Міністерство освіти і науки України
Сумський державний університет

Сучасні інформаційні технології в кібербезпеці

Монографія

За редакцією В. К. Ободяка, І. В. Шелехова

Рекомендовано вченою радою Сумського державного університету

Суми
Сумський державний університет
2021

УДК 004.056.5
С 91

Рецензенти:
О. І. Цопа – доктор технічних наук, професор (Харківський національний університет радіоелектроніки);
А. С. Опанасюк – доктор фізико-математичних наук, професор (Сумський державний університет)

Рекомендовано до видання
вченою радою Сумського державного університету
як монографія
(протокол № 9 від 12 лютого 2021 року)

С 91 **Сучасні** інформаційні технології в кібербезпеці : монографія / А. С. Довбиш, В. К. Ободяк, І. В. Шелехов та ін. ; за ред. В. К. Ободяка, І. В. Шелехова. – Суми : Сумський державний університет, 2021. – 348 с.
ISBN 978-966-657-858-0

У монографії розглянуто питання квантової криптографічної технології, управління ризиками інформаційної безпеки, виявлення шкідливого програмного забезпечення, стандартизації та термінології кібербезпеки і підготовки студентів за спеціальністю «Кібербезпека». Значну увагу приділено вирішенню завдання інформаційно-екстремального синтезу системи виявлення кібератак у рамках розробленого авторами методу машинного навчання. Викладений у монографії матеріал може бути корисним фахівцям, аспірантам і студентам спеціальності «Кібербезпека».

УДК 004.056.5

ISBN 978-966-657-858-0

ЗМІСТ

ПЕРЕДМОВА

Сьогодні відбувається перехід інформаційного суспільства в його вищу соціально-економічну фазу – так зване знанняорієнтоване суспільство, в якому більша половина світового валового продукту виробляється за допомогою інтелектуальних інформаційних технологій. У той самий час останні науково-технічні досягнення в інформаційній галузі стають доступними злочинним угрупованням, про що свідчить стрімке збільшення кіберзагроз, спрямованих на несанкціонований доступ до інформаційних ресурсів державних установ, бізнес-структур і приватних осіб. Безпека інформації стає важливою складовою функціональної ефективності інформаційно-телекомунікаційної системи. Водночас варто усвідомлювати, що оскільки організовані кіберзлочинці на високому професійному рівні володіють сучасними апаратно-програмними засобами інформаційно-комунікаційних технологій, то основними шляхами нейтралізації їх злочинної діяльності є розроблення і використання в комплексних системах захисту інформації нових перспективних інтелектуальних інформаційних технологій аналізу даних. У зв'язку з цим фахівцям із кібербезпеки буде корисним ознайомитися з викладеним у монографії перспективним напрямом інформаційного синтезу інтелектуальної комп'ютерно-інтегрованої системи виявлення атак у рамках розробленої науковою школою кафедри комп'ютерних наук Сумського державного університету так званої інформаційно-екстремальної інтелектуальної технології аналізу даних, що має ряд важливих переваг перед іншими технологіями Data Mining.

У монографії також розглядаються питання пост-квантової криптографії на основі математичних структур

логарифмічних підписів і покриттів кінцевих груп.

Окремий розділ присвячено оцінюванню кіберзагроз та керуванню захищеністю шляхом використання апарату аналізу та управління ризиками. Запропоновано механізм оцінювання ризиків на основі нечітких нейронних мереж.

Значна увага в монографії приділена як методам створення завадозахищеного програмного забезпечення, так і методам розпізнавання шкідливих програм, зокрема, в пристроях Інтернету речей із застосуванням штучних нейронних мереж. Крім того, сформовано рекомендації щодо розроблення національної методології оцінювання кіберзахищеності, що дозволить максимально повно та якісно оцінювати захищеність без урахування різноманітності програмного забезпечення інформаційно-комунікаційних систем.

У монографії можна виділити окремий блок, присвячений упровадженню розроблених викладацьким складом Сумського державного університету інноваційних технологій підготовки фахівців за спеціальністю «Кібербезпека».

У цілому колективна монографія відбиває широке коло завдань кібербезпеки згідно з професійною спрямованістю авторів і може становити інтерес як для фахівців, так і для аспірантів та студентів спеціальності 125 «Кібербезпека».

А. С. Довбиш

РОЗДІЛ 1
ОСНОВИ ІНФОРМАЦІЙНО-ЕКСТРЕМАЛЬНОГО СИНТЕЗУ АВТОМАТИЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ КІБЕРЗАХИСТОМ

А. С. Довбиш, В. К. Ободяк, І. В. Шелехов, Д. В. Великодний

Вступ

Цифризація всіх галузей соціально-економічної сфери та розвиток інформаційних технологій одночасно обумовлюють зростання нових загроз національній та економічній безпеці України. Поряд із катастрофами природного і техногенного походження невпинно збільшуються кількість та потужність кібератак, умотивованих злочинними інтересами окремих держав, груп та осіб. Тому в Стратегії кібербезпеки України пріоритетне місце займає створення національної системи кібербезпеки як складової системи забезпечення національної безпеки України [1]. Водночас, як зазначено в Стратегії, гарантування безпеки й сталого функціонування електронних комунікацій та державних електронних інформаційних ресурсів повинні бути складовими державної політики у сфері розвитку інформаційного простору та становлення інформаційного суспільства в Україні. Одним з основних напрямів досягнення цієї мети є створення комп'ютерно-інтегрованої в інформаційно-комунікаційну систему (ІКС) автоматизованої системи керування кіберзахистом (СККЗ) для своєчасного виявлення, запобігання та нейтралізації кіберзагроз. Також СККЗ розглядають як складову комплексної системи захисту інформації (КСЗІ). У Законі України «Про основні засади забезпечення кібербезпеки України» [2] кіберзахист розглядають як сукупність

організаційних, правових, інженерно-технічних заходів, а також заходів криптографічного і технічного захисту інформації, спрямованих на запобігання кіберінцидентам, виявлення та захист від кібератак, ліквідацію їх наслідків, відновлення сталості та надійності функціонування комунікаційних, технологічних систем.

Сучасна СККЗ є складною комп'ютерно-інтегрованою системою, що неперервно контролюється, здатна оперативно реагувати на кібератаки та несанкціоновані дії, накопичувати знання про способи протидії, виявлення і реагування на атаки і несанкціоновані дії та використовувати їх для забезпечення надійного захисту ІКС різного призначення. Для ІКС виділяють три рівні кіберзахисту:

1) технічні засоби інформації;

2) засоби захисту, які забезпечують моніторинг поточного стану ІКС та керування захистом системи шляхом виявлення атак та несанкціонованих дій і відповідної адаптації до них стратегій безпеки і компонентів КСЗІ;

3) нормативно-правовий рівень.

Перший рівень захисту складається з технічних засобів, що дозволяють реалізувати функції ідентифікації та аутентифікації, криптографічного захисту, розмежування доступу, контролю цілісності, реєстрації та обліку, міжмережевого екранування тощо.

Другий рівень на цей час перебуває на стадії розвитку і орієнтований на використання сучасних інтелектуальних інформаційних технологій аналізу даних для вирішення проблеми захисту інформації.

Водночас підвищення функціональної ефективності СККЗ істотно залежить від можливостей інтелектуальних інформаційних технологій забезпечувати високо достовірний та оперативний аналіз великих обсягів

даних, що відбувається за умови довільних початкових умов процесів захисту інформації й істотного перетину класів розпізнавання, які характеризують поточні функціональні стани ІКС.

Нормативно-правовий рівень є прерогативою державних та юридичних органів і так само перебуває на стадії неперервного розвитку та удосконалення.

У розділі 1 автори увагу концентрують на другому рівні кіберзахисту, забезпечення якого на цей час є нагальним завданням через різке збільшення кіберзагроз, обумовлене потраплянням до злочинних угрупувань сучасних технічних засобів та новітніх інформаційно-комунікаційних технологій.

Зрозуміло, що створена автоматизована СККЗ із високою функціональною ефективністю набуває стратегічного значення, стає національним надбанням країни-розробника, а можливість її придбання іншими країнами обумовлена не лише комерційними міркуваннями, а насамперед політичними рішеннями. Водночас єдиним шляхом вирішення цієї надскладної проблеми є застосуванням сучасних інтелектуальних інформаційних технологій аналізу даних на основі машинного навчання та розпізнавання образів.

Ураховуючи складність поставленого завдання автори розглядають викладені в розділі наукові результати як вступ до інформаційного синтезу автоматизованої СККЗ. Водночас метод дослідження реалізовано в рамках власної так званої інформаційно-екстремальної інтелектуальної технології (ІЕІ-технології) аналізу даних, що ґрунтується на принципі максимізації інформаційної спроможності системи в процесі машинного навчання.

1.1. Системний підхід та методологічні принципи системного аналізу системи керування кіберзахистом

Розвиток комп'ютерно-інтегрованих технологій обумовив появу нового класу так званих складних систем, до яких відносять і СККЗ, здатних функціонувати за умови невизначеності даних через вплив зовнішніх і внутрішніх випадкових та неконтрольованих факторів. Особливість цього класу систем полягає в непридатності класичних математичних методів до їх моделювання. Цей факт сприяв посиленому розвитку методів аналізу складних систем у рамках системного підходу.

Системний підхід – напрям ***методології*** досліджень, що полягає в дослідженні об'єкта як цілісної множини елементів у сукупності відношень, зв'язків між ними і властивостей. Методологія системного підходу базується на таких основних принципах, які найбільш сконцентровано викладено в працях [3, 4]:

1) сукупності елементів системи, що розглядається, як одне ціле, а не просте об'єднання елементів;

2) властивостях системи, які не є сумою властивостей її елементів. Тим самим постулюється можливість того, що система має особливі властивості, яких може і не бути в окремих елементах;

3) складових загальної ефективності, насамперед таких як функціональна й економічна ефективності, надійність, що залежать від умов побудови і функціонування системи. Оскільки на параметри функціонування системи накладаються обмеження, то третій принцип системного аналізу можна сформувати як максимізацію складових загальної ефективності. Таким чином, цей принцип установлює органічну єдність аналізу та синтезу системи, що досліджується;

4) системі, що є частиною (підсистемою) деякої більш загальної системи.

Вищенаведені чотири принципи доцільно розглядати як директивні принципи системного аналізу, з якими інші методологічні принципи не повинні суперечити. До таких методологічних принципів, наприклад, можна віднести:

1) принцип кінцевої мети, який встановлює абсолютний пріоритет кінцевої (глобальної) мети;

2) принцип ієрархічності, який полягає в тому, що вивчення складних об'єктів повинно базуватися на уявленні про ієрархічність їх структури. Водночас об'єктом аналізу може бути як сама система, так і її атрибути, зокрема і вхідні дані;

3) принцип декомпозиції, що полягає в розкладанні системи (проблеми) на окремі підсистеми (завдання);

4) принцип динамічності, який полягає в тому, що системний підхід вимагає розглядати аналізований об'єкт у його розвитку на всіх етапах життєвого циклу;

5) принцип структуризації, що дозволяє аналізувати елементи системи, їх взаємозв'язки і дані в рамках конкретної організаційної структури.

Оскільки теорія систем постійно розвивається, то наведений перелік принципів не претендує на повноту.

1.2. Основні властивості системи керування кіберзахистом

Розглянемо основні властивості СККЗ, комп'ютерно-інтегрованої в ІКС.

1. Функціональна ефективність – це властивість, що характеризує здатність СККЗ виконувати поставлене перед нею основне завдання.

2. Функціональна стійкість до кіберзагроз – це властивість СККЗ повертатися до попереднього робочого

функціонального стану після припинення дії збурювальних факторів, зокрема у вигляді атак або несанкціонованих дій.

3. Керованість – це існування необмеженого керування, що може перевести СККЗ із довільного початкового функціонального стану в будь-який інший заданий стан за кінцевий інтервал часу.

4. Спостережуваність – це можливість визначення поточного функціонального стану СККЗ шляхом аналізу вхідної, робочої та вихідної інформації за заданого керувального сигналу за кінцевий період часу.

З точки зору теоретико-інформаційного підходу умовою спостережуваності СККЗ є наявність умовної кількості інформації, яку вона одержує, аналізує й передає. тобто повинна виконуватися нерівність

$$I = H - H(\gamma) > 0,$$

або

$$0 < I \le H,$$

де H – апріорна (безумовна) ентропія, що характеризує невизначеність даних на вході ІТС і визначається за формулою

$$H = -\sum_{l=1}^{M} p(\gamma_l)\log_2 p(\gamma_l), \qquad (1)$$

де $p(\gamma_l)$ – безумовна ймовірність прийняття гіпотези γ_l ;

M – кількість гіпотез;

$H(\gamma)$ – апостеріорна ентропія, що характеризує залишкову невизначеність після прийняття СККЗ класифікаційного рішення і визначається як

$$H(\gamma) = -\sum_{l=1}^{M}\sum_{m=1}^{M} p(\gamma_l)\, p(\mu_m / \gamma_l) \log_2 p(\mu_m / \gamma_l), \quad (2)$$

де $p(\mu_m / \gamma_l)$ – апостеріорна умовна ймовірність прийняття рішення μ_m за умови, що прийнята гіпотеза γ_l.

СККЗ може втратити спостережуваність як за умови впливу зовнішніх, наприклад, наслідків кібератаки, так і внутрішніх збурювальних факторів. Прикладом внутрішнього фактору може бути випадок невиправданого збільшення або зменшення поля контрольних допусків на ознаки розпізнавання, що має наслідком збіг структурованих векторів ознак різних класів розпізнавання, які характеризують відповідні профілі трафіку. Тобто спостережуваність інтелектуальної СККЗ можна трактувати як властивість системи розрізняти класи розпізнавання.

5. Інформаційна спроможність, що визначається кількісними інформаційними характеристиками СККЗ.

6. Точність, що визначається через точнісні характеристики відповідної системи оцінювань рішень. Наприклад, для двоальтернативної системи оцінювань достовірність СККЗ характеризується повною ймовірністю правильного прийняття класифікаційних рішень:

$$P_t = p_1 D_1 + p_2 D_2,$$

де p_1 – безумовна ймовірність прийняття основної гіпотези γ_1;

p_2 – безумовна ймовірність прийняття альтернативної гіпотези γ_2;

D_1 – перша достовірність;

D_2 – друга достовірність;

і повною ймовірністю неправильного прийняття класифікаційних рішень:

$$P_f = p_1\alpha + p_2\beta,$$

де α – помилка першого роду;

β – помилка другого роду.

Оскільки гіпотези γ_1 і γ_2 складають повну групу подій, то має місце

$$P_t + P_f = 1.$$

7. Надійність СККЗ. Це властивість системи забезпечувати виконання заданих функцій, зберігаючи в часі функціональну ефективність системи в заданих межах.

8. Стабільність СККЗ**.** Це властивість системи зберігати незмінними свої характеристики в процесі експлуатації.

9. Енерговитрати**,** пов'язані з функціонуванням СККЗ.

10. Вартість як сукупні витрати на всіх етапах життєвого циклу системи.

Зрозуміло, що властивості СККЗ не обмежуються вищенаведеними, оскільки теорія проєктування систем цього класу перебуває в стані невпинного розвитку.

1.3. Методи виявлення вторгнень

Кібератака (далі атака) – спрямовані (навмисні) дії в кіберпросторі, які здійснюються за допомогою засобів

електронних комунікацій (враховуючи інформаційно-комунікаційні технології, програмні, програмно-апаратні засоби, інші технічні й технологічні засоби та обладнання) з метою одержати несанкціонований доступ до інформаційних ресурсів системи (мережі) або порушити її нормальне функціонування [2]. Незважаючи на існування різноманітних механізмів захисту інформації, статистика останніх років свідчить про різке зростання кількості злочинів, пов'язаних з порушенням конфіденційності, цілісності та доступності інформації. Відновлення функціонування ІКС після встановленої атаки характеризується відносно великою тривалістю в часі, що призводить до великих економічних збитків і дозволяє зловмисникам замітати сліди. Цим пояснюється значна увага, що приділяється останніми роками, наприклад, квантовій криптографії, технології завадозахищеного програмування блок-чейн, яка вже дозволяє скоротити часовий період незахищеності мережі до 10 хвилин тощо. Особливі надії покладаються на розвиток нового напряму у сфері захисту інформації, пов'язаного із створенням комп'ютерно-інтегрованих систем виявлення атак (СВА) (Intrusion Detection Systems, IDS). Призначенням таких систем є аналіз інформації, яка надходить із різних хостів ІКС з метою виявлення як спроб, так і реальних вторгнень. Проте СВА розглядається як обов'язкова підсистема СККЗ.

Оскільки будь-яка атака на систему може бути виявлена в ході аналізу мережевого трафіку або системних ресурсів, то відповідно СВА діляться на два рівні [5]: мережевий (Network Intrusion Detection Systems, NIDS) і системний, або система виявлення атак на базі хосту (Host Intrusion Detection Systems, HIDS). СВА вміщує кілька детекторів, функціями яких є безпосереднє виявлення атак, ґрунтуючись на певних умовах виявлення. Серед

базових методів, що використовуються для вирішення цього завдання, зазвичай виділяють дві основні групи:

1) методи сигнатурного аналізу;

2) методи виявлення аномалій.

Обидва методи базуються на припущенні, що кожну атаку можна описати за допомогою певного набору правил або за допомогою деякої формальної моделі. Виявлення атак за методом сигнатурного аналізу зводиться до порівняння поточних дій користувача або вхідного / вихідного трафіку з відомими шаблонами (сигнатурами) атак, що зберігаються в базі знань. Наприклад, велика кількість TCP-з'єднань з різними портами свідчить про те, що відбувається скануванням TCP-портів. Інший приклад мережевої атаки – атака SYN Flood, коли зловмисник закидає вебсервер великою кількістю SYN-пакетів, які ініціалізують з'єднання, викликаючи тим самим відмову в обслуговуванні (Denial of Service, DoS) легального користувача. Перевагою методів сигнатурного аналізу є висока ефективність під час виявлення відомих атак і незначна кількість «помилкових тривог», тобто помилок першого роду. Недоліками таких методів є:

– необхідність постійно поповнювати базу знань сигнатурами нових атак, тому що в іншому разі виникає потенційна небезпека «пропуску атаки», що характеризується помилкою другого роду;

– великі обчислювальні витрати.

Суть методів виявлення аномалій полягає в тому, що СВА має певний набір знань про нормальний функціональний стан ІКС. Будь-які відхилення від такого стану вона ідентифікує як аномальну поведінку системи.

Під час побудови профілю користувача або мережевого трафіку потрібно брати до уваги такі показники [5], як:

– середня кількість записів аудиту, оброблюваних для елемента системи, що захищають за одиницю часу;

– розподіл різних типів дій, пов’язаних із доступом до файлів, операціями введення-виведення тощо;

– відносна частота реєстрації в системі (логінів) із кожного фізичного місця перебування користувача;

– кількість файлів, до яких звертається користувач за певний інтервал часу;

– кількість невдалих спроб входу в систему тощо.

Перевагою методів виявлення аномалій є здатність розпізнавати нові типи атак.

Недоліки методів виявлення аномалій:

– потребують тривалого машинного навчання;

– характеризуються невисокою оперативністю і великими обчислювальними витратами;

– часто призводять до помилок першого роду, тобто «помилкових невиправданих тривог».

