіним Сергієм В'ячеславовичем під керівництвом доктора технічних наук, професора Жукова Ігоря Анатолійовича продукт "Діагностування несанкціонованих дій в комп'ютерній мережі" введений в експлуатацію та роботу в компанії ТОВ"Газбудсервіс".

Запропонована модель діагностування несанкціонованих дій в комп'ютерній мережі виконано у повній відповідності до технічного завдання і дозволяє визначати наступні параметри:

1) тип несанкціонованих дій;
2) час початку несанкціонованих дій;
3) вплив на систему і комп'ютерну мережу;
4) реакцію операційної системи на дані дії.

Проведені експериментальні дослідження розробленого продукту для діагностування несанкціонованих дій в комп'ютерній мережі на прикладах конкретних програм показали його високу ефективність.

Керівник відділу

ПОГОДЖЕНО
Проректор з науко
Національного ави

"02" 02 2017

АКТ

впровадження у навчальний процес Національного авіаційного університету результатів дисертаційної роботи **Балакіна Сергія В'ячеславовича** «Методи та засоби підвищення достовірності ідентифікації несанкціонованих дій та атак в комп'ютерній мережі», представленої на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.05 – комп'ютерні системи та компоненти.

Комісія у наступному складі: голова комісії – професор кафедри комп'ютерних систем та мереж, кандидат технічних наук, доцент Гузій Микола Миколайович, члени комісії: кандидат технічних наук, доцент кафедри комп'ютерних систем та мереж, доцент Андрєєв Володимир Ілліч, кандидат технічних наук, доцент кафедри комп'ютерних систем та мереж, доцент Проценко Микола Михайлович склали даний акт про те, що результати кандидатської дисертаційної роботи Балакіна Сергія В'ячеславовича «Методи та засоби підвищення достовірності ідентифікації несанкціонованих дій та атак в комп'ютерній мережі» впроваджені у навчальний процес та використовуються на кафедрі комп'ютерних систем та мереж з 2017-2018 н. р. при викладанні дисциплін «Телекомунікаційні технології комп'ютерних мереж» і «Архітектура комп'ютерів», а також у магістерських атестаційних роботах та дипломному проектуванні.

Голова комісії:
професор кафедри, кандидат технічних наук, доцент — М. М. Гузій

Члени комісії:
кандидат технічних наук, доцент — В. І. Андрєєв

кандидат технічних наук, доцент — М. М. Проценко

ПОГОДЖЕНО
Проректор з науко
Національного ав

"02" 02 2017 р.

АКТ

впровадження у навчальний процес Національного авіаційного університету результатів дисертаційної роботи **Балакіна Сергія В'ячеславовича** «Методи та засоби підвищення достовірності ідентифікації несанкціонованих дій та атак в комп'ютерній мережі», представленої на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.05 – комп'ютерні системи та компоненти.

Комісія у наступному складі: голова комісії – професор кафедри комп'ютерних систем та мереж, кандидат технічних наук, доцент Гузій Микола Миколайович, члени комісії: кандидат технічних наук, доцент кафедри комп'ютерних систем та мереж, доцент Андрєєв Володимир Ілліч, кандидат технічних наук, доцент кафедри комп'ютерних систем та мереж, доцент Проценко Микола Михайлович склали даний акт про те, що результати кандидатської дисертаційної роботи Балакіна Сергія В'ячеславовича «Методи та засоби підвищення достовірності ідентифікації несанкціонованих дій та атак в комп'ютерній мережі» впроваджені у навчальний процес та використовуються на кафедрі комп'ютерних систем та мереж з 2017-2018 н. р. при викладанні дисциплін «Телекомунікаційні технології комп'ютерних мереж» і «Архітектура комп'ютерів», а також у магістерських атестаційних роботах та дипломному проектуванні.

Назва розділів дисертаційної роботи, що впроваджуються	Форма впровадження	Результати впровадження
1	2	3
1. Аналіз стану питання і постановка завдань дослідження	Лекції, магістерські атестаційні роботи та дипломне проектування	На основі аналізу сучасних методів виявлення несанкціонованих дій в комп'ютерних мережах сформульовано необхідні критерії та вимоги для забезпечення своєчасного виявлення вторгнень в комп'ютерних мережах
2. Організація розпізнавання несанкціонованих дій засобами штучних імунних мереж	Лекції, магістерські атестаційні роботи та дипломне проектування, розрахунково-графічні роботи	Запропонована модель виявлення вторгнень в комп'ютерній мережі, який базується на використанні операторів штучних імунних мереж, дає можливість автономно розпізнавати невідомі вторгнення і мінімізувати хибні спрацювання
3. Організація діагностування несанкціонованих дій	Лекції, лабораторні роботи, магістерські атестаційні роботи, дипломне проектування, розрахунково-графічні роботи	Представлення моделі виявлення вторгнень в комп'ютерній мережі засобами інспектування стану системи на виникнення аномальної поведінки використовуванням дерева діагностування дає змогу відстежувати активність системи і симптомізувати дії користувача