Отже, згідно з результатами порівняльного аналізу переваг і недоліків вищенаведених методів виявлення атак перспективним напрямом підвищення функціональної ефективності СВА у складі СККЗ є їх побудова на основі ідей та методів машинного навчання й розпізнавання образів. На рисунку 1.1 зображено структурну схему СККЗ, що навчається.

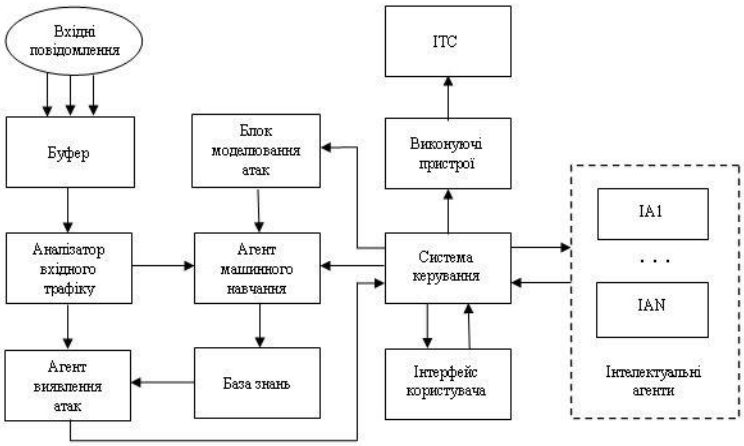


Рисунок 1.1 – Структурна схема інтелектуальної СККЗ

Наведена СККЗ має централізовано-розподілену багатоагентну структуру. Джерелом інформації для СККЗ є вхідний / вихідний та внутрішні системні трафіки, аналізовані інтелектуальними агентами (ІА) хостів. Вхідні повідомлення надходять в аналізатор вхідного трафіку, що формує вектори структурованих ознак розпізнавання. У разі функціонування СККЗ в режимі самонавчання, вектори ознак вхідного трафіку із заданим часовим періодом надходять на входи агентів машинного навчання та виявлення атак.

Агент машинного навчання способом кластеризації вхідних даних визначає їх належність до навчальної матриці для попередньо сформованого алфавіту класів розпізнавання. Якщо з'являється новий кластер, автоматично формується його навчальна матриця, що після досягнення репрезентативного обсягу додається до основної навчальної матриці, і запускається алгоритм машинного навчання. Водночас основним завданням агента машинного навчання є побудова детермінованих

вирішальних правил. Оскільки загалом навчальна матриця є нечіткою, то агент машинного навчання дефазифікує нечіткі дані.

Побудовані за результатами машинного навчання вирішальні правила надходять в базу знань, із якої вони за запитами передаються агенту виявлення атак, що класифікує поточний вхідний трафік і передає в блок керування відповідне повідомлення. Блок керування залежно від змісту повідомлення формує керувальні команди й через виконуючі пристрої може впливати на інфраструктуру та програмне забезпечення ІКС для усунення кіберзагрози. Крім того, до блока керування надходить інформація від інтелектуальних агентів хостів для формування відповідних керувальних команд. Щоб накопичити вирішальні правила для розпізнавання аномальної поведінки, СККЗ має можливість здійснювати машинне навчання способом використання результатів моделювання атак.

Блок моделювання атак є підсистемою СККЗ і містить симулятор ІКС, базу даних, що накопичує моделі (сигнатури) атак, та апаратно-програмні засоби генерування атак.

На рисунку 1.2 зображено структурну схему інтелектуального агента, що так само має власну комп'ютерно-інтегровану СВА на хост.

Рисунок 1.2 – Структурна схема інтелектуального агента

Згідно з рисунком 1.2 блоки інтелектуального агента хоста мають таке саме призначення, як і на рисунку 1.2. Відмінність полягає лише у функціях блоку керування агента, обмежених завданнями кіберзахисту відповідного хоста.

Останніми роками підвищується інтерес до побудови комбінованих СВА, що поєднують у собі переваги двох зазначених вище напрямків виявлення атак. Подібні можливості на сьогодні реалізують способом

застосування для аналізу атак нейромережевих технологій, що дають такі можливості [6, 7]:

– виявлення раніше невідомих атак;

– аналізу неповних і спотворених даних, фільтрації даних;

– паралельного (багатопроцесорного) оброблення даних.

Недоліками штучних нейронних мереж (ШНМ) є те, що вони за умови варіативної невизначеності даних можуть давати різну збіжність. Водночас основний недолік нейроподібних структур – залежність їх функціональної ефективності, основними складовими якої є достовірність та оперативність, від багатовимірності простору ознак й алфавіту класів розпізнавання під час виявлення атак. Крім того, структурні методи машинного навчання негнучкі в разі перенавчання системи через розширення словника ознак та алфавіту класів розпізнавання.

У праці [8] як один із напрямів підвищення достовірності й оперативності розпізнавання функціонального стану ІКС розглядають технології нечітких систем. Водночас варто зазначити, що галуззю застосування нечітких методів подання та виведення знань є системи з якісними шкалами вимірювання ознак розпізнавання. У разі виявлення атак здебільшого використовують кількісні шкали вимірювання. Оскільки функція належності Заде є аналогом функції щільності ймовірностей, методи багатовимірного статистичного аналізу [9] доцільніші для машинного навчання й розпізнавання з огляду на їх розвинений математичний апарат.

Одним із перспективних способів аналізу й синтезу здатних навчатися СККЗ є використання ідей і застосування методів ІЕІ-технології аналізу даних, що

ґрунтується на максимізації інформаційної спроможності СВА у процесі машинного навчання [10, 11].

1.4. Основні принципи та положення інформаційно-екстремальної інтелектуальної технології аналізу даних

Згідно з працею [10] основна ідея машинного навчання в рамках ІЕІ-технології аналізу даних полягає в трансформації апріорного узагальненого нечіткого розбиття простору ознак у чітке розбиття класів розпізнавання способом оптимізації параметрів функціонування системи. Водночас здійснюється цілеспрямований пошук глобального максимуму багатоекстремальної функції статистичного інформаційного критерію в робочій (допустимій) області її визначення з одночасним відновленням оптимальних роздільних гіперповерхонь, що будуються в радіальному базисі бінарного простору ознак розпізнавання.

Методи інформаційно-екстремального машинного навчання базуються, крім відомих принципів системного аналізу, на таких специфічних принципах:

– максимізації інформації, обґрунтованому екстремальністю сенсорного сприйняття образу, що експериментально доведено вченими-фізіологами. Цей принцип реалізують способом уведення додаткових інформаційних обмежень, що збільшують різноманітність класифікованих об’єктів;

– дуальності, що полягає в реалізації на етапі апріорного моделювання простих алгоритмів за умови їх цілеспрямованого уточнення способом поглиблення машинного навчання для наближення вирішальних правил до безпомилкових за навчальною матрицею;

– апріорної недостатності обґрунтування гіпотез (принцип Бернуллі – Лапласа), згідно з яким за умов

апріорної невизначеності даних доцільно розглядати апріорні гіпотези однаково ймовірними, тобто рішення приймаються системою за найгірших у статистичному розумінні умов;

– рандомізації вхідних даних, що дозволяє досліджувати детерміновано-статистичні характеристики процесу;

– редукції даних, що обумовлює необхідність оптимізації в інформаційному розумінні словника ознак розпізнавання способом видалення з нього неінформативних та ознак, які заважають;

- відкладених рішень О. Г. Івахненка, що полягає в необхідності повторення процедур машинного навчання для досягнення мети побудови безпомилкових за навчальною матрицею вирішальних правил;

– зовнішнього доповнення, що обґрунтовує необхідність використання навчальної або контрольної (екзаменаційної) вибірки для оцінювання функціональної ефективності машинного навчання

Вирішальні правила в процесі оптимізації параметрів машинного навчання в рамках IEI-технології будують згідно з принципом відкладених рішень О. Г. Івахненка за багатоциклічною ітераційною процедурою пошуку максимального граничного значення усередненого за алфавітом класів розпізнавання інформаційного критерію оптимізації:

$$g_{\xi}^{*} = \arg\max_{G_{\xi}}\{\max_{G_{\xi-1}}\{\ldots\{\max_{G_1 \cap G_E}\frac{1}{M}\sum_{m=1}^{M}E_m\}\ldots\}\}, \qquad (3)$$

де E_m – інформаційний критерій оптимізації параметрів навчання системи розпізнавати реалізації класу X_m^o;

G_ξ – допустима область значень ξ-ї ознаки розпізнавання;

G_E – допустима область визначення функції інформаційного критерію оптимізації параметрів машинного навчання.

Водночас на алгоритм інформаційно-екстремального машинного навчання (3) накладають обмеження

$$(\forall X_m^o \in \tilde{\Re}^{|M|})\left[X_m^o \neq \varnothing\right], \qquad (4)$$

де $\tilde{\Re}^{|M|}$ – розбиття простору ознак на класи розпізнавання, потужність якого $Card\,\tilde{\Re} = M$;

$$(\exists X_k^o \in \tilde{\Re}^{|M|})(\exists X_l^o \in \tilde{\Re}^{|M|})\left[X_k^o \neq X_l^o \rightarrow X_k^o \cap X_l^o \neq \varnothing\right]; \qquad (5)$$

$$(\forall X_k^o \in \tilde{\Re}^{|M|})(\forall X_l^o \in \tilde{\Re}^{|M|})\left[X_k^o \neq X_l^o \rightarrow KerX_k^o \cap KerX_l^o \neq \varnothing\right], \qquad (6)$$

де $KerX_k^o$ – ядро класу розпізнавання X_k^o;

$Ker\,X_l^o$ – ядро класу розпізнавання X_l^o, найближчого до класу розпізнавання X_k^o;

$$(\forall X_k^o \in \tilde{\Re}^{|M|})(\forall X_l^o \in \tilde{\Re}^{|M|})\left[\begin{array}{l} X_k^o \neq X_l^o \rightarrow [(d_k^* < d(x_k \oplus x_l))\,\& \\ \&\,(d_l^* < d(x_k \oplus x_l))] \end{array}\right] \qquad (7)$$

де d_l^* – оптимальний радіус контейнера класу розпізнавання X_l^o;

$d(x_k \oplus x_l)$ – кодова відстань між вектором x_k, усередненим за ансамблем векторів ознак класу

розпізнавання X_k^o, і відповідним вектором x_l класу розпізнавання X_l^o;

$$\bigcup_{X_m^o \in \ddot{Y}} X_m^o \subseteq \Omega_B, \qquad (8)$$

де $\Omega_Б$ – бінарний простір Хеммінга.

У виразах (6)–(8) $k \neq l$ і $k,l,m = \overline{1,M}$.

Глибина інформаційно-екстремального машинного навчання характеризується кількістю параметрів функціонування системи, оптимізованих за інформаційним критерієм. Водночас внутрішній цикл процедури (3) реалізує так званий базовий алгоритм інформаційно-екстремального машинного навчання, що оптимізує геометричні параметри контейнерів класів розпізнавання.

Основна ідея машинного навчання методами ІЕІ-технології, так само як і у ШНМ – адаптація вхідного математичного опису системи розпізнавання до максимальної повної ймовірності класифікаційних рішень. Принципова відмінність методів інформаційно-екстремального машинного навчання від нейроподібних структур полягає в тому, що їх розробляють у рамках функціонального підходу до моделювання когнітивних процесів, притаманних людині під час формування та прийняття класифікаційних рішень, тобто вони безпосередньо моделюють механізм природного інтелекту. Водночас процес машинного навчання розглядають як оптимізацію параметрів системи розпізнавання, що впливають на її функціональну ефективність. Такі параметри називають параметрами машинного навчання. Як критерій оптимізації в методах ІЕІ-технології можна використовувати будь-яку

статистичну інформаційну міру різноманітності аналізованих об'єктів. Якщо в ШНМ глибина машинного навчання обумовлена кількістю прихованих шарів, то в методах ІЕІ-технології її визначають за кількістю параметрів машинного навчання, що оптимізуються. Водночас достатню глибину інформаційно-екстремального машинного навчання визначають згідно з принципом відкладених рішень О. Г. Івахненка за умови досягнення граничного максимального значення усередненого за алфавітом класів розпізнавання інформаційного критерію оптимізації. Вирішальні правила будують за одержаними в процесі машинного навчання оптимальними геометричними параметрами контейнерів класів розпізнавання, що відновлюються в радіальному базисі бінарного простору ознак Хеммінга. Побудова вирішальних правил у рамках геометричного підходу робить їх майже інваріантними до багатовимірності простору ознак розпізнавання, тому що сучасні комп'ютерні комплекси можуть обробляти двійкові вектори, що містять 2^{85} ознак розпізнавання. Крім того, такі вирішальні правила характеризуються високою оперативністю прийняття класифікаційних рішень у разі функціонування СККЗ в режимі моніторингу, що є важливим фактором для виявленні атак.

Отже, використання ідей і застосування методі ІЕІ-технології відкриває широкі перспективи для вирішення проблеми інформаційного синтезу здатної навчатися автоматизованої СККЗ як складової КСЗІ.

1.5. Формалізована постановка задачі інформаційного синтезу системи виявлення атак

Розглянемо формалізовану постановку задачі інформаційного синтезу здатної навчатися СВА в рамках ІЕІ-технології. Нехай дано алфавіт $\{X_m^o \mid m=\overline{1,M}\}$ класів розпізнавання, що характеризують можливі профілі трафіку ІКС, і навчальну матрицю типу «об'єкт – властивість» $||y_{m,i}^{(j)}||, i=\overline{1,N},\ j=\overline{1,n}$, де N, n – кількість ознак розпізнавання й структурованих векторів ознак (далі – реалізації) класів розпізнавання відповідно. Водночас рядок матриці $\{y_{m,i}^{(j)} \mid i=\overline{1,N}\}$ визначає j-ту реалізацію, а стовпчик $\{y_{m,i}^{(j)} \mid j=\overline{1,n}\}$ – навчальну випадкову вибірку значень i-ї ознаки. Відомо, що концепція ІЕІ-технології полягає в перетворенні вхідної навчальної матриці Y на робочу бінарну матрицю X, що способом допустимих перетворень у процесі машинного навчання адаптується до максимальної повної ймовірності прийняття правильних класифікаційних рішень. Тому для бінарного простору Хеммінга задамо множину $\{g_m\}$ структурованих векторів параметрів функціонування, що впливають на функціональну ефективність машинного навчання СВА. У подальшому такі параметри функціонування називатимемо параметрами машинного навчання. Вектор параметрів машинного навчання СВА розпізнавати реалізації класу X_m^o наведемо як структуру

$$g = < g_1,...,g_{\xi_1},...,g_{\Xi_1},\ f_1,...,f_{\xi_2},...,f_{\Xi_2} >,\ \Xi_1+\Xi_2=\Xi, \qquad (9)$$

де $< g_1,...,g_{\xi_1},...,g_{\Xi_1} >$ – генотипні параметри функціонування, що впливають на параметри розподілу реалізацій класу розпізнавання;

$< f_1,..., f_{\xi_2},..., f_{\Xi_2} >$ – фенотипні параметри функціонування, що впливають на геометрію контейнерів класів розпізнавання, відновлюваних у радіальному базисі простору ознак.

Водночас відомі обмеження на відповідні параметри машинного навчання:

$$R_{\xi_1}(g_1,..., g_{\xi_1},..., g_{\Xi_1}) \le 0, \quad R_{\xi_2}(f_1,..., f_{\xi_2},..., f_{\Xi_2}) \le 0.$$

Необхідно:

1) визначити оптимальні значення параметрів машинного навчання (9) $\{g_\xi^* \mid \xi = \overline{1, \Xi_1 + \Xi_2}\}$, що забезпечують максимум усередненого за алфавітом класів розпізнавання інформаційного критерію

$$\overline{E}^* = \frac{1}{M} \sum_{m=1}^{M} \max_{G_E \cap \{k\}} E_m^{(k)}, \qquad (10)$$

де $E_m^{(k)}$ – інформаційний критерій оптимізації параметрів машинного навчання СВА розпізнавати реалізації класу X_m^o, значення якого обчислено на k-му кроці машинного навчання;

G_E – допустима область визначення функції інформаційного критерію оптимізації, що далі називатимемо робочою областю;

$\{k\}$ – упорядкована множина кроків машинного навчання (відновлення контейнерів класів розпізнавання в радіальному базисі дискретного простору ознак);

2) для апріорно класифікованого нечіткого розбиття $\tilde{\Re}^{|M|}$ побудувати шляхом допустимих перетворень у субпарацептуальному бінарному просторі ознак розпізнавання Хеммінга оптимальне (тут і далі в

роботі в інформаційному розумінні) чітке розбиття класів розпізнавання $\mathfrak{R}^{|M|}$, на основі якого сформувати безпомилкові за навчальною матрицею вирішальні правила;

3) на етапі екзамену для перевірки функціональної ефективності машинного навчання прийняти рішення про належність реалізації образу, що розпізнається, до одного з класів заданого алфавіту $\{X_m^o\}$.

Отже, завдання інформаційного синтезу здатної навчатися СККЗ зводиться до оптимізації в процесі інформаційно-екстремального машинного навчання параметрів функціонування (9) за інформаційним критерієм (10) та прийняття в режимі екзамену класифікаційного рішення за побудованими на етапі навчання вирішальними правилами.

1.6. Інформаційні критерії оптимізації параметрів машинного навчання

Центральним питанням інформаційного синтезу здатної навчатися системи розпізнавання є оцінювання функціональної ефективності процесу машинного навчання, основними критеріями якої є достовірність та оперативність класифікаційних рішень. Як критерії оптимізації параметрів машинного навчання в методах ІЕІ-технології можуть використовувати різні критерії, що задовольняють такі властивості інформаційної міри:

– інформаційна міра є дійсна і знакододатна функція від імовірності;

– кількість інформації для детермінованих подій ($p_i = 1$ або $p_i = 0$) дорівнює нулю;

– інформаційна міра має екстремум за значення ймовірності $p_i = \frac{1}{m}$, де m – кількість якісних ознак розпізнавання;

– сумісна інформаційна міра двох незалежних повідомлень дорівнює сумі їх відповідних інформаційних мір.

Серед інформаційних мір для оцінювання функціональної ефективності СВА, що навчається, перевагу варто віддавати статистичним логарифмічним критеріям, що дозволяють працювати з порівняно малими навчальними вибірками. Серед таких критеріїв найчастіше використовуються ентропійні міри Шеннона та інформаційна міра Кульбака [11–13].

Подамо нормований ентропійний критерій оптимізації параметрів машинного навчання системи розпізнавати реалізації класу X_m^o як [12]:

$$E_m^{(k)} = \frac{I_m^{(k)}}{I_{\max}^{(k)}} = \frac{H_m^{(k)} - H_m^{(k)}(\gamma)}{H_m^{(k)}}, \qquad (11)$$

де $I_m^{(k)}$ – кількість умовної інформації, оброблюваної на k-му кроці машинного навчання системи розпізнавати вектори ознак класу X_m^o;

$I_{\max}^{(k)}$ – максимальна кількість умовної інформації, одержаної на k-му кроці машинного навчання;

$$H_m^{(k)} = -\sum_{l=1}^{M} p(\gamma_{l,k}) \log_2 p(\gamma_{l,k}) - \qquad (12)$$

апріорна (безумовна) ентропія, яка наявна на k-му кроці машинного навчання системи розпізнавати вектори ознак класу X_m^o;

$$H_m^{(k)} = -\sum_{l=1}^{M}\sum_{m=1}^{M} p(\gamma_{l,k})\, p(\mu_{m,k}/\gamma_{l,k}) \log_2 p(\mu_{m,k}/\gamma_{l,k}) - \quad (13)$$

апостеріорна (умовна) ентропія, що характеризує залишкову невизначеність після k-го кроку навчання системи розпізнавати реалізації класу X_m^o;

$p(\gamma_{l,k})$ – безумовна ймовірність прийняття на k-му кроці машинного навчання гіпотези $\gamma_{l,k}$;

$p(\mu_{m,k}/\gamma_{l,k})$ – апостеріорна ймовірність прийняття на k-му кроці машинного навчання рішення $\mu_{m,k}$ за умови, що прийнята гіпотеза $\gamma_{l,k}$.

Для двоальтернативної системи оцінювань ($M = 2$) і рівноймовірних гіпотез, що характеризує згідно з принципом Бернуллі – Лапласа найбільш важкий у статистичному розумінні випадок прийняття рішень, після відповідної підстановки ентропій (12) і (13) у вираз (11) та заміни за формулою Байєса відповідних апостеріорних ймовірностей на апріорні критерій (11) набирає вигляду

$$E_m^{(k)} = 1 + \frac{1}{2}\left(\frac{\alpha_m^{(k)}(d)}{\alpha_m^{(k)}(d) + D_{2,m}^{(k)}(d)} \log_2 \frac{\alpha_m^{(k)}(d)}{\alpha_m^{(k)}(d) + D_{2,m}^{(k)}(d)} + \right.$$

$$+\frac{\beta_m^{(k)}(d)}{D_{1,m}^{(k)}(d)+\beta_m^{(k)}(d)}\log_2\frac{\beta_m^{(k)}(d)}{D_{1,m}^{(k)}(d)+\beta_m^{(k)}(d)}+$$

$$+\frac{D_{1,m}(d)}{D_{1,m}^{(k)}(d)+\beta_m^{(k)}(d)}\log_2\frac{D_{1,m}(d)}{D_{1,m}^{(k)}(d)+\beta_m^{(k)}(d)}+$$

$$+\frac{D_{2,m}^{(k)}(d)}{\alpha_m^{(k)}(d)+D_{2,m}^{(k)}(d)}\log_2\frac{D_{2,m}^{(k)}(d)}{\alpha_m^{(k)}(d)+D_{2,m}^{(k)}(d)}\Bigg), \qquad (14)$$

де $\alpha_m^{(k)}(d)$ – помилка першого роду прийняття рішення на *k*-му кроці машинного навчання;

$\beta_m^{(k)}(d)$ – помилка другого роду;

$D_{1,m}^{(k)}(d)$ – перша достовірність;

$D_{2,m}^{(k)}(d)$ – друга достовірність;

d – дистанційна міра, яка визначає радіуси гіперсферичних контейнерів класів розпізнавання, побудованих в радіальному базисі бінарного простору Хеммінга.

Оскільки точнісні характеристики є функціями відстані роздільної гіперповерхні від геометричних центрів контейнерів відповідних класів розпізнавання, то критерій (14) потрібно розглядати як нелінійний і взаємно-неоднозначний функціонал від точнісних характеристик, що потребує перебування в процесі машинного навчання робочої (допустимої) області його визначення.

Розглянемо процедуру обчислення в практичних задачах критерію (14). Оскільки інформаційний критерій є функціоналом від точнісних характеристик, то за репрезентативного обсягу навчальної вибірки необхідно користуватися їх оцінками:

$$D_{1,m}^{(k)}(d)=\frac{K_{1,m}^{(k)}(d)}{n_{\min}};\ \ \alpha_{m}^{(k)}(d)=\frac{K_{2,m}^{(k)}(d)}{n_{\min}};$$

$$\beta_{m}^{(k)}(d)=\frac{K_{3,m}^{(k)}(d)}{n_{\min}};\ \ D_{2,m}^{(k)}(d)=\frac{K_{4,m}^{(k)}(d)}{n_{\min}}, \qquad (15)$$

де $K_{1,m}^{(k)}(d)$ – кількість подій, які означають належність «своїх» реалізацій до класу розпізнавання X_m^o;

$K_{2,m}^{(k)}(d)$ – кількість подій, які означають неналежність «своїх» реалізацій до класу розпізнавання X_m^o;

$K_{3,m}^{(k)}(d)$ – кількість подій, які означають належність «чужих» реалізацій до класу розпізнавання X_m^o;

$K_{4,m}^{(k)}(d)$ – кількість подій, які означають неналежність «чужих» реалізацій до класу розпізнавання X_m^o;

$n_{\min}$ – мінімальний обсяг репрезентативної навчальної вибірки, який визначається за методом, запропонованим у праці [11].

Після підставлення відповідних позначень (15) у вираз (14) одержимо робочу формулу для обчислення в рамках IEI-технології ентропійного критерію оптимізації параметрів машинного навчання системи розпізнавати структуровані вектори ознак класу X_m^o:

$$E_m^{(k)} = 1 + \frac{1}{2}\left(\frac{K_{1,m}^{(k)}(d)}{K_{1,m}^{(k)}(d) + K_{3,m}^{(k)}(d)} \log_2 \frac{K_{1,m}^{(k)}(d)}{K_{1,m}^{(k)}(d) + K_{3,m}^{(k)}(d)} + \right.$$
$$+ \frac{K_{2,m}^{(k)}(d)}{K_{2,m}^{(k)}(d) + K_{4,m}^{(k)}(d)} \log_2 \frac{K_{2,m}^{(k)}(d)}{K_{2,m}^{(k)}(d) + K_{4,m}^{(k)}(d)} +$$
$$+ \frac{K_{3,m}^{(k)}(d)}{K_{1,m}^{(k)}(d) + K_{3,m}^{(k)}(d)} \log_2 \frac{K_{3,m}^{(k)}(d)}{K_{1,m}^{(k)}(d) + K_{3,m}^{(k)}(d)} +$$
$$\left. + \frac{K_{4,m}^{(k)}(d)}{K_{2,m}^{(k)}(d) + K_{4,m}^{(k)}(d)} \log_2 \frac{K_{4,m}^{(k)}(d)}{K_{2,m}^{(k)}(d) + K_{4,m}^{(k)}(d)} \right). \quad (16)$$

У праці [10] запропоновано модифікацію диференціальної інформаційної міри Кульбака, яка подається як добуток відношення правдоподібності на міру відхилень відповідних розподілів імовірностей. Для двохальтернативних апріорно рівноймовірних рішень модифікований критерій Кульбака, який обчислюється на k-му кроці машинного навчання системи розпізнавати реалізації класу X_m^o, має вигляд

$$E_m^{(k)} = \log_2 \left(\frac{2 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d))}{\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)} \right) * \left[1 - (\alpha_m^{(k)}(d) + \beta_m^{(k)}(d)) \right]. \quad (17)$$

Нормована форма критерію (17) має вигляд

$$E_{K,m}^{(k)} = \frac{E_{K\,m}^{(k)}}{E_{K\max}^{(k)}}, \quad (18)$$

де $E_{K\max}^{(k)}$ – значення інформаційного критерію (17) при

$$D_{1,m}^{(k)}(d) = D_{2,m}^{(k)}(d) = 1 \quad \text{і} \quad \alpha_m^{(k)}(d) = \beta_m^{(k)}(d) = 0.$$

Робоча модифікація критерію (17) після відповідного підставлення оцінок (15) набрала вигляду

$$E_m^{(k)} = \frac{\left[n - (K_{2,m}^{(k)}(d) + K_{3,m}^{(k)}(d))\right]}{n} * \\ * \log_2 \left\{ \frac{2n + 10^{-r} - \left[K_{2,m}^{(k)}(d) + K_{3,m}^{(k)}(d)\right]}{\left[K_{2,m}^{(k)}(d) + K_{3,m}^{(k)}(d)\right] + 10^{-r}} \right\}, \qquad (19)$$

де 10^{-r} – достатньо мале число, яке вводиться для уникнення поділу на нуль;

r – число, яке рекомендується на практиці вибирати з інтервалу

$$1 < r \le 3.$$

Розглянемо схему обчислення змінних $K_1^{(k)} - K_4^{(k)}$ у формулах (16) і (19). На рисунку 1.3 показано структуру навчальної матриці під час побудови оптимального контейнера класу розпізнавання X_1^o. Навчальна матриця складається з векторів реалізацій двох найближчих сусідніх класів розпізнавання: $\{x_1^{(j)}\} \in X_1^o$ і $\{x_2^{(j)}\} \in X_2^o$.

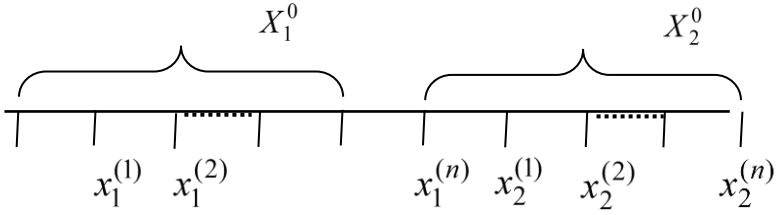


Рисунок 1.3 – Структура навчальної матриці найближчих сусідніх класів розпізнавання

Алгоритм обчислення змінних $K_1^{(k)}$ – $K_4^{(k)}$ має такий вигляд у предикатній формі:

$$(\forall X_1^o \in \Re^{|\Lambda|})(\forall X_2^o \in \Re^{|\Lambda|})[if\ \ x_1^{(j)} \in X_1\ \ then$$
$$K_1(j) := K_1(j-1)+1\ \ else\ \ K_2(j-1)+1],$$
$$(\forall X_1^o \in \Re^{|\Lambda|})\ (\forall X_2^o \in \Re^{|\Lambda|})[if\ \ x_2^{(j)} \in X_1\ \ then$$
$$K_3(j) := K_3(j-1)+1\ else\ \ K_4(j) := K_4(j-1)+1].$$

Водночас визначення належності, наприклад, вектора $x_m^{(j)}$ до свого класу здійснюється за таким правилом:

1) обчислюється кодова відстань $d[x_m \oplus x_m^{(j)}]$;
2) якщо

$$d[x_m \oplus x_m^{(j)}] \le d_m,$$

то $x_m^{(j)} \in X_m^o$, інакше $x_m^{(j)} \notin X_m^o$.

Таким чином, інформаційні критерії (14) і (17) є функціоналами як від точнісних характеристик класифікаційних рішень, так і від дистанційних критеріїв, що дозволяє їх вважати загальними критеріями валідності машинного навчання, оскільки вони є узагальненням відомих статистичних і детермінованих (дистанційних) критеріїв близькості.

1.7. Базовий алгоритм інформаційно-екстремального машинного навчання системи виявлення атак

Базовий інформаційно-екстремальний алгоритм оптимізації просторово-часових параметрів функціонування СВА реалізується у внутрішньому циклі

процедури машинного навчання (3), що й обумовило його назву. Призначенням базового алгоритму навчання є:

- оптимізація геометричних параметрів контейнерів класів розпізнавання;
- обчислення інформаційного критерію оптимізації параметрів машинного навчання системи;
- пошук глобального максимуму інформаційного критерію в робочій (допустимій) області визначення його функції.

Категорійну модель інформаційно-екстремального машинного навчання СВА за базовим алгоритмом подамо у вигляді спрямованого графа відображення операторами одна на одну відповідних множин, застосовуваних у процесі навчання. Вхідний математичний опис здатної навчатися СВА подамо у вигляді структури

$$I_{вх} = < G, T, \Omega, Z, Y, X; f_1, f_2 > , \qquad (20)$$

де G – простір вхідних сигналів (факторів);

T – множина моментів часу зняття інформації;

Ω – простір ознак розпізнавання;

Z – простір можливих станів кіберзахисту ІКС;

Y – вхідна навчальна матриця;

X – робоча бінарна навчальна матриця, яка в процесі інформаційно-екстремального машинного навчання адаптується до максимальної повної ймовірності прийняття правильних діагностичних рішень;

f_1 – оператор формування вхідної навчальної матриці;

Y – із джерела інформації, яке задається декартовим добутком $G \times T \times \Omega \times Z$;

f_2 – оператор перетворення вхідної навчальної матриці Y на робочу бінарну навчальну матрицю X.

Категорійну модель інформаційно-екстремального машинного навчання СВА за базовим алгоритмом показано на рисунку 1.4.

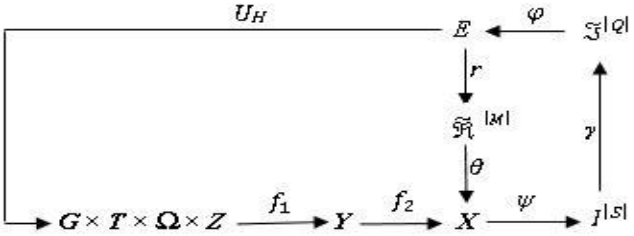


Рисунок 1.4 – Категорійна модель машинного навчання за базовим алгоритмом

На рисунку 1.4 терммножина E, яка складається з обчислених на кожному кроці машинного навчання значень інформаційного критерію, є загальною для всіх контурів оптимізації параметрів. Оператор $r: E \to \widetilde{\Re}^{|M|}$ в процесі машинного навчання відновлює в радіальному базисі бінарного простору ознак контейнери класів розпізнавання, які утворюють розбиття $\widetilde{\Re}^{|M|}$. Оператор θ відображає розбиття $\widetilde{\Re}^{|M|}$ на нечіткий розподіл апріорно класифікованих двійкових векторів ознак класів розпізнавання. Далі оператор $\psi: X \to I^{|S|}$, де $I^{|S|}$ – множина гіпотез, перевіряє основну статистичну гіпотезу $\gamma_1: x_m^{(j)} \in X_m^o$. Оператор γ визначає множину точнісних характеристик $\Im^{|Q|}$, де $Q = S^2$, а оператор φ обчислює множину значень E інформаційного критерію оптимізації,

який є функціоналом від точнісних характеристик. Контур оптимізації контрольних допусків замикається через терммножину D, елементами якої є значення контрольних допусків на ознаки розпізнавання. Оператор u регламентує процес машинного навчання.

Згідно з принципом відкладених рішень О. Г. Івахненка для максимізації інформаційної спроможності системи може бути необхідною оптимізація інших параметрів, які впливають на функціональну ефективність машинного навчання. У цьому разі категорійна модель буде мати додаткові контури оптимізації цих параметрів. Крім того, згідно з принципом повної композиції контури оптимізації повинні мати загальну терммножину E, елементи якої обчислюються на кожному кроці машинного навчання.

Вхідною інформацією для навчання за базовим алгоритмом є тривимірний масив реалізацій класів розпізнавання $\{y_{m,i}^{(j)} \mid m=\overline{1,M}; i=\overline{1,N}; j=\overline{1,n}\}$; значення параметра поля контрольних допусків δ на ознаки розпізнавання і рівні селекції (квантування) $\{\rho_m\}$ координат усереднених двійкових векторів ознак класів розпізнавання, які за замовчуванням дорівнюють 0,5 для всіх класів розпізнавання.

За базовий беруть клас розпізнавання X_1^o, який характеризує нормальний стан функціонування ІКС і стосовно якого визначаються контрольні допуски.

Розглянемо етапи реалізації алгоритму:

1. Обчислюється для навчальної матриці класу розпізнавання X_1^o усереднений вектор ознак $\{y_{1,i} \mid i=\overline{1,N}\}$.

2. Формується масив $\{x_{1,i}^{(j)}\}$ двійкових векторів ознак класу розпізнавання X_1^o за правилом

$$x_1^{(j)} = \begin{cases} 1, & \text{if} \quad y_{1,i} - \delta \le y_{1,i}^{(j)} \le y_{1,i} + \delta, \\ 0, & \text{if} \quad \text{else}. \end{cases} \tag{21}$$

3. Формується масив усереднених двійкових векторів-реалізацій $\{x_{m,i} \mid m = \overline{1,M},\ i = \overline{1,N}\}$, елементи яких визначаються за правилом

$$x_{m,i} = \begin{cases} 1, & if \quad \dfrac{1}{n}\sum_{j=1}^{n} x_{m,i}^{(j)} > \rho_m, \\ 0, & if \quad else, \end{cases}$$

де ρ_m – рівень селекції координат двійкового вектора $x_m \in X_m^o$.

4. Розбиття множини усереднених векторів ознак на пари найближчих «сусідів» $\Re_m^{|2|} = < x_m , x_l >$, де x_l – усереднений вектор ознак сусіднього класу X_l^o, за такою схемою:

а) структурують множину векторів $\{x_m\}$, починаючи з вектора x_1 базового класу X_1^o, що характеризує нормальний стан функціонування ІКС;

б) будують матрицю розмірності $M \times M$ кодових відстаней між усередненими векторами ознак усіх класів розпізнавання;

в) для кожного рядка матриці кодових відстаней обчислюють мінімальний елемент, який належить стовпчику вектора, найближчого до вектора, що визначає рядок. За наявності декількох однакових мінімальних елементів вибирають із них будь-який, оскільки вони є рівноправними;

г) формують структуровану множину елементів попарного розбиття $\{\Re_m^{|2|} \mid m = \overline{1, M}\}$, яка задає план машинного навчання.

5. Здійснюють оптимізацію кодової відстані d_m, яка змінюється за заданим законом. У цьому разі беруть $E_m(0) = 0.$

6. Процедура закінчується в разі знаходження максимуму інформаційного критерію оптимізації параметрів машинного навчання в робочій області визначення його функції.

Таким чином, базовий алгоритм навчання є ітераційною процедурою пошуку глобального максимуму інформаційного критерію оптимізації параметрів машинного навчання в робочій області визначення його функції:

$$d_m^* = \arg \max_{G_E \cap \{d\}} E_m^*(d). \qquad (22)$$

На рисунку 1.5 показано геометричну інтерпретацію реалізації базового алгоритму інформаційно-екстремального машинного навчання СВА на прикладі побудови оптимального гіперсферичного контейнера класу розпізнавання X_1^o.