Голова комісії:
професор кафедри, кандидат технічних наук, доцент М. М. Гузій

Члени комісії:
кандидат технічних наук, доцент В. І. Андрєєв

кандидат технічних наук, доцент М. М. Проценко

ДОДАТОК Б. Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації

1. Балакин С. В. Выявление компьютерных атак с помощью мониторинга сетевых объектов. *Технологический аудит и резервы производства*. Харьков, 2015. № 5-6(25). С.36-38. (Входить до міжнародних наукометричних баз Index Copernicus, РИНЦ, EBSCO Publishing, DOAJ).
2. Zhukov I. A., Balakin S.V. Detection of computer attacks using outliner. *Науковий журнал «Молодий вчений»*. К.: 2016. № 9(36). С. 91-93. (Входить до міжнародних наукометричних баз РИНЦ, ScholarGoogle, OAJI, CiteFactor, Research Bible, Index Copernicus).
3. Балакин С. В. Организация пресечения вторжений в компьютерные сети алгоритмами выявления изменений. *Вісник НТУ «ХПІ». Серія: Механіко-технологічні системи та комплекси.* Харків, 2017. № 20(1242). С.3-7. (Входить до міжнародної наукометричної бази OAJI).
4. Жуков И. А., Балакин С. В. Обнаружение компьютерних атак с помощью метода отклонений. *Радіоелектронні і комп'ютерні системи: наук. – техн. жур.* Харків, 2016. №5(79). С. 33-37. (Реферується наукометричними базами ВАК, Index Copernicus, INSPEC IDEAS).
5. Жуков И. А., Балакин С. В. Идентификация атак в компьютерной сети методом усредненного времени обращений. *Проблеми інформатизації та управління: зб. наук. праць.* К.: НАУ, 2015. № 2(50). С.65–69. (Реферується наукометричною базою Україніка наукова, входить до міжнародних наукометричних баз РИНЦ, ScholarGoogle).
6. Балакін С. В. Застосування штучних імунних систем при виявленні шкідливих программ в комп'ютерній мережі. *Проблеми інформатизації та управління: зб. наук. праць.* К.: НАУ, 2017. № 1-2(57-58). С. 61-68. (Реферується наукометричною базою Україніка наукова, входить до міжнародних наукометричних баз РИНЦ, ScholarGoogle).
7. Жуков И. А., Балакин С. В. Исследование эффективности метода обнаружения вторжений в компьютерные сети на основе искусственных иммуных систем. *Проблеми інформатизації та управління: зб. наук. праць.* К.: НАУ, 2017. № 3(59). С. 65-69. (Реферується наукометричною базою Україніка наукова, входить до міжнародних наукометричних баз РИНЦ, ScholarGoogle).
8. Спосіб запобігання комп'ютерним атакам у мережі за допомогою фільтрації вхідних пакетів: пат. 110330 Україна: МПК G06F 12/14. №201602196; заявл. 09.03.16; опубл. 10.10.16, Бюл. №19. 4 с.

9. Спосіб діагностування несанкціонованих дій в комп'ютерній мережі: пат. 123634 Україна: МПК G06F 12/14. №201702719; заявл. 23.03.17; опубл. 12.03.18, Бюл. №5. 4 с.

10. Балакин С. В. Методы и средства повишения достоверности идентификации несанкционированных воздействий и атак в компьютерной сети. *Комп'ютерні системи та мережні технології :* зб. тез доп. VIII міжнар. наук.-техн. конф., м. Київ, 16-18 квіт. 2015 р. Київ, 2015. С. 11-12.

11. Балакин С. В. Системы предотвращения атак в компьютерной сети на основе сигнатурных методов. *Комп'ютерні системи та мережні технології :* зб. тез доп. IX міжнар. наук.-техн. конф., м. Київ, 21-23 квіт. 2016 р. Київ, 2016. С. 12-13.

12. Балакин С. В. Средства диагностирования несанкционированных воздействий и атак в компьютерной сети. *Комп'ютерні системи та мережні технології :* зб. тез доп. X міжнар. наук.-техн. конф., м. Київ, 20-22 квіт. 2016 р. Київ, 2017. С. 13-14.

13. Balakin S. V. Traffic analysis for intrusion detection systems in telecommunication networks. *Політ. Сучасні проблеми науки :* зб. тез доп. XIV міжнар. наук.-практ. конф., м. Київ, 2-3 квіт. 2014. С. 59-60.

14. Балакин С. В. Оптимизация искусственных иммунных систем при идентификации несанкционированных сетевых воздействий. *Комп'ютерні системи та мережні технології :* зб. тез доп. XI міжнар. наук.-техн. конф., м. Київ, 19-21 квіт. 2018 р. Київ, 2018. С. 7-8.

Апробація результатів дисертації. Основні положення дисертаційної роботи доповідалися на XIV-й міжнародній науково-практичній конференції «Політ.Сучасні проблеми науки» (Київ, 2-3 квітня 2014 р.), на VIII міжнародній науково-технічній конференції «Комп'ютерні системи та мережні технології» (м. Київ, 16 – 18 квітня 2015 р.), міжнародній науково-технічній конференції «Cyber forum DESSERT 2016 B2S – S2B » (Чернівці, 18 – 23 травня 2016 р.), на IX міжнародній науково-технічній конференції «Комп'ютерні системи та мережні технології» (м. Київ, 21 – 23 квітня 2016 р.), на X-й міжнародній науково-технічній конференції «Комп'ютерні системи та мережні технології» (м. Київ, 20 – 22 квітня 2017 р.).

Публікації. За результатами досліджень опубліковано 14 наукових праць, у тому числі 7 статей у наукових фахових виданнях України [1-7], які включені до міжнародних наукометричних баз, 2 патенти України на корисну модель [8, 9] та 5 тез доповідей в збірниках матеріалів конференцій [10-14].