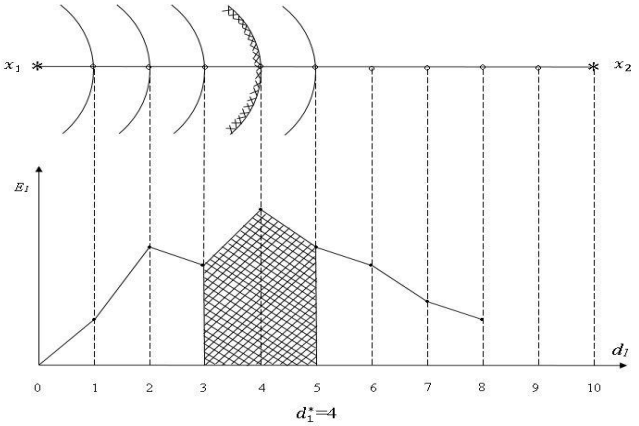


Рисунок 1.5 – Схема реалізації базового алгоритму інформаційно-екстремального машинного навчання

На рисунку 1.5 показано в бінарному просторі Хеммінга вершини усередненого двійкового вектора x_1 класу розпізнавання X_1^o й усередненого двійкового вектора x_2 класу розпізнавання X_2^o, який є найближчим сусідом для класу розпізнавання X_1^o. На схемі кодова відстань Хеммінга між векторами x_1 і x_2 дорівнює $d = (x_1 \oplus x_2) = 11$. У процесі машинного навчання за базовим алгоритмом здійснюється покрокове збільшення радіуса d_1 контейнера класу розпізнавання X_1^o на одну кодову одиницю. Водночас згідно з умовою (7) на величину радіуса d_1 накладається обмеження

$$d_1 < d(x_1 \oplus x_2) - 1.$$

На кожному кроці машинного навчання обчислюється значення інформаційного критерію E_1 оптимізації радіуса контейнера класу розпізнавання X_1^o. Як оптимальний радіус беруть екстремальне значення глобального максимуму критерію E_1, обчислене в робочій (допустимій) області визначення функції інформаційного критерію. На рисунку 1.5 робоча область обчислення інформаційного критерію позначена заштрихованою ділянкою. Для двоальтернативних рішень робоча область існує за умови, що їх перша і друга достовірності перевищують відповідно помилки першого та другого родів. Аналіз рисунка 1.5 засвідчує, що максимальне значення критерію E_1^* одержане на четвертому кроці машинного навчання, тобто оптимальний радіус контейнера класу розпізнавання X_1^o дорівнює $d_1^* = 4$ (тут і далі в кодових одиницях).

Таким чином, основною функцією базового алгоритму машинного навчання в рамках IEI-технології є обчислення на кожному кроці навчання інформаційного критерію та організація пошуку його глобального максимуму в робочій області визначення функції критерію з метою визначення оптимальних геометричних параметрів розбиття простору ознак на класи розпізнавання. При гіперсферичному контейнері класів розпізнавання такими параметрами при інформаційно-екстремальному машинному навчанні за базовим алгоритмом є оптимальні кодові відстані $\{d_m^*\}$ і оптимальні усереднені вектори-реалізації $\{x_m^*\}$ для заданого алфавіту $\{X_m^o\}$.

1.8. Функціонування системи виявлення атак у режимі екзамену

За одержаними в процесі машинного навчання оптимальними геометричними параметрами контейнерів класів розпізнавання будуються вирішальні правила, які в продукційній формі подамо у вигляді

$$(\forall X_k^o \in \tilde{\Re}^{|M|})(\forall X_l^o \in \tilde{\Re}^{|M|})\Big(f\ [(\mu_m > 0)\,\&\,(\mu_m = \max_{\{m\}}\{\mu_m\}] \quad then\ x^{(j)} \in X_m^o \quad else\ x^{(j)} \notin X_m^o\Big), \tag{23}$$

де $x^{(j)}$ – вектор, що розпізнається;

μ_m – функція належності вектора $x^{(j)}$ до контейнера класу розпізнавання X_m^o.

У виразі (23) функція належності для гіперсферичного контейнера класу розпізнавання X_m^o визначається за формулою

$$\mu_m = 1 - \frac{d(x_m^* \oplus x^{(j)})}{d_m^*}, \tag{24}$$

де x_m^* – оптимальний усереднений двійковий вектор ознак;

d_m^* – оптимальний радіус гіперсферичного контейнера.

Таким чином, при функціонуванні СВА в режимі екзамену визначається за вирішальними правилами (23) належність реалізації розпізнавального класу до одного з класів із заданого алфавіту. Водночас вирішальні правила через малу обчислювальну трудомісткість відрізняються високою оперативністю.

Оцінювання функціональної ефективності інформаційно-екстремального машинного навчання здійснюється під час функціонування СВА в режимі екзамену, алгоритм якого аналогічний алгоритму функціонування системи безпосередньо в робочому режимі. У рамках IEI-технології категорійну модель у вигляді орієнтованого графа відображень множин, що застосовуються на етапі екзамену, показано на рисунку 1.6.

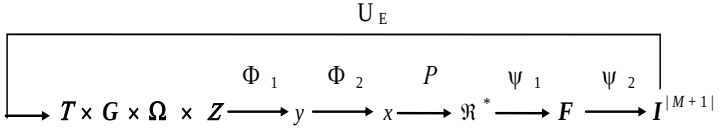


Рисунок 1.6 – Категорійна модель функціонування СВА в режимі екзамену

У категорійній моделі (рис. 1.6) оператор Φ_1 формує екзаменаційний вектор ознак y класу, що розпізнається, аналогічний за структурою вектору з навчальної матриці. Оператор Φ_2 за заданими на етапі машинного навчання контрольними допусками формує двійковий вектор x, а оператор P відображає цей вектор на побудоване на етапі машинного навчання оптимальне розбиття $\Re^*$ класів розпізнавання. Оператор Ψ_1 для кожного вектора ознак, що розпізнається, обчислює значення побудованих на етапі машинного навчання вирішальних правил і формує терммножину F, а оператор Ψ_2 за максимальним значенням вирішального правила відносить вектор x, до одного з класів заданого алфавіту $\{X_m^o\}$. У цьому разі множина гіпотез $I^{|M+1|}$ містить додаткову гіпотезу γ_{M+1}, що характеризує

некласифіковане рішення, тобто вектор, який розпізнається, не належить до жодного класу із заданого алфавіту класів розпізнавання. Призначенням оператора U_E є регламентація процесу екзамену.

Розглянемо схему реалізації алгоритму екзамену:

1) ініціалізація лічильника класів розпізнавання $m := 0$;

2) $m := m + 1$;

3) ініціалізація лічильника кількості реалізацій $j := 0$;

4) $j := j + 1$;

5) обчислення функції належності (24);

6) порівняння: якщо $j \le n$, то виконується пункт 4, інакше – пункт 7;

7) обчислення усередненого значення функції належності (24):

$$\bar{\mu}_m = \frac{1}{n} \sum_{j=1}^{n} \mu_{m,j}; \qquad (25)$$

8) порівняння: якщо $m \le M$, то виконується пункт 2, інакше – пункт 9;

9) обчислення максимального значення функції (25):

$$\bar{\mu}_m^* = \max_{\{m\}} \bar{\mu}_m;$$

10) визначення класу розпізнавання за максимальним значенням функції (25);

11) якщо для всіх класів розпізнавання максимальні значення функції (25) від'ємні, то екзаменаційна реалізація не класифікується;

12) зупин.

Таким чином, побудовані в процесі машинного навчання в рамках геометричного підходу вирішальні правила є чіткими і відрізняються незначною обчислювальною трудомісткістю і тому характеризуються високою оперативністю, що є важливим фактором під час виявлення атак.

Розглянуті вище категорійні моделі відбивають притаманні людині перетворення інформації та інформаційні потоки, які мають місце за когнітивних процесів формування та прийняття класифікаційних рішень. Тому їх можна розглядати як узагальнені структурні схеми алгоритмів інформаційно-екстремального машинного навчання і функціонування СККБ в режимі екзамену.

1.9. Машинне навчання системи виявлення атак з оптимізацією контрольних допусків на ознаки розпізнавання

У рамках ІЕІ-технології адаптація вхідного математичного опису здатної навчатися системи до її максимальної інформаційної спроможності здійснюється шляхом оптимізації параметрів машинного навчання за інформаційним критерієм. Реалізація базового алгоритму машинного навчання в загальному випадку не гарантує високої достовірності розпізнавання трафіку в разі функціонування СККЗ в режимі моніторингу, оскільки стартові контрольні допуски на ознаки розпізнавання зазвичай є неоптимальними. Таким чином, виникає необхідність збільшення глибини машинного навчання за допомогою оптимізації системи контрольних допусків, які істотно впливають як на геометричні параметри контейнерів класів розпізнавання, так і на точнісні характеристики класифікаційних рішень. Нехай задано

вектор параметрів машинного навчання для алфавіту класів розпізнавання $\{X_m^o\}$

$$g_m =< x_m, d_m, \delta >, \qquad (26)$$

де x_m – усереднений структурований вектор ознак класу розпізнавання X_m^o;

d_m – радіус гіперсферичного контейнера класу розпізнавання X_m^o, який відновлюється в радіальному базисі простору ознак розпізнавання;

δ – параметр поля контрольних допусків на ознаку розпізнавання.

Параметр δ дорівнює половині симетричного поля контрольних допусків на ознаки розпізнавання, як це показано на рисунку 1.7.

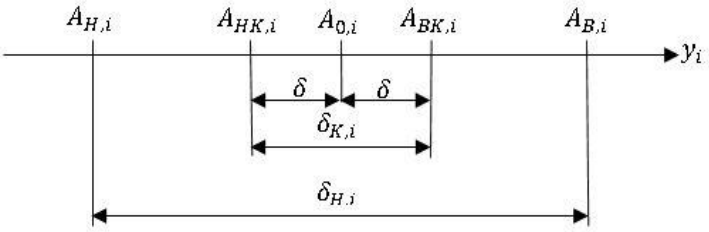


Рисунок 1.7 – Поле допусків на ознаку розпізнавання

На рисунку 1.7 взято такі позначення: $A_{0,i}$ – номінальне значення ознаки y_i; $A_{H,i}$ – нижній нормований (експлуатаційний) допуск; $A_{B,i}$ – верхній нормований допуск; $A_{HK,i}$ – нижній контрольний допуск;

$A_{BK,i}$ – верхній контрольний допуск; $\delta_{K,i}$ – поле контрольних допусків; $\delta_{H,i}$ – поле нормованих допусків.

Двобічне симетричне поле контрольних допусків через параметр δ визначається за формулою

$$\delta_{K,\,i} = 2\delta \frac{A_{B,i} - A_{H,i}}{a}, \qquad (27)$$

де a – кількість градацій контрольного поля допусків, яка для всіх ознак розпізнавання є однаковою.

Область значень радіуса контейнера класу розпізнавання X_m^o задається нерівністю

$$d_m < d(x_m \oplus x_c),$$

де $d(x_m \oplus x_c)$ – міжцентрова відстань найближчих класів розпізнавання X_m^o і X_c^o, яка визначається як кодова відстань між відповідними векторами ознак x_m і x_c.

На практиці при $a = 100$ параметр δ може визначатися як кількість відсотків відхилення i-ї ознаки розпізнавання від її номінального значення $A_{0,i}$.

Категорійну модель навчання СВА з оптимізацією контрольних допусків на ознаки розпізнавання з урахуванням моделі базового алгоритму навчання показано на рисунку 1.8.

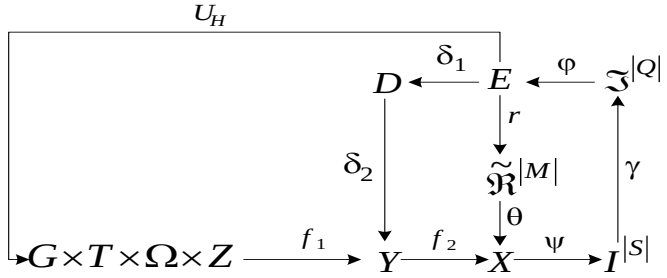


Рисунок 1.8 – Категорійна модель машинного навчання з оптимізацією контрольних допусків

Категорійна модель (рис. 1.8) порівняно з моделлю машинного навчання за базовим алгоритмом (рис. 1.5) містить додатковий контур операторів оптимізації контрольних допусків на ознаки розпізнавання, який замикається через термножину D допустимих значень системи контрольних допусків. Водночас оператор δ_1 на кожному кроці машинного навчання змінює контрольне поле, а оператор δ_2 оцінює залежність ознак розпізнавання заданому контрольному полю допусків.

Алгоритм навчання системи з оптимізацією контрольних допусків на ознаки розпізнавання згідно з категорійною моделлю (рис. 1.8) має вигляд

$$\{\delta^*_{K,i} \mid i = \overline{1,N}\} = \arg \max_{G_\delta}\{\max_{G_E \cap \{k\}} \overline{E}^{(k)}\}, \qquad (28)$$

де $\overline{E}^{(k)} = \frac{1}{M}\sum_{m=1}^{M} E_m^{(k)}$ – усереднене за алфавітом класів розпізнавання значення інформаційного критерію оптимізації, обчислене на k-му кроці машинного навчання;

G_δ – область допустимих значень контрольних допусків на ознаки розпізнавання.

Оптимізація контрольних допусків на ознаки розпізнавання може здійснюватися за такими основними схемами:

– алгоритм паралельної оптимізації, за яким контрольні допуски оптимізуються для всіх ознак одночасно;

– алгоритм послідовної оптимізації, за яким контрольні допуски оптимізуються послідовно для кожної ознаки розпізнавання за фіксованих (стартових) значень інших ознак;

– алгоритм оптимізації за зведеним полем допусків, який доцільно застосовувати як послідовно-паралельний алгоритм за наявності різних шкал вимірювання для окремих груп ознак розпізнавання.

Перевагою паралельного алгоритму оптимізації контрольних допусків є висока оперативність реалізації алгоритму, але він не дозволяє одержати точне значення глобального максимуму інформаційного критерію в робочій області визначення його функції. Тому екстремальні значення параметрів функціонування СВА, одержані в процесі їх оптимізації за паралельним алгоритмом, є квазіоптимальними.

Алгоритми послідовної оптимізації системи контрольних допусків дозволяють обчислювати точні значення глобального максимуму інформаційного критерію в робочій області, але характеризуються низькою оперативністю. З метою поєднання переваг цих алгоритмів оптимізацію контрольних допусків доцільно здійснювати за паралельно-послідовним алгоритмом. При цьому реалізація паралельного алгоритму дозволяє визначити стартові контрольні допуски, які є вхідними для алгоритму послідовної оптимізації. Це дозволяє

підвищити оперативність послідовного алгоритму, оскільки стартові квазіоптимальні контрольні допуски вже знаходяться в робочій області визначення функції інформаційного критерію.

Структурований алгоритм послідовної оптимізації поля контрольних допусків на ознаки розпізнавання має вигляд

$$\delta^{*}_{K,i} = \arg\left[\overset{L}{\underset{l=1}{\otimes}} \max_{G_\delta} \{ \max_{G_E \cap G_d} \overline{E}_l^{(i)}(d,\delta) \} \right], \qquad (29)$$

де $\overline{E}_l^{(i)}(d,\delta)$ – усереднене значення інформаційного критерію оптимізації параметрів машинного навчання СВА, обчислене під час оптимізації контрольних допусків i-ї ознаки розпізнавання на l-му прогоні ітераційної процедури оптимізації системи контрольних допусків на ознаки розпізнавання;

G_{δ_i} – область допустимих значень поля контрольних допусків для i-ї ознаки;

G_E – робоча область визначення функції інформаційного критерію оптимізації;

G_d – область допустимих значень кодової відстані d, яка визначає радіус гіперсферичного контейнера класу розпізнавання;

$\otimes$ – символ операції повторення;

L – кількість прогонів ітераційної процедури оптимізації системи контрольних допусків на ознаки розпізнавання;

N – кількість ознак розпізнавання.

Необхідно на етапі машинного навчання СККЗ оптимізувати параметри вектора (26), які забезпечують максимальне значення інформаційного критерію

оптимізації (10) в робочій (допустимій) області визначення його функції.

Розглянемо інформаційно-екстремальне машинне навчання СВА з гіперсферичним класифікатором, в якому відновлення контейнерів класів розпізнавання відбувається шляхом паралельної оптимізації контрольних допусків на ознаки розпізнавання. Вхідною інформацією для алгоритму навчання є масив навчальної матриці $\{y_{m,i}^{(j)}\}$ і система полів нормованих допусків $\{\delta_{\text{Н},i}\}$ на ознаки розпізнавання, яка задає область значень відповідних контрольних допусків.

Розглянемо основні етапи реалізації алгоритму машинного навчання СВА з паралельною оптимізацією контрольних допусків на ознаки розпізнавання, за якою на кожному кроці навчання змінюються контрольні допуски для всіх ознак розпізнавання одночасно:

1) ініціалізація лічильника кроків зміни параметра δ поля контрольних допусків: $\delta := 0$;

2) $\delta := \delta + 1$;

3) обчислюються для всіх ознак розпізнавання нижні $A_{НК,i}$ і верхні $A_{ВК,i}$ контрольні допуски на ознаки розпізнавання:

$$A_{НК,i} = \overline{y}_i - \delta; \quad A_{ВК,i} = \overline{y}_i + \delta. \tag{30}$$

4) реалізується базовий алгоритм, за яким для кожного значення параметра δ визначаються згідно з процедурою (22) оптимальні значення радіусів контейнерів класів розпізнавання;

5) якщо $\delta < \delta_{\text{Н}} / 2$, то виконується пункт 2, інакше – пункт 6;

6) обчислюється усереднене за алфавітом класів розпізнавання максимальне значення інформаційного критерію $\overline{E}^*$;

7) визначається оптимальний параметр поля контрольних допусків, який забезпечує максимальне значення усередненого критерію $\overline{E}^*$:

$$\delta^* = \arg \overline{E}^*;$$

8) обчислюються за формулами (30) відповідні оптимальні контрольні допуски на ознаки розпізнавання

$$A^*_{НК,i} = \overline{y}_i - \delta^*; \quad A^*_{ВК,i} = \overline{y}_i + \delta^*;$$

9) у базі знань запам'ятовуються оптимальні параметри машинного навчання (26);

10) зупин.

Визначені на етапі паралельної оптимізації контрольні допуски на ознаки розпізнавання є квазіоптимальними, оскільки вони були обчислені за однакову кількість кроків навчання для всіх ознак. Для визначення оптимальних контрольних допусків на ознаки розпізнавання в методах інформаційно-екстремального машинного навчання здійснюється їх послідовна оптимізація. Водночас визначені на етапі паралельної оптимізації контрольні допуски беруться як стартові для алгоритму послідовної оптимізації. Оскільки під час оптимізації i-ї ознаки інші наступні ознаки мають неоптимальні контрольні допуски, то послідовна оптимізація потребує у цьому разі проведення ітераційних прогонів до того часу, поки значення інформаційного критерію оптимізації не буде змінюватися. Це призводить

до зменшення оперативності алгоритму машинного навчання. Але завдяки тому, що стартовими обрано квазіоптимальні контрольні допуски, то під час послідовної оптимізації обчислені на кожному кроці навчання значення інформаційного критерію оптимізації будуть перебувати постійно в робочій області визначення його функції. Тому на практиці оперативність алгоритму інформаційно-екстремального машинного навчання з паралельно-послідовною оптимізацією контрольних допусків на ознаки розпізнавання вже за кількості ознак $N \geq 10$ перевершує оперативність послідовної оптимізації.

Розглянемо основні етапи алгоритму (29) послідовної оптимізації контрольних допусків на діагностичні ознаки.

1. Ініціалізація лічильника прогонів процедури оптимізації параметрів машинного навчання: $s := 0$.

2. $s := s + 1$.

3. Ініціалізація лічильника ознак розпізнавання: $i := 0$.

4. $i := i + 1$.

5. Визначення екстремального значення параметра поля контрольних допусків за процедурою (28)

6. Порівняння: якщо $i \leq N$, то виконується крок 4, інакше − крок 7.

7. Обчислюється усереднене за алфавітом класів розпізнавання значення інформаційного критерію $\overline{E}^{(s)}$.

8. Якщо $\left|\overline{E}^{(s-1)} - \overline{E}^{(s)}\right| \leq \varepsilon$, де ε – будь-яке мале додатне число, то виконується крок 9, інакше − крок 2.

9. Обчислюються оптимальні параметри поля контрольних допусків на ознаки розпізнавання шляхом операції присвоювання

$$\{\delta_i^* := \delta_i^{(s)} \mid i = \overline{1,N}\},$$

10. За формулами (30) обчислюються оптимальні нижні та верхні контрольні допуски.

11. Запам'ятовуються оптимальні координати структурованого вектора (26):

$\{x_m^* \mid m = \overline{1,M}\}$ – оптимальні усередненні вектори ознак класів розпізнавання із заданого алфавіту;

$\{d_m^* \mid m = \overline{1,M}\}$ – оптимальні радіуси контейнерів класів розпізнавання;

$\{A_{НК,i}^* \mid i = \overline{1,N}\}, \{A_{ВК,i}^* \mid i = \overline{1,N}\}$ – оптимальні верхні контрольні допуски на ознаки розпізнавання..

12. Зупин.

Отже, одержані за результатами інформаційно-екстремального машинного навчання оптимальні параметри функціонування СВА дозволяють побудувати в просторі ознак розпізнавання вирішальні правила (23) для прийняття класифікаційних рішень.

1.10. Інформаційно-екстремальне машинне навчання системи виявлення атак за ієрархічною структурою даних

Якщо побудова вирішальних правил у межах геометричного підходу робить їх практично інваріантними до багатовимірності простору ознак розпізнавання, то при цьому залишається проблема багатовимірності алфавіту класів розпізнавання, потужність якого під час виявлення атак може досягати великих значень. Відомо, що під час збільшення потужності алфавіту класів розпізнавання в незмінному просторі ознак збільшується ступінь перетину класів

розпізнавання і відповідно достовірність буде зменшуватися. Основним напрямом зменшення впливу багатовимірності алфавіту класів розпізнавання на функціональну ефективність машинного навчання інтелектуальної системи є перехід від лінійних структур даних до ієрархічних.

Розглянемо ієрархічну структуру даних у вигляді дерева з донизу східним перенесенням атрибутів вершин. На відміну від рекурсивної ієрархічної структури таку структуру будемо називати декурсивною.

На рисунку 1.9 показано приклад декурсивного дерева, в якому атрибути – класи розпізнавання передаються з вершин вищого ярусу у вершини відповідних страт нижнього ярусу.

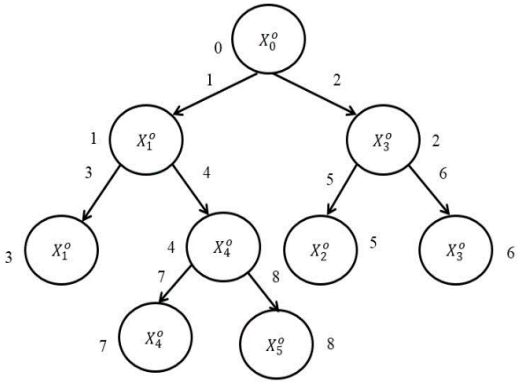


Рисунок 1.9 – Декурсивна ієрархічна структура даних

Атрибутами вершин декурсивного дерева (рис. 1.9) є навчальні матриці класів розпізнавання з заданого алфавіту $\{X^{o}_{h,s,m}\}$, де h – порядковий номер ярусу, s – порядковий номер страти h-го ярусу і m – порядковий номер класу розпізнавання в алфавіті страти. На рисунку 1.9 верхній ярус (перший за дендрографічною

класифікацією) має одну страту, яка складається з двох класів $X^{o}_{1,1,1}$ і $X^{o}_{1,1,3}$, а другий ярус – дві страти, кожна з яких складаються з двох класів $X^{o}_{2,1,1}$, $X^{o}_{2,1,4}$ і $X^{o}_{2,2,2}$, $X^{o}_{2,2,3}$ відповідно. Отже, показана на рисунку 1.9 структура дозволяє підвищувати ймовірність побудови безпомилкових за навчальною матрицею вирішальних правил через зменшення ступеня перетину між класами.

Матрицю інциденції $A=\{a_{\pi,\varsigma}\}$ декурсивного дерева будемо визначати так:

$a_{\pi,\varsigma}$=1, якщо початок ребра ς з'єднується з вершиною π і має напрям від вершини;

$a_{\pi,\varsigma}=-1$, якщо кінець (стрілка) ребра ς з'єднується з іншою вершиною і має напрям від вершини π;

$a_{\pi,\varsigma}=0$, якщо початок ребра ς не з'єднується з вершиною π;

$a_{\pi,\varsigma}=*$, якщо початок ребра ς з'єднується з вершиною π і має напрям від вершини π до вершини страти нижнього ярусу з однаковим атрибутом.

Для декурсивного дерева (рис. 1.9) матриця інциденції наведена в таблиці 1.1.

Таблиця 1.1 – Матриця інциденції

ς \ π	1	2	3	4	5	6	7	8
0	-1	-1	0	0	0	0	0	0
1	1	0	-1	-1	0	0	0	0
2	0	1	0	0	-1	-1	0	0
3	0	0	*	0	0	0	-1	0
4	0	0	0	1	0	0	0	-1
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	*	0	0
7	0	0	0	0	0	0	*	0
8	0	0	0	0	0	0	0	1

У таблиці 1.1 змінна ζ характеризує порядковий номер ребер графа (рис. 1.9), а π – номер вершин. Аналіз таблиці 1.1 показує, що вона враховує перехід класів розпізнавання в свою страту нижнього рівня, що є особливістю декурсивної ієрархічної структури. Цей факт підтверджується наявністю в таблиці позначених символом $*$ елементів. Водночас для матриці інциденцій декурсивного дерева специфічну відмінність від орієнтованого графа встановлює наступна лема.

Лема. Для декурсивного графа з ς ребрами кількість стовпчиків матриці інциденцій, які мають нульову суму елементів, дорівнює $\varsigma - \pi^*$, де π^* – кількість вершин, які передають свої атрибути.

Розглянемо формалізовану постановку задачі інформаційного синтезу здатної навчатися за декурсивною ієрархічною структурою даних СВА. Нехай задано алфавіт класів розпізнавання у вигляді декурсивної ієрархічної структури $\{X^{o}_{h,s,m} \mid h=\overline{1,H}; s=\overline{1,S}; m=\overline{1,M}\}$, де H – кількість ярусів ієрархічної структури; S –

кількість страт на h-му ярусі; M – кількість класів розпізнавання в заданому алфавіті. Водночас кожний клас розпізнавання характеризує відповідний профіль ІКС. За результатами моделювання атак сформовано для кожного класу розпізнавання вхідну навчальну матрицю $\| y^{(j)}_{h,s,m,i} | i = \overline{1,N}, j = \overline{1,n} \|$, де N – кількість ознак розпізнавання в структурованому векторі-реалізації класу розпізнавання $X^{o}_{h,s,m}$; n – кількість реалізацій, яка дорівнює кількості часових інтервалів аналізу трафіка.

Отже i-й стовпчик матриці $\| y^{(j)}_{m,i} \|$ містить значення навчальної вибірки, а j-й рядок є реалізацією класу розпізнавання $X^{o}_{h,s,m}$. Крім того, нехай задано структурований вектор параметрів машинного навчання СВА розпізнавати реалізації класу $X^{o}_{h,s,m}$:

$$g_{h,s} = < x_{h,s,m}, d_{h,,s,m}, \delta_{K,h,s,m,i} >, \qquad (31)$$

де $x_{h,s,m}$ – двійкова усереднена реалізація класу розпізнавання $X^{o}_{h,s,m}$;

$d_{h,,s,m}$ – кодова відстань, яка визначає радіус гіперсферичного контейнера класу $X^{o}_{h,s,m}$;

$\delta_{K,h,s,m,i}$ – параметр, який дорівнює половині симетричного поля контрольних допусків i-ї ознаки вектора $x_{h,s,m}$.

При цьому задано обмеження:

1) закон розподілу реалізацій, наприклад, класу розпізнавання $X^{o}_{h,s,m}$, за якими визначається усереднений

вектор ознак $x_{h,s,m}$, повинен згідно з гіпотезою нечіткої компактності бути наближеним до нормального;

2) областю значень радіуса контейнера класу розпізнавання $X^{o}_{h,s,m}$ є

$$d_{h,s,m} \in [0; d(x_{h,s,m} \oplus x_{h,s,c}) - 1],$$

де $d(x_{h,s,m} \oplus x_{h,s,c})$ – кодова відстань між вектором ознак $x_{h,s,m} \in X^{o}_{h,s,m}$ і аналогічним вектором $x_{h,s,c}$ найближчого класу розпізнавання $X^{o}_{h,s,c}$;

3) для двобічних симетричних допусків на діагностичні ознаки має місце

$$\delta_{K,h,s,i} \in [0; \delta_{E,h,s,i} / 2)],$$

де $\delta_{E,h,s,m,i}$ – поле нормованих допусків i-ї ознаки вектора $x_{h,s,m}$.

На етапі машинного навчання необхідно:

1) оптимізувати параметри вектора (31) за усередненим за алфавітом класів розпізнавання $\{X^{o}_{h,s,m}\}$ інформаційним критерієм

$$\overline{E}_{h,s} = \frac{1}{M} \sum_{m=1}^{M} \max_{G_E \cap G_d} E_{h,s,m}(d_{h,s,m}), \qquad (32)$$

де $E_{h,s,m}(d_{h,s,m})$ – інформаційний критерій оптимізації параметрів машинного навчання СВА розпізнавати реалізації класу $X^{o}_{h,s,m}$.

Згідно з працею [14] категорійну модель

машинного навчання СВА з ієрархічною структурою даних подамо у вигляді орієнтованого графа (рис. 1.10). На відміну від категорійної моделі машинного навчання за лінійним алгоритмом (рис. 1.9) категорійна модель, показана на рисунку 1.10, має додатковий контур оптимізації, який замикається через множину H – задану ієрархічну структуру даних.

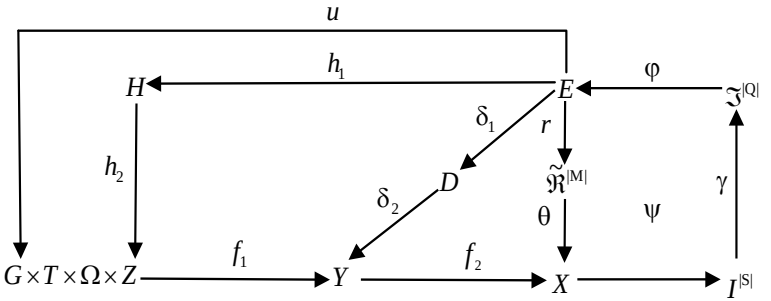


Рисунок 1.10 – Категорійна модель ієрархічного навчання СВА

На рисунку 1.10 оператор h_1 вибирає відповідний ярус і страту в заданій структурі H, тобто задає план машинного навчання, а оператор h_2 змінює простір функціональних станів СККЗ, змінюючи цим алфавіт класів розпізнавання.

Як критерій оптимізації параметрів машинного навчання СВА за ієрархічною структурою даних будемо використовувати модифіковану інформаційну міру Кульбака, яка для рівноймовірних двоальтернативних гіпотез має вигляд

$$E_{h,s,m}^{(k)} = \frac{1}{2}\{2-[\alpha_{h,s,m}^{(k)}(d)+\beta_{h,s,m}^{(k)}(d)]\}\times$$
$$\times\log_2\frac{2-[\alpha_{h,s,m}^{(k)}(d)+\beta_{h,s,m}^{(k)}(d)]+10^{-r}}{\alpha_{h,s,m}^{(k)}(d)+\beta_{h,s,m}^{(k)}(d)]+10^{-r}}, \quad (32)$$

де $\alpha_{h,s,m}^{(k)}(d)$ – помилка першого роду під час прийняття класифікаційних рішень, обчислена в процесі відновлення гіперсферичного контейнера класу розпізнавання $X_{h,s,m}^{o}$;

$\beta_{h,s,m}^{(k)}(d)$ – помилка другого роду при прийнятті класифікаційних рішень, обчислена в процесі відновлення гіперсферичного контейнера класу розпізнавання $X_{h,s,m}^{o}$ з радіусом $d_{h,s,m}$;

$d_{h,s,m}$ – радіус гіперсферичного контейнера класу розпізнавання $X_{h,s,m}^{o}$;

10^{-r} – достатньо мале число, яке вводиться для уникнення поділу на нуль.

Під час обчислення інформаційного критерію оптимізації (32) на практиці в процесі реалізації алгоритму машинного навчання замість точнісних характеристик використовують їх оцінки:

$$\alpha_{h,s,m}^{(k)}(d) = \frac{K_{1,h,s,m}^{(k)}(d)}{n};\ \ \beta_{h,s,m}^{(k)} = \frac{K_{2,h,s,m}^{(k)}(d)}{n}, \quad (33)$$

де $K_{1,h,s,m}^{(k)}(d)$ – кількість подій, за яких реалізації класу розпізнавання $X_{h,s,m}^{o}$ не належать до свого класу;

$K_{2,h,s,m}^{(k)}(d)$ – кількість подій, за яких «чужі» реалізації помилково належать до класу розпізнавання $X_{h,s,m}^{o}$;

n – обсяг репрезентативної навчальної вибірки.

Після підстановки оцінок точнісних характеристик (33) у формулу (32) одержимо робочу формулу для обчислення інформаційного критерію оптимізації

$$E_{h,s,m}^{(k)}(d) = \frac{1}{n}\{n - [K_{1,h,s,m}^{(k)}(d) + K_{2,h,s,m}^{(k)}(d)]\} \times$$
$$\times \log_2 \frac{2n - [K_{1,h,s,m}^{(k)}(d) + K_{2,h,s,m}^{(k)}(d)] + 10^{-r}}{[K_{1,h,s,m}^{(k)}(d) + K_{2,h,s,m}^{(k)}(d)] + 10^{-r}}. \qquad (34)$$

Нормований критерій оптимізації параметрів машинного навчання подамо у вигляді

$$J_{h,s,m}^{(k)}(d) = \frac{E_{h,s,m}^{(k)}(d)}{E_{\max}},$$

де $E_{\max}$ – максимальне значення критерію (34), яке він набуває у разі підстановки

$$K_{1,h,s,m}^{(k)}(d) = K_{2,h,s,m}^{(k)}(d) = 0.$$

У праці [15] згідно з категорійною моделлю (рис. 1.10) схема інформаційно-екстремального машинного навчанням за ієрархічною декурсивною структурою даних має такий вигляд:

1) обнулення лічильника ярусів $h := 0$;

2) ініціалізація лічильника ярусів $h := h + 1$;

3) обнулення лічильника страт h-го ярусу ієрархічної структури $s := 0$;

4) ініціалізація лічильника страт яруса $s := s + 1$;

5) обнулення лічильника кроків зміни параметра

поля контрольних допусків $\delta_{K,h,s} := 0$;

6) ініціалізація лічильника кроків зміни параметра поля контрольних допусків $\delta_{K,h,s} := \delta_{K,h,s} + 1$;

7) реалізація базового алгоритму машинного навчання, який для *s*-ї страти *h*-го ярусу ієрархічної структури обчислює максимальне значення інформаційного критерію $E_{h,s,m}(d_{h,s,m})$ та визначає оптимальні геометричні параметри контейнерів класів розпізнавання за процедурою

$$d^*_{m,h,s} = \arg \max_{G_E \cap G_d} \overline{E}_{h,s}(d_{h,s,m}),\ m = \overline{1, M}_{h,s},$$

де $M_{h,s}$ – кількість класів розпізнавання *s*-ї страти *h*-го ярусу;

8) якщо $\delta < \delta_H / 2$, то виконується пункт 6, інакше – пункт 9;

9) обчислюється максимальне значення усередненого за алфавітом класів розпізнавання *s*-ї страти *h*-го ярусу ієрархічної структури значення інформаційного критерію $\overline{E}^*_{h,s}$;

10) визначається для *s*-ї страти *h*-го ярусу оптимальне значення параметра поля контрольних допусків на ознаки розпізнавання за процедурою

$$\delta^*_{K.h,s} = \arg \max_{G_{\delta,h,s}} \{ \max_{G_E \cap G_d} \overline{E}_{h,s}(d) \},$$

де $G_{\delta,h,s}$ – область допустимих значень параметра $\delta_{K,h,s}$ поля контрольних допусків на ознаки для класів розпізнавання *s*-ї страти *h*-го ярусу ієрархічної структури.

11) обчислюються оптимальні нижні $A^*_{HK_i}$ та верхні $A^*_{BK_i}$ контрольні допуски:

$$A^*_{HK_i} = y_{h,s,1,i} - \delta^*_{h,s}\ ; A^*_{BK_i} = y_{h,s,1,i} + \delta^*_{h,s},$$

де $y_{h,s,1,i}$ – значення i-ї координати усередненого вектора ознак базового класу $X_{h,s,1}$, щодо якого задається система контрольних допусків;

12) якщо $s \le S_h$, то виконується пункт 4, інакше – пункт 13;

13) якщо $h \le H,$ то виконується пункт 2, інакше – пункт 14;

14) зупин.

За оптимальними геометричними параметрами гіперсферичних контейнерів класів розпізнавання побудовані вирішальні правила, які в предикатній формі мають вигляд

$$(\forall X^o_{m,h,s} \in \Re^{|M|})(\forall x^{(j)} \in \Re^{|M|})\{if\ [(\mu_m > 0)\,\&\,(\mu_m = \max_{\{m\}}\{\mu_m \mid m = \overline{1,M}\}]$$
$$then\ x^{(j)} \in X^o_{m,h,s}\ else\ x^{(j)} \in X^o_{m,h,s}\}, \qquad (35)$$

де $x^{(j)}$ – вектор ознак, який розпізнається;

μ_m – функція належності вектора $x^{(j)}$ гіперсферичному контейнеру класу розпізнавання $X^o_{m,h,s}$, яка визначається за формулою

$$\mu_m = 1 - \frac{d(x^{(j)} \oplus x^*_{h,s,m})}{d^*_{h,s,m}}. \qquad (36)$$

Отже, вектор ознак $x^{(j)}$ належить до того класу із заданого алфавіту відповідної страти, для якого функція належності (36) є додатною і максимальною. Крім того, побудовані в межах геометричного підходу вирішальні правила (35) дозволяють приймати класифікаційні рішення в реальному темпі часу, що є актуальним під час застосування ієрархічної структури даних за великої потужності алфавіту класів розпізнавання.

1.11. Результати моделювання

Як приклад реалізації інформаційно-екстремального машинного навчання СВА розглядалося розпізнавання профілю трьох трафіків, один із яких характеризував нормальний функціональний стан ІКС, а два інших аномальні стани. Спочатку з відкритого джерела інформації було взято два трафіки нормального (клас розпізнавання X_1^o) й аномального (клас розпізнавання X_2^o) профілів. Кожний із цих трафіків характеризувався структурованими векторами із 114 ознак розпізнавання, отриманих за результатами числового аналізу, який обчислює кількісні характеристики пакетів, такі як розмір блоку даних, час відгуку пакета, інтервал між пакетами тощо, і статистичного аналізу. Із структурованих векторів ознак було сформовано вхідну тривимірну навчальну матрицю двох класів розпізнавання, які характеризували, відповідно нормальний та аномальний трафіки.

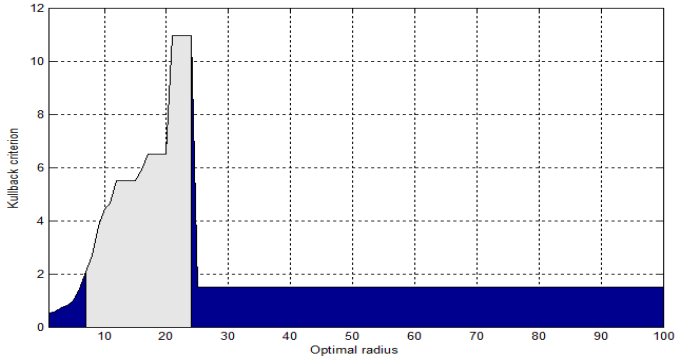
Навчальна матриця для кожного класу розпізнавання складалася із 100 векторів ознак. За навчальною матрицею було визначено усереднені структуровані вектори ознак для кожного класу розпізнавання, аналіз яких довів, що вони не перетинаються в просторі ознак. Цей факт є необхідною,

але недостатньою умовою, тому що класи не перетинаються в просторі ознак. Такий випадок є характерним, наприклад, для аномального трафіка, отриманого внаслідок атаки на сервер. З метою перевірки функціональної ефективності інформаційно-екстремального машинного навчання СВА його результати порівнювалися з результатами багатошарової ШНМ зі зворотним поширенням помилки. За результатами моделювання було встановлено, що повна ймовірність правильного прийняття рішень під час використанні ШНМ дорівнювала $P_t = 0{,}99$. Водночас приблизно така достовірність була отримана в разі використання вирішальних правил, побудованих за результатами інформаційно-екстремального машинного навчання за базовим алгоритмом за заданого, як далі було встановлено, неоптимальному параметрі поля контрольних допусків $\delta = 20$ (тут і далі у відсотках відхилення від номінальних значень ознак). Вирішальні правила, побудовані за результатами інформаційно-екстремального машинного навчання з паралельною оптимізацією контрольних допусків на ознаки розпізнавання, виявилися безпомилковими за оптимального значення параметра $\delta^* = 27$.

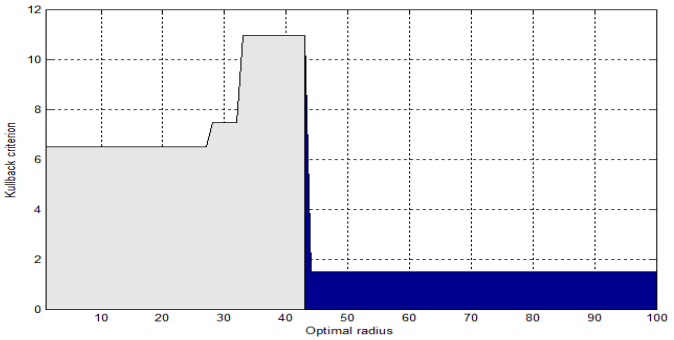
З метою моделювання найбільш складного у статистичному розумінні випадку прийняття класифікаційних рішень кількість класів розпізнавання було збільшено до трьох. Водночас усереднений вектор ознак нового класу розпізнавання X_3^o був сформований так, що він перетинався з аналогічними векторами ознак перших двох класів, що є необхідною і достатньою умовою перетину класів у просторі ознак розпізнавання. Навчальна матриця для нового класу розпізнавання генерувалася за допомогою функції Random у середовищі

Matlab. За результатами машинного навчання ШНМ було отримано асимптотичну усереднену повну ймовірність правильного прийняття рішень $\overline{P_t} = 0{,}73$. Оскільки фактична повна ймовірність правильного прийняття класифікаційних рішень у робочому режимі не може перевищувати отриману за результатами машинного навчання асимптотичну, то такий результат можна визнати непридатним для функціонування СВА.

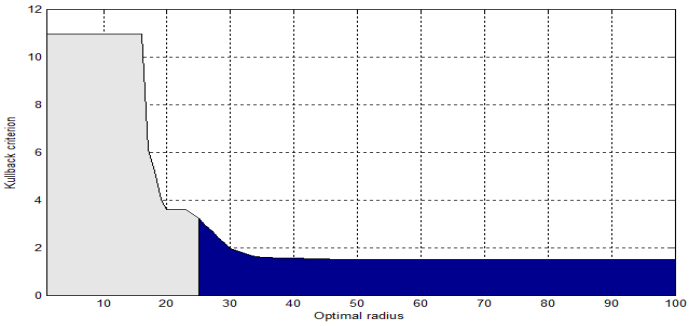
Для сформованого алфавіту з трьох класів розпізнавання було реалізовано інформаційно-екстремальне машинне навчання з оптимізацією контрольних допусків на ознаки розпізнавання. На рисунку 1.11 показано графіки залежності інформаційного модифікованого критерію Кульбака (19) від радіусів контейнерів класів розпізнавання за оптимального параметра поля контрольних допусків $\delta^* = 37$, отриманого за результатами інформаційно-екстремального машинного навчання з паралельною оптимізацією контрольних допусків на ознаки розпізнавання. На рисунку 1.11 на графіках світлою ділянкою позначено робочу (допустиму) область визначення функції інформаційного критерію, в якій перша та друга достовірності перевершують відповідно помилки першого та другого роду.



а



б



в

Рисунок 1.11 – Графіків залежності інформаційного критерію від радіусів контейнерів класів розпізнавання: а – клас X_1^o; б – клас X_2^o; в – клас X_3^o

Аналіз рисунка 1.11 доводить, що усереднене за алфавітом класів розпізнавання значення інформаційного критерію дорівнює $\overline{E}^{*} = 11{,}20$ за максимально граничного значення $\overline{E}_{гран} = 12{,}75$, яке обчислюється за формулою (19) за параметрів $n = 100,\ r = 2$ і $K_{2,m}^{(k)}(d) = K_{3,m}^{(k)}(d) = 0$. Оскільки за значення критерію $\overline{E}^{*} = 11{,}20$ усереднена за алфавітом класів розпізнавання відповідна перша достовірність дорівнювала $\overline{D}_1^{*} = 0{,}98$, а друга достовірність – $\overline{D}_2^{*} = 0{,}94$, то усереднена повна ймовірність прийняття правильних класифікаційних рішень дорівнювала $\overline{P}_t = 0{,}96$, що істотно перевищує результат, отриманий у разі застосуванні ШНМ.

За графіками, поданими на рисунку 1.11, визначають оптимальні геометричні параметри контейнерів класів розпізнавання, необхідні для побудови вирішальних правил (23).

На графіках (рис. 1.11) наявні ділянки типу «плато», на яких функція критерію не є взаємно однозначною. Тому визначення екстремальних значень радіусів контейнерів класів розпізнавання здійснювалося за мінімального значення коефіцієнта нечіткої компактності [10]:

$$\eta = \frac{d_m^{*}}{d(x_m \oplus x_c)}, \qquad (37)$$

де d_m^{*} – оптимальний радіус контейнера класу розпізнавання X_m^{o};

$d(x_m \oplus x_c)$ – міжцентрова відстань у кодових одиницях між найближчими класами X_m^o і X_c^o.

За мінімальних значень коефіцієнта (37) оптимальні радіуси контейнерів класів розпізнавання дорівнюють $d_1^* = 22$ (тут і далі в кодових одиницях) для класу X_1^o, $d_2^* = 31$ для класу X_2^o і $d_3^* = 16$ для класу X_3^o.

Оскільки інформаційний критерій не досягає свого максимального граничного значення, то розбиття простору ознак на класи розпізнавання все ще залишається нечітким, тобто існує перетин класів розпізнавання. Для підвищення функціональної ефективності машинного навчання необхідно збільшити його глибину за допомогою оптимізації додаткових параметрів функціонування СВА, зокрема параметрів формування вхідного математичного опису. Крім того, у разі розширення алфавіту класів розпізнавання доцільним є перехід на інформаційно-екстремальне машинне навчання СВА за ієрархічною структурою даних.

Висновок

Стримувальним чинником створення СККЗ, комп'ютерно-інтегрованих в ІКС різного призначення, є необхідність подолання ускладнень науково-методологічного характеру за інформаційного синтезу СВА, яка є обов'язковою складовою інтелектуальної СККЗ. Як один із перспективних напрямів підвищення функціональної ефективності СВА є застосування ідей і методів ІЕІ-технології, яка ґрунтується на принципі максимізації інформації в процесі машинного навчання.

Основна перевага методів інформаційно-екстремального машинного навчання перед іншими методами, зокрема нейроподібними структурами, полягає в тому, що вони розробляються в межах функціонального

підходу до моделювання механізму когнітивних процесів, притаманних природному інтелекту під час формування та прийнятті класифікаційних рішень. Завдяки цьому навчена в межах IEI-технології система набуває властивості адаптивності до апріорної невизначеності даних і гнучкості під час перенавчання через розширення алфавіту класів розпізнавання. Оскільки загалом розбиття простору ознак розпізнавання є апріорно нечітким, то його дефазифікація відбувається в процесі інформаційно-екстремального машинного навчання за допомогою статистичної корекції детермінованих вирішальних правил, що усуває необхідність застосування апарату нечіткої логіки з притаманними йому методологічними недоліками. Водночас побудовані в межах геометричного підходу вирішальні правила є практично інваріантними до багатовимірності словника ознак розпізнавання, що не потребує екстракції вхідних даних, яка призводить до втрати інформації в ШНМ.

Відомо, що з розвитком інформаційно-комунікаційних технологій одночасно збільшується кількість видів атак, що ставить перед розробниками завдання зменшення впливу потужності алфавіту на функціональну ефективність СВА. Тому важливого значення набуває розроблення методів інформаційно-екстремального машинного навчання за ієрархічною структурою даних. Водночас першочерговим є завдання створення методів автоматичної оптимізації запропонованої в цьому розділі декурсивної ієрархічної структури даних у разі розширення алфавіту класів розпізнавання. Крім того, подальші дослідження повинні бути спрямовані на збільшення глибини машинного навчання, зокрема оптимізацію параметрів формування вхідного математичного опису в межах прогресивної технології DPI (Deep Packet Inspection), методи

моделювання атак і надання СВА властивості самонавчання і прогностичної класифікації.

Автори, не заперечуючи важливість розроблення інших методів інтелектуального аналізу даних, щиро сподіваються на зацікавленість фахівців, аспірантів і студентів інформаційно-екстремальними методами машинного навчання і відкриті до співробітництва в розвитку теоретичних основ інформаційного синтезу здатних навчатися СВА і розробленні відповідних засобів інформаційної технології.

СПИСОК ЛІТЕРАТУРИ

1. Стратегія розвитку кібербезпеки України [Електронний ресурс]. – Режим доступу : https://zakon.rada.gov.ua/laws/show/392/2020#Text.

2. Закон України Про основні засади забезпечення кібербезпеки України [Електронний ресурс]. – Режим доступу : https://zakon.rada.gov.ua/laws/show/2163-19#Text.

3. Системний аналіз та прийняття рішень в інформаційній безпеці : підручник / В. Л. Бурячок, С. В. Толюпа, А. О. Аносов, В. А. Козачок. – Київ : ДУТ, 2015.

4. Інформаційна та кібербезпека: соціотехнічний аспект : підручник / В. Л. Бурячок, В. Б. Толубко, В. О. Хорошко, С. В. Толюпа ; за заг. ред. д-ра техн. наук, професора В. Б. Толубка. – Київ : ДУТ, 2015. – 288 с.

5. Васильев В. И. Интеллектуальные системы защиты информации : учеб. пособие / В. И. Васильев. – 2-е изд., испр. и доп. – Москва : Машиностроение, 2013. – 172 с.

6. Deep Learning Applications for Cyber Security / Alazab, Mamoun, Tang, MingJian (Eds.). – Springer International Publishing, 2019. – 246 p.

7. Das R. Practical al for Cybersecurity / R. Das. – 1st

Edition. – Auerbach Publications, 2021. – 292 p.

8. Dua Sumeet. Data Mining and Machine Learning in Cybersecurity / Sumeet Dua, Xian Du. 1st Edition. – Auerbach Publications, 2011. – 256 p.

9. Медведєв М. Г. Теорія ймовірностей та математична статистика : підручник / М. Г. Медведєв, І. О. Пащенко. – Київ : Ліра-К, 2008. – 536 с.

10. Довбиш А. С. Основи проектування інтелектуальних систем : навчальний посібник / А. С. Довбиш. – Суми : Видавництво СумДУ, 2009. – 171 с.

11. Москаленко В. В. Вступ до інформаційного аналізу і синтезу інфокомунікаційних систем : навч. посіб. / В. В. Москаленко, А. С. Довбиш. – Суми : Сумський державний університет, 2016. – 226 с.

12. Кузьмін І. В. Основи теорії інформації та кодування : підручник / І. В. Кузьмін, І. В. Троцишин, А. І. Кузьмін ; за ред. І. В. Кузьміна. – Вид. 3-тє, переробл. та допов. – Хмельницький : ХНУ, 2009. – 373 с.

13. Кульбак С. Теория информации и статистика / С. Кульбак. – Москва : Наука, 1967. – 406 с.

14. Довбиш А. С. Оптимізація ієрархічної структури даних інтелектуальної системи функціонального діагностування технічного стану складної машини / А. С. Довбиш, В. І. Зимовець, М. В. Бібик // Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. – Харків, 2018. – № 44 (1320). – С. 42–49.

15. Dovbysh A. S. Hierarchical algorithm of the machine learning for the system of functional diagnostics of the electric drive / A. S. Dovbysh, V. I. Zimovets // Advanced information systems and technologies : proceedings of the VI International scientific conference. – Sumy. – May 16–18. – 2018. – Sumy, 2018. – P. 85–88.

РОЗДІЛ 2
КРИПТОСИСТЕМИ НА ОСНОВІ ЛОГАРИФМІЧНИХ ПІДПИСІВ ДЛЯ ПОСТКВАНТОВОЇ КРИПТОГРАФІЇ

Г. З. Халімов, Є. В. Котух, А. О. Теницька, К. О. Зарудна, Р. І. Біленький

Абстракт

У праці запропонований огляд перспективного напряму розвитку криптографічних систем, що належить до постквантової криптографії, на основі математичних структур логарифмічних підписів і покриттів кінцевих груп. Актуальний стан цього напряму й праці останніх років дають підстави припускати, що завдання факторизації елемента кінцевої групи є обчислювально складним і забезпечить необхідний рівень криптографічного захисту в разі використання, комп'ютера, що застосовує квантові принципи обчислень. У розділі наведено основні визначення логарифмічних підписів і покриттів кінцевих груп, їх класифікацію. У рамках аналізу підходів розглянута криптосистема MST_3 із використанням груп Сузукі та узагальнених груп Сузукі.

Вступ

Розглянемо основні позначення, визначення й фундаментальні теореми, що стосуються математичних та обчислювальних аспектів. Репрезентуємо логарифмічні підписи й дамо визначення односторонньої функції на основі логарифмічних підписів.

У запропонованому аналізі поєднуються інструменти з декількох математичних галузей. У праці не розглянуті основні поняття теорії груп, лінійної алгебри,

кінцевих полів і теорії графів, але використані загальноприйняті позначення в рамках цих дисциплін.

Нехай G – кінцева група. Логарифмічний підпис α для групи G є послідовністю підмножин $A_i \subseteq G$ виду

$$\alpha = [A_1, \ \ldots, \ A_s],$$

таких, що для кожного елемента g групи G є лише одна факторизація $(*)$ $g = a_q \cdot a_2 \cdot \ \ldots \ \cdot a_s$, де $a_i \in A_i$ для $i = 1, \ldots, s$. Множини A_i називають блоками. Розмір списку блоків позначають через $r_i := |A_i|$. Для спрощення ми називаємо елементи $A_i \cup \ \ldots \ \cup A_s$ елементами логарифмічного підпису α. За певних умов ми розглядаємо впорядкування елементів блоку, тоді для $k_i = \ 0, \ldots, \ r_i - 1$ позначаємо через a_{ik_i} кожний $(k_i + 1)$-й елемент блоку A_i. Вектор $(r_1, \ \ldots, \ r_s)$ називають типом α, а

$$\ell(\alpha) = \sum_{i \, = \, 1}^{s} r_i -$$

довжиною логарифмічного підпису. Множину логарифмічних підписів групи позначають через $\Lambda(G)$.

Із визначення одержуємо певні властивості логарифмічних підписів. Через одноманітність факторизації (*) ми маємо

$$\prod_{i \, = \, 1}^{s} r_i = |G|,$$

і в такому разі r_i ділить $|G|$ для всіх i . Це також показує, чому зазначені послідовності називають логарифмічними підписами. Логарифмічна функція перетворює добуток на суму, логарифмічні підписи скорочено відображають усі елементи групи довжини $r_1 \ + \ \ldots \ + \ r_s$, де група містить у собі $r_1 \cdot \ \ldots \ \cdot \ r_s$ елементів.

Якщо $e_G \in A_1 \cap \ldots \cap A_s$, тоді ми можемо стверджувати, що α є нормалізованою. З огляду на одиничність факторизації ми навіть маємо $\bigcap_{i \, = \, 1}^{s} A_i = \{e_G\}$

для нормалізованих логарифмічних підписів із більше ніж одним блоком. Легко помітити, що в абелевих $|A_i \cap A_j| \leq 1$ для $i \neq j$.

У праці [1] доведено, що для $|G| = \prod_{j=1}^{t} p_j^{b_j}$ (p_j – просте число) є нижня границя довжини для будь-якого логарифмічного підпису групи G, одержаного так:

$$\ell(\alpha) \geq \sum_{j=1}^{t} b_j \cdot p_j.$$

Логарифмічний підпис α називають мінімальним, якщо $\ell(\alpha)$ досягає нижньої границі, тобто кожен блок має простий ступінь або ступінь 4. У кількох статтях порушене питання існування мінімальних логарифмічних підписів у кінцевих полях [1; 2]. Такі дослідження пропонують мінімальні логарифмічні підписи для всіх кінцевих груп.

Приклад 2.1. Нехай $n \in \mathbb{N}$. Для циклічної групи $(\mathbb{Z}_{2^n}, +)$ послідовність виду

$$\alpha = \left[[0, 2^{n-1}], [0, 2^{n-2}], \ \ldots, \ [0,2], [0,1]\right]$$

є нормалізованим логарифмічним підписом типу $(2, \ldots, 2)$. Обчислення факторизації елемента еквівалентне обчисленню його двійкового відображення, зокрема, якщо $n = 4$, $9 = 1001$ має факторизацію $2^3 + 0 + 0 + 2^0$.

Розглянемо можливість обчислення факторизації елемента групи для зазначеного логарифмічного підпису й певного елемента групи. Для прикладу, атака повним перебором за допомогою пошуку всіх можливих факторизацій, репрезентованих логарифмічним підписом $\alpha = [A_1, \ \ldots, \ A_s]$ групи G, становить $|G| \times (s-1)$ групових операцій у найгіршому разі. Такий перебір знаходить правильну факторизацію для будь-якого логарифмічного підпису, але загалом неможливе.

Приклад 2.1 показує, що для певних логарифмічних підписів легко обчислити факторизації. Для практичного

використання в криптосистемах MST необхідно визначити логарифмічні підписи, для яких факторизація є обчислювально нездійсненною, а також підписи, для яких є ефективні алгоритми розкладання. Здебільшого терміни «прості» й «складні» логарифмічні підписи використовують для позначення різниці між логарифмічними підписами, для яких відповідно обчислювально легко та складно одержати факторизації [3, 4].

Факторизація одного логарифмічного підпису становить постійний час, але, коли ми вивчаємо питання ефективності обчислень для логарифмічних підписів, то розглядаємо сім'ю (G_n, α_n) логарифмічних підписів

$$\alpha_n = \left[A_1^n, \ldots, A_{s_n}^n\right]$$

груп G_n для $n \in \mathbb{N}$. Далі ми припускаємо, що

$$|G_n| \leq |G_{n+1}|$$

і $\alpha_n \neq \alpha_m$ для $n \neq m$.

Крім того, нехай одноманітний опис елементів G_n є таким, що $|g|_2 = \mu$ для всіх $g \in G_n$.

Зауважуємо, що дає $\mu_n \geq log|G_n|$ для всіх $n \in \mathbb{N}$.

Робимо одне базове припущення: для зазначеної сім'ї $(G_n,\ \alpha_n)$ є детермінований алгоритм поліноміальною часу A, такий, що, маючи вхідні значення $\left(a_1,\ \ldots,\ a_{s_n}\right)$ із $A_1^n \times \ldots \times A_{s_n}^n$, алгоритм A обчислює добуток $a_1 \cdot \ldots \cdot a_{s_n}$. Ми ідентифікуємо A з ін'єктованою функцією, що його обчислює.

Визначення 2.2. Для $n \in \mathbb{N}$ нехай α_n буде як у вищенаведеному прикладі. Тоді сім'ю $(G_n,\ \alpha_n)_{n \in \mathbb{N}}$ називають складною, якщо для кожного ймовірного алгоритму поліноміальною часу A' для кожного позитивного полінома p і всіх істотно великих n маємо

$$pr\left(A'(g_n, \alpha_n) = A^{-1}(g_n)\right) < \frac{1}{p(\lceil log|G_n| \rceil)},$$

де g_n позначає випадковий елемент, одноманітно вибраний із G_n.

Для сім'ї складних логарифмічних підписів $(G_n, \alpha_n)_{n \in \mathbb{N}}$ функція A визначає односторонню функцію, як зазначено в [5]. Серед логарифмічних підписів, що не є складними, ми визначаємо ті, які мають ефективні факторизації для всіх елементів групи.

Визначення 2.3. Для $n \in \mathbb{N}$ нехай α_n буде логарифмічним підписом групи G_n. Сім'ю $(G_n, \alpha_n)_{n \in \mathbb{N}}$ називають простою, якщо є алгоритм A'', що, одержуючи на вході елементи $g_n \in G_n$ і α_n, обчислює факторизацію g_n щодо α_n за поліноміальний час.

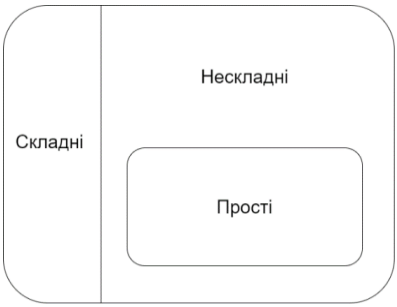


Рисунок 2.1 – Складні й прості логарифмічні підписи групи G

Різницю між «нескладний» і «простий» уперше розглянуто в праці [6], і на відміну від попередніх визначень (у яких ці два поняття еквівалентні), зазначене є більш точним. Логарифмічний підпис, що не є ні простим, ні складним, може бути таким, для якого ми можемо ефективно знайти факторизацію для рівно половини групових елементів. Є алгоритм із заданими g і α_n, що випадково вгадує факторизацію з імовірністю $\frac{1}{|G_n|}$. Отже, алгоритм A' у другому визначенні дає значно

більший шанс знайти факторизацію, як і раніше менший, ніж $\frac{1}{p(\lceil log|G_n| \rceil)}$ (для будь-якого p). Алгоритм A' додатково одержує n на вхід для кодування групи G_n , такий, що A' обчислює G_n. Але твердження, що, маючи α_n на вході, гарантовано A' обчислює групу G_n, є не підтвердженими. Також припустимо, що для вхідних даних $(a_1, \ldots, a_{s_n})$ алгоритм A знає, яку групову операцію застосовувати. Обчислення добутку елементів s_n, кожний із яких був випадково одноманітно вибраним із різних блоків a_n (щодо A), еквівалентне вибору випадкового елемента з G_n.

Відзначимо, що питання наявності варіантів для складних логарифмічних підписів залишається відкритим. Усі логарифмічні підписи, описані в літературі, репрезентовані простими [1, 2, 3, 4, 7]. Криптосистеми MST використовують як складні, так і прості логарифмічні підписи для побудови криптосистем відкритого ключа. Криптосистема MST_1 ґрунтується на складних логарифмічних підписах; MST_3 також використовує прості логарифмічні підписи в елементарних абелевих 2 групах. Отже, нас цікавить структура простих логарифмічних підписів, а також груп, для яких можуть існувати логарифмічні підписи. Далі ми розглянемо групи Сузукі й узагальнені групи Сузукі як такі групи. Для подальшого аналізу груп важливими є три параметри. Для $n \in \mathbb{N}$ логарифмічний підпис $\alpha_n = [A_1^n, \ldots, A_{s_n}^n]$ повністю описаний як

$$|\alpha_n| = \sum_{i=1}^{s} |A_i^n|$$

елементів із групи G_n. Можливо, є коротше репрезентування певних логарифмічних підписів, але загалом нам необхідний

$$\sum_{i=1}^{s} |A_i^n| \cdot \mu_n$$

біт, щоб виразити логарифмічний підпис, у якому елемент G_n репрезентований за допомогою μ_n біт. Нехай

$$r_n := max\{|A_i^n|, \dots, |A_{s_n}^n|\}.$$

Алгоритм факторизації A' у другому визначенні. бере як вхідні дані елемент G_n і логарифмічний підпис α_n для певних $n \in \mathbb{N}$. Довжина цих вхідних даних дорівнює не більше за

$$(s_n \cdot r_n + 1) \cdot \mu_n$$

біт. Отже, ми пропонуємо використовувати три таких значення (s_n, r_n, μ_n) як параметри вимірювання ефективності алгоритму факторизації для α_n. Зазначимо, що ми одержуємо перші два параметри зі структури α_n, а третій параметр є незалежним від α_n та обумовлений лише репрезентуванням елементів G_n.

Зауваження 2.4. Для $n \in \mathbb{N}$ нехай (α_n) буде логарифмічним підписом у G_n. Якщо для всіх $g \in G_n$ факторизацію щодо α_n можна одержати за поліноміальний час із трьома параметрами s_n, r_n, μ_n, тоді сім'я $(\alpha_n)_n$ є простою.

Приклад 2.5. Візьмемо сім'ю логарифмічних підписів у групі $G_n = \mathbb{Z}_{2^n}$ із прикладу 2.1. Три параметри ефективності дорівнюватимуть $s_n = n, r_n = 2, \mu_n = n$. Для заданого елемента $g \in \mathbb{Z}_{2^n}$ за допомогою його двійкового подання $(g_1, \dots, g_n)$, де $g_i \in \{0,1\}$ – його факторизація, яка щодо α дорівнює $(g_1 \cdot 2^{n-1}, \dots, g_n \cdot 1)$, що одержано за допомогою не більше ніж n множень. Отже, маємо лінійний час у n . Тоді $(\alpha_n)_{n \in \mathbb{N}}$ є простою.

У праці розглянуті певні перетворення логарифмічних підписів, зокрема такі, щоб факторизація щодо вихідного логарифмічного підпису була однаково

ефективною щодо перетвореного логарифмічного підпису. Ідея полягає в тому, що алгоритм факторизації одного логарифмічного підпису в певному наборі дає алгоритм факторизації для всіх логарифмічних підписів цього набору. Наведемо стандартний підхід до класифікації логарифмічних підписів відповідно до структури блоків. Ми розглядаємо п'ять перетворень логарифмічних підписів, що не змінюють властивостей належності до простих або складних логарифмічних підписів. Нехай $\alpha = [A_1, \dots, A_s]$ буде логарифмічним підписом групи G. Ми маємо справу з перетвореннями блоків $A_1, \dots, A_s$ логарифмічного підпису α на блоки $B_1, \dots, B_s$, що є результуючою послідовністю $\beta = [B_1, \dots, B_s]$ і також репрезентують логарифмічний підпис G.

Перетворення 2.6. Нехай φ буде автоморфізмом G і $B_i = \varphi(A_i)$ для $i = 1, \dots, s$. Тоді β також є логарифмічним підписом. І якщо $a_1 \cdot a_2 \cdot \dots \cdot a_3$ є факторизацією елементів $g \in G$, $\varphi(a_1) \cdot \varphi(a_2) \cdot \dots \times$ $\times \varphi(a_s)$ є факторизацією $\varphi(g)$ щодо β.

Перетворення 2.7. Нехай $g_0, \dots, g_s$ будуть елементами групи G і $B_i = g_{i-1}^{-1} A_i g_i$ для $i = 1, \dots, s$. Тоді послідовність β також є логарифмічним підписом G, що називають трансляцією α. Якщо $g_0 = g_s = e_G$, β – сендвіч α. Зазначимо, що, якщо $a_1 \cdot a_2 \cdot \dots \cdot a_s$ є факторизацією елементів $g \in G$, то $g_0^{-1} a_1 g_1 \cdot \dots \cdot g_{s-1}^{-1} a_s g_s$ є факторизацією $g_0^{-1} g g_s$ щодо β.

Необхідно пам'ятати, що в абелевих групах блоки трансляції β від α мають вигляд $B_i = A_i + h_i$ для елементів $h_1, \dots, h_s$ групи G, а отже, будь-яка факторизація елемента $g \in G$ щодо α миттєво дає факторизацію $g +$ $+ \sum_{i=1}^{s} h_i$ щодо β.

Перетворення 2.8. Для $i = 1, \ldots, s$ нехай π_i буде перестановкою в S_{r_i} і $B_i = \left[a_{i\pi_i(1)}, \ldots, a_{i\pi_i(r_i)}\right]$ для $j = 1, \ldots, r_i$, тобто елементи блоку B_i – перестановка елементів блоку A_i. Тоді β також є логарифмічним підписом. І, якщо $a_{1k_1} \cdot a_{2k_2} \cdot \ldots \cdot a_{sk_s}$ – факторизація елемента $g \in G$, $a_{1\pi_1(k_1)} \cdot a_{2\pi_2(k_2)} \cdot \ldots \cdot a_{s\pi_s(k_s)}$ – факторизація g щодо β.

Перетворення 2.9. Тепер нехай G буде абелевою групою, π – перестановкою в S_s і $B_i = A_{\pi(i)}$, тобто $b_{ij} = a_{\pi(i)_j}$. Тоді послідовність β є логарифмічним підписом групи G. Її також називають перетворенням блочної перестановки α. Зазначимо, що, якщо $a_{1i_1} + a_{2i_2} + \ldots + a_{si_s}$ є факторизацією елемента $g \in G$ щодо α, $a_{\pi(1)(i_1)} \cdot a_{\pi(2)(i_2)} \cdot \ldots \cdot a_{\pi(s)(i_s)}$ є факторизацією g щодо β. Крім того в не абелевих групах β може не бути логарифмічним підписом.

Перетворення 2.10. Для певних $j \in \{1, \ldots, s-1\}$ нехай

$$B_j = A_j \cdot A_{j+1} = \left[x \cdot y | x \in A_j, y \in A_{j+1}\right]$$

та $B_i = A_i$ для $i = 1, \ldots, s-1$ і $i \neq j, j+1$. Послідовність $\beta = [B_1, \ldots, B_{s-1}]$ є логарифмічним підписом, одержаним з α в результаті перетворення – злиття двох блоків. І, якщо $a_1 \cdot a_2 \cdot \ldots \cdot a_s$ є факторизацією $g \in G$ щодо α, $a_1 \cdot a_{j-1} \cdot a \cdot a_{j+2} \ldots \cdot a_s$, де $a = a_j \cdot a_{j+1}$ є факторизацією g щодо β. Зворотну операцію називають перетворенням розподілу. Для кожного з п'яти наведених перетворень ми описали, як факторизація елемента щодо логарифмічного підпису негайно приводить до факторизації щодо перетвореного логарифмічного підпису. Якщо ми розглядаємо ці перетворення для сімей логарифмічних підписів (α_n), то легко помітити, що перемикання між алгоритмами

факторизації (α_n) і перетвореннями (α_n) виконується за поліноміальний час, якщо перетворення є відомим або ефективно обчислюваним. Це справедливо й для нормалізації.

Визначення 2.11. Нехай (α_n) і (β_n) є сім'ями логарифмічних підписів для груп G_n з параметрами r_n, s_n, μ_n для (α_n). Тоді ми стверджуємо, що (α_n) перетворюється на (β_n), якщо (β_n) можна обчислити з (α_n) за допомогою $1 - 5$(повторюваних) перетворень за поліноміальний час для r_n, s_n, μ_n. Зазначимо, що в такому разі кількість перетворень між (α_n) і (β_n) є кінцевою.

$T -$ множина (α_n), визначена як

$T(\alpha_n) = \{(\beta_n) | \beta_n \Lambda (G_n)$ і α_n перетвориться для всіх $n\}$.

Приклад 2.12. Візьмемо логарифмічний підпис

$$\alpha_n = \left[[0, 2^{n-1}], \dots, [0,2], [0,1]\right]$$

групи $\mathbb{Z}_{2^n}$ із прикладу 2.1. Для $n > 4$ нехай

$$\beta_n = \left[[1, 5], [0, 1, 2, 3], [0, 8], [0, 16], \dots, [0, 2^{n-1}]\right]$$

І $\gamma_n = \left[[0, 1, 2, 3, \dots, 2^n - 1]\right]$. Тоді $(\beta_n) \in T(\alpha_n)$ і $(\gamma_n) \notin T(\alpha_n)$.

Визначення 2.13. Нехай $g_0 = 1$ і $g_i = \left(\prod_{j=1}^{i} a_{j1}\right)^{-1}$ для $i = 1, \dots, s$. За допомогою трансляції α в $g_0, \dots, g_s$ ми одержуємо логарифмічний підпис β, у якому

$$b_{i1} = g_{i-1}^{-1} a_{i1} g_i = \left(\left(\prod_{j=1}^{i-1} a_{j1}\right)^{-1}\right)^{-1} a_{i1} \left(\prod_{j=1}^{i} a_{j1}\right)^{-1} =$$

$$= \left(\prod_{j=1}^{i} a_{j1}\right) \cdot \left(\prod_{j=1}^{i} a_{j1}\right)^{-1} = 1,$$

тобто перший елемент у кожному блоці є нейтральним. Ми вважаємо β нормалізацією α.

В абелевих групах ми можемо нормалізувати логарифмічний підпис, застосовуючи трансляцію $B_i = A_i - a_{i1}$. Тоді перший елемент кожного блоку дорівнюватиме $a_{i1} - a_{i1} = 0$.

Інші класи мають стандартне позначення. Нехай G буде кінцевою групою. Ми називаємо точно-поперечним логарифмічний підпис $\alpha \in \Lambda(G)$, якщо такий ланцюг підгруп виду

$$e_G = G_0 < G_1 < \ \dots \ < G_{s-1} < G_s = G,$$

що A_i є поперечною групи $G_{i\ -\ 1}$ у G_i, тобто $G_{i-1}A_i = G_i$ та $|G_{i-1}||A_i| = |G_i|$. Зазначимо, що блок $A_1 = G_1$ є підгрупою G. Відповідний клас позначають через ε. Якщо α є сендвічем точно поперечної логарифмічної групи, α називають поперечною. Клас поперечних логарифмічних підписів позначимо через T_{LS}.

Усі інші логарифмічні підписи належать до NT_{LS} – класу не поперечних логарифмічних підписів. Ми описуємо два підкласи NT_{LS}. Якщо жодний із блоків не є підмножиною (нетривіальною) підгрупи G, логарифмічний підпис є елементом TNT_{LS} – класу абсолютно не поперечних логарифмічних підписів. Клас TA_{LS} повністю аперіодичних логарифмічних підписів містить усі логарифмічні підписи, що не мають навіть періодичного блоку, тобто об’єднані підмножиною G. Отже, маємо $TA_{LS} \subseteq TNT_{LS} \subseteq NT_{LS}$ і $\varepsilon \subseteq T_{LS}$. На рисунку 2.2 наведені класи логарифмічних підписів.

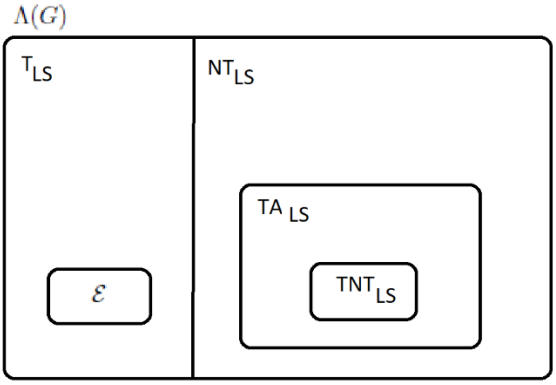


Рисунок 2.2 – Стандартні класи логарифмічних підписів групи G

Необхідно пам'ятати, що властивість простоти для точно поперечних логарифмічних підписів породжується в певних групах.

Визначення 2.14. Нехай (α_n) – сім'я точних логарифмічних підписів, а тестування на належність підгрупи до групи G_n проведено за поліноміальний час μ_n. Тоді (α_n) є простим.

Для простішого читання опустимо індекс n. Нехай

$$e_G = G_0 < G_1 < \ \dots \ < G_{s-1} < G_s = G$$

буде ланцюгом підгруп відповідних логарифмічних підписів $\alpha = [A_1, \ \dots \ , A_s]$. Нехай $g \in G$. Є саме одна факторизація g, і вона містить $a_1, \ \dots \ , a_s$, де

$$g \cdot a_s^{-1} \cdot \dots \cdot a_{k+1}^{-1} = a_1 \cdot \dots \cdot a_k \in G_k$$

для $k = 1, \ \dots \ , s-1$ та $a_1 \cdot \dots \cdot a_s = g$. Наступний простий алгоритм знаходить факторизацію g щодо α. Алгоритм є детермінованим. Буде потрібно $O(r_n \cdot s_n)$ раундів, у кожному з яких необхідні одне множення, одне звернення й одне тестування на належність до підгрупи, виконувані за поліноміальний час μ_n, s_n, r_n. Крім того, зрозуміло, що, якщо μ_n, s_n, r_n є поліноміальними в

$\lceil log|G_n| \rceil$ та (α_n) є простою сім'єю нормалізованих точно поперечних логарифмічних підписів, тоді існує ефективний тест на належність до підгрупи для всіх підгруп G_i групи G_n: $g \in G_i$, лише якщо у факторизації $g = a_1 \cdot \ldots \cdot a_s$ одержуємо результат $a_{i+1} = \ldots = a_s = e_G$. Можливо, непоперечні й навіть абсолютно не поперечні логарифмічні підписи є гарними варіантами, щоб бути складними. Проте в праці [8] було доведено, що досить легко побудувати прості сім'ї логарифмічних підписів для симетричних груп, що чергуються, які належать до класу TNT_{LS}. Отже, класи не мають критеріїв, щоб розрізняти прості й складні логарифмічні підписи.

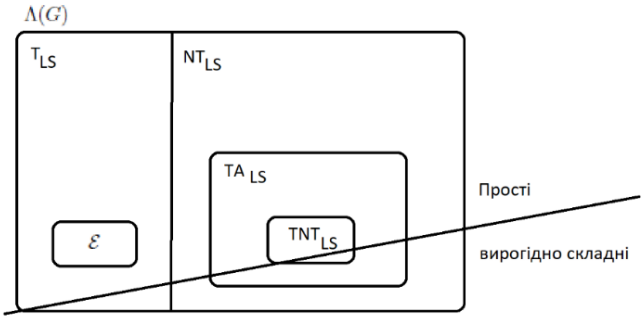


Рисунок 2.3 – Розподіл простих логарифмічних підписів серед усіх логарифмічних підписів загальної групи G

Якщо немає ефективного тесту на належність до підгрупи для груп G_n, у цих групах також будуть складні логарифмічні підписи, що є поперечними.

Очевидно, що для груп, для яких є ефективний тест на належність до підгрупи, точно поперечні логарифмічні підписи є простими. Для формалізації цього зауваження використовуємо наступне визначення.

Визначення 2.15. Для $n \in \mathbb{N}$ нехай P_n буде булевим предикатом, що можна застосовувати до будь-якого

логарифмічного підпису групи G_n. Ми стверджуємо, що P є властивістю, яка породжує простоту для G_n, якщо всі сім'ї $(\alpha_n)_{n\in\mathbb{N}}$ логарифмічних підписів (G_n), для яких правильне твердження: $P(\alpha_n)$ правдиве для всіх $n \in \mathbb{N}$, де n є простим.

Приклад 2.16. Переглянемо знову приклад 2.1. Для $n \in \mathbb{N}$ нехай P_n =«кожним блоком форми $[0, 2^i]$ для $0 \leq i \leq n''$». Тоді ми стверджуємо в прикладі 2.5, що P є властивістю породження простоти для $\mathbb{Z}_{2^n}$.

У групах, для яких є ефективний тест на належність до підгрупи, належність до поперечних логарифмічних підписів також є властивістю породження простоти.

Криптосистеми сім'ї MST

Науковці С. Магліверас, Д. Стінсон і Т. ван Транг розробили два підходи до побудови криптосистем відкритого ключа, названі MST_1 та MST_2 [3]. Відповідно до MST_1 використовують логарифмічні підписи, а MST_2 базується на покриттях, які відрізняються від логарифмічних підписів тим, що факторизація для них не є унікальною. У третій версії криптосистеми MST_3, презентованій В. Лемпкеном, Т. ван Трангом, С. Магліверасом та В. Вейєм у [4], використовують прості логарифмічні підписи, такі як покриття. У праці ми пропонуємо огляд початкового налаштування сценарію й генерації початкових вихідних даних, шифрування та дешифрування, реалізованих у двох криптосистемах, що базуються на логарифмічних підписах.

Для опису систем використані певні позначення. Логарифмічний підпис породжує ступінь елементів групи G. Це потрібно для визначення функції: нехай $\alpha = [A_1, \ldots, A_s]$ буде логарифмічним підписом типу $(r_1, \ldots, r_s)$ для групи G. Якщо ми розглядаємо впорядковані блоки, то одержуємо бієкцію з $\mathbb{Z}_{|G|}$ у G.

Спочатку беремо канонічну бієкцію, що ідентифікує $\mathbb{Z}_{r_1} \times \ldots \times \mathbb{Z}_{r_s}$ з $\mathbb{Z}_{|G|}$:

$$\tau: \mathbb{Z}_{r_1} \times \ldots \times \mathbb{Z}_{r_s} \to \mathbb{Z}_{|G|},$$
$$(k_1, \ \ldots \ , k_s) \mapsto \sum_{i=1}^{s} \left(\prod_{j=1}^{i-1} r_j\right) \cdot k_i.$$

Тоді α породжує функцію

$$\alpha: \mathbb{Z}_{|G|} \to G,$$
$$x \mapsto \prod_{i=1}^{s} a_{ij_i},$$

де $\tau^{-1}(x) = (j_1, j_2, \ \ldots \ j_s)$.

Зазначимо, що з τ і τ^{-1} можна ефективно обчислити функцію α, але, щоб мати змогу ефективно обчислити $\alpha^{-1}(g)$ для будь-якого елемента $g \in G$, логарифмічні підписи α повинні бути простими.

Визначимо добуток двох логарифмічних підписів для деякого довільного нефіксованого простого логарифмічного підпису η групи G. Для $\alpha, \beta \in \Lambda(G)$ і $x \in \mathbb{Z}_{|G|}$ одержимо

$$\alpha \cdot \beta(x) = \alpha\left(\eta^{-1}\left(\beta(x)\right)\right).$$

У праці [9] доведено: для нетривіальної кінцевої групи G, що не є циклічною групою порядку простого числа або квадратом простого числа, кожен логарифмічний підпис G є добутком поперечних логарифмічних підписів. У праці [10] запропонована спеціальна версія цієї теореми. Загальний підхід до породження схеми шифрування наведено нижче.

Вихідні дані сценарію

Аліса вибирає кінцеву групу G і генерує:

1) простий логарифмічний підпис η групи G;

2) k точно поперечних логарифмічних підписів $\theta_1, \ \ldots \ , \theta_k$, таких, що $\alpha = \theta_1 \cdot \ \ldots \ \cdot \theta_k$ є складним логарифмічним підписом.

Вона публікує свій публічний ключ $(\alpha, \ \eta)$, а $\theta_1, \ \ldots \ , \theta_k$ тримає в секреті як приватний ключ.

Шифрування

Якщо Боб хоче надіслати повідомлення $x \in \mathbb{Z}_{|G|}$ Алісі, він:

1) обчислює, $c = \alpha\eta^{-1}(x)\left(\in \mathbb{Z}_{|G|}\right)$;

2) надсилає c Алісі.

Дешифрування

Аліса знає факторизацію α і також може обчислити

$$x = \theta_k^{-1} \cdot \ldots \cdot \theta_1^{-1}(c).$$

У праці [8] застосована частково інтервальна атака, щоб продемонструвати, що логарифмічні підписи в класі TNT_{LS} загалом не є безпечними складовими криптосистеми MST_1.

Далі в праці [7] побудовані прості сім'ї логарифмічних підписів для симетричних і груп, що чергуються, які належать до класу TNT_{LS}. Криптосистема MST_3 поєднує логарифмічні підписи й покриття.

Нехай G буде кінцевою не абельовою групою з нетривіальним центром Z, таким, що G не можливо поділити на Z, тобто немає підгрупи $H < G$ із $H \cap Z = e_G$, такої, що $G = H \cdot Z$. Припустимо також, що Z є істотно великим. Отже, вирішення проблеми вичерпного пошуку є обчислювально неможливим у Z.

Вихідні дані сценарію

Аліса вибирає велику групу G, що була раніше описаною та генерує:

1) простий логарифмічний підпис $\beta = [B_1, \ldots, B_s]: = \left(b_{i,\,j}\right)$ типу $(r_1, \ldots, r_s)$ для Z;

2) випадкове накриття $\alpha = [A_1, \ldots, A_s]: = \left(a_{i,\,j}\right)$ такого самого типу, що й для певної підмножини ∂ групи G, тобто такого, що $A_1, \ldots, A_s \subseteq G \backslash Z$.

Пізніше вона вибирає елементи $t_0, t_1, \ldots, t_s \in G \backslash Z$ та обчислює:

1) $\tilde{\alpha} = \left[\tilde{A}_1,\ \tilde{A}_2,\ \ldots,\ \tilde{A}_s\right]$, де $\tilde{A}_i = t_i^{-1}{}_{-1} A_i t_i$ для $i = 1,\ \ldots,\ s$;

2) $\gamma = \left(h_{ij}\right) = \left(b_{ij}\tilde{a}_{ij}\right)$.

Аліса публікує свій публічний ключ (α, γ) та зберігає $\left(\beta, (t_0, \ldots, t_s)\right)$ – секретний ключ.

Шифрування

Якщо Боб хоче надіслати повідомлення $x \in \mathbb{Z}_{|Z|}$ Алісі, то він:

1) обчислює $y_1 = \tilde{\alpha}(x)$ і $y_2 = \gamma(x)$;

2) надсилає (y_1, y_2) Алісі.

Дешифрування

Маємо $y_2 = \gamma(x) = \breve{\beta}(x)t_0^{-1}\breve{\alpha}(x)t_s = \breve{\beta}(x)t_0^{-1}y_1 t_s$.

Далі, маючи секретний ключ, Аліса може обчислити

$$x = \breve{\beta}^{-1}(y_2 t_s^{-1} y_1^{-1} t_0).$$

Наразі було запропоновано використовувати Сузукі 2-гу групу ступеня $2^{2m} (n \in \mathbb{N})$ для криптосистеми MST_3. У криптосистемі MST_3 із використанням Сузукі 2-ї групи перша частина секретного ключа є логарифмічним підписом центра зазначеної групи. Центр Сузукі 2-ї групи є елементарною абелевою підгрупою. У праці [3] доведено, що криптосистема небезпечна, якщо є логарифмічним підписом, у якому кожний блок – підгрупа. Згідно з [7] криптосистема небезпечна, якщо використовувані логарифмічні підписи одержані з точно поперечних логарифмічних підписів за допомогою перетворень одного на п’ять.

Розвитком використання Сузукі 2-ї групи як основи для створення й використання логарифмічних підписів стала праця [11], у якій розглянуті узагальнені групи.

Налаштування сценарію

Для вироблення ключової пари Алісі знадобиться велика група з $A_l(n,\ \theta) = \{S(a_1,\ a_2,\ \ldots,\ a_l)|a_i \in F_q\}q = 2^n$ із центром Z:

1) Аліса обирає прості логарифмічні підписи типу $\beta_k = [B_{1(k)},\ \ldots,\ B_{s(k)}] = (b_{ij})_k = S(0,\ \ldots,\ 0, b_{ij_{(l/2\ +\ k)}},\ 0,\ \ldots,\ 0), (r_{1(k)},\ \ldots,\ r_{s(k)})i = \overline{1,\ s},\ j = \overline{1, r_{i(k)}}\ b_{ij_{(l/2\ +\ k)}} \in F_q k = \overline{1,\ l/2}$.

Простий логарифмічний підпис визначають як бієкцію й факторизоване відображення $\beta_k(R)$;

2) Аліса вибирає випадкове покриття

$$\alpha_k = [A_{1(k)},\ \ldots,\ A_{s(k)}] = (a_{ij})_k = S(0,\ \ldots,\ 0, a^{(1)}_{ij(k)},\ 0,\ \ldots,\ 0, a^{(2)}_{ij(l/2\ +\ k)}, 0,\ \ldots,\ 0)$$

того самого типу, що й β, у якому $a_{ij} \in A_l(n, \theta)$ $a^{(1)}_{ij(k)},\ a^{(2)}_{ij(k)} \in F_q\backslash\{0\}, i = \overline{1,\ s}, j = \overline{1, r_{i(k)}}, k = \overline{1,\ l/2}$;

3) Аліса вибирає $t_{i(k)} = S(t_{i1(k)}, ..., t_{il(k)})$ $t_{ij(k)} \in F^{\times} i = \overline{0,\ s} j = \overline{1,\ l}\ k = \overline{1, l/2}\, t_{s\ (v)} = t_{0(v\ +\ 1)}$ і нехай $t_{0(k)},\ t_{1(k)},\ \ldots,\ t_{s(k)} \in A_l(n,\ \theta)\backslash Z v = \overline{1,\ l/2 - 1}$;

4) Аліса будує гомоморфізм певним виразом $f(S(a_1,\ \ldots,\ a_l)) = S(0,\ \ldots,\ 0,\ a'_{\frac{l}{2}\ +\ 1} = a_1,\ \ldots,\ a'_l = a_{\frac{l}{2}})$;

5) Аліса обчислює

$$\gamma_k = [h_{1(k)},\ \ldots,\ h_{s(k)}] = (h_{ij})_k = t^{-1}_{(i\ -\ 1)(k)} f\left((a_{ij})_k\right)(b_{ij})_k t_{i(k)} i = \overline{1,\ s} j = \overline{1,\ r_i} k = \overline{1,\ l/2},$$

де $f\left(\left(a_{ij}\right)_k\right)\left(b_{ij}\right)_k = S\left(0,\ \ldots,\ 0, \left(a_{ij}^{(1)}\right)_{l/2\,+\,k} + \right.$

$\left. + \left(b_{ij}\right)_{l/2\,+\,k}, 0,\ \ldots,\ 0\right)$;

6) у результаті Аліса одержує $[f, (\alpha_k,\ \gamma_k)]$ – публічний ключ, $\left[\beta_k, \left(t_{0_{(k)}},\ \ldots,\ t_{s(k)}\right)\right] k = \overline{1,\ l/2}$ – секретний ключ.

Шифрування

1. Боб вибирає випадкове $R = (R_1,\ R_2,\ \ldots,\ R_{l/2})$ $R_1,\ R_2,\ \ldots,\ R_{l/2_l} \in \mathbb{Z}_{|F_q|}$.
2. Боб обчислює

$$y_1 = \alpha^{'(R)} \cdot x =$$

$$= \alpha_1^{'(R_1)} \cdot \alpha_2^{'(R_2)} \cdot \alpha_3^{'(R_3)} \ \cdots\ \alpha_{\frac{l}{2}}^{'}\left(R_{\frac{l}{2}}\right) * \cdot x =$$

$$= S\Big(a_1^{(1)}(R_1), a_2^{(1)}(R_2) + *,\ \ldots,\ a_{\frac{l}{2}}^{(1)}\left(R_{\frac{l}{2}}\right)\ *,$$

$$a_{l/2\,+\,1}^{(2)}(R_1) + x_{l/2\,+\,1} + *,\ \ldots,\ a_l^{(2)}\left(R_{l/2}\right) + x_l + *\Big).$$

Компоненти формули визначають перехресними розрахунками в груповій операції добутку.

3. Боб обчислює

$$y_2 = \gamma^{'(R)} = \gamma_1^{'(R_1)} \cdot \gamma_2^{'(R_2)} \ \cdots\ \gamma_{\frac{l}{2}}^{'}\left(R_{\frac{l}{2}}\right) =$$

$$= S(*, *,\ \ldots, *, a_{\frac{l}{2}\,+\,1}^{(1)}(R_1) + \beta_{\frac{l}{2}\,+\,1}(R_1)\ +$$

$$+ *, \ldots,\ a_l^{(1)}\left(R_{\frac{l}{2}}\right) + \beta_l\left(R_{\frac{l}{2}}\right) + *).$$

У такому разі компоненти (*) обчислюють перехресними розрахунками в груповій операції добутку $t_{0_{(k)}}, \ldots,\ t_{s(k)}, k = \overline{1,\ l/2}$.

4. Боб одержує на виході $(y_1,\ y_2)$

Дешифрування

Щоб розшифрувати повідомлення x Алісі необхідно відновити випадкові числа

$$R = (R_1,\ R_2,\ \ldots,\ R_{l/2}).$$

Параметр $a_1^{(1)}(R_1)$ відомий із y_1 як перший і він належить до $l/2+1$ компоненти y_2, тому що $a_{l/2\,+\,1}^{(1)}(R_1) = a_1^{(1)}(R_1)$.

1. Аліса обчислює

$$D^{(1)}\left(R_1,\ R_2,\ \ldots,\ R_{\frac{l}{2}}\right) = t_{0(1)} \cdot y_2 t^{-1}_{s\left(\frac{l}{2}\right)} =$$

$$= S(0,\ \ldots,\ 0, a^{(1)}_{\frac{l}{2}\,+\,1}(R_1) + \beta_{\frac{l}{2}\,+\,1}(R_1),\ \ldots,$$

$$a_l^{(1)}\left(R_{l/2}\right) + \beta_l\left(R_{l/2}\right)).$$

і

$$D^*(R) = D^{(1)}\left(R_1,\ R_2,\ \ldots,\ R_{\frac{l}{2}}\right) f(y_1) =$$

$$= S(0,\ \ldots,\ 0, \beta_{\frac{l}{2}\,+\,1}(R_1), a^{(1)}_{\frac{l}{2}\,+\,2}(R_2) +$$

$$+\beta_{l/2\,+\,2}(R_2) + *,\ \ldots).$$

2. Аліса відновлює R_1 із $\beta_{l/2\,+\,1}(R_1)$ за допомогою $\beta_{l/2\,+\,1}(R_1)^{-1}$, тому що β – це простий логарифмічний підпис.

Для подальших розрахунків необхідно видалити компоненти масивів $\alpha_1'(R_1)$ і $\gamma_1'(R_1)$із шифротексту (y_1, y_2):

3. Аліса продовжує обчислення:

$$y_1^{(1)} = \alpha_1^{'(R_1)^{-1}} \cdot y_1 = \alpha_2^{'(R_2)} \cdot \alpha_3^{'(R_3)}\ \cdots\ \alpha^{'}_{\frac{l}{2}}\left(R_{\frac{l}{2}}\right) \cdot x\ =$$

$$= S\Big(0,\ a_2^{(1)}(R_2),\ a_3^{(1)}(R_3)\ + *,\ \ldots,\ a^{(1)}_{\frac{l}{2}}\left(R_{\frac{l}{2}}\right) + *,$$

$$x_{\frac{l}{2}\,+\,1} + *,\ a^{(2)}_{\frac{l}{2}\,+\,2}(R_2) + x_{\frac{l}{2}\,+\,2} + *,\ \ldots,$$

$a_l^{(2)}\left(R_{l/2}\right) + x_l + *)$

і

$y_2^{(1)} = \gamma_1^{'(R_1)^{-1}},\ y_2 = \gamma_2^{'(R_2)}\ \cdots\ \gamma_{\frac{l}{2}}'\left(R_{\frac{l}{2}}\right) =$

$= S(*,\ *,\ \ldots,\ *,\ a_{\frac{l}{2}+2}^{(1)}(R_2) + \beta_{\frac{l}{2}+2}(R_2) + *,\ \ldots,$

$a_l^{(1)}\left(R_{l/2}\right) + \beta_l\left(R_{l/2}\right) + *).$

Повторює розрахунки:

$D^{(2)}\left(R_2,\ \ldots,\ R_{\frac{l}{2}}\right) = t_{0(2)} \cdot y_2 t^{-1}_{s\left(\frac{l}{2}\right)} =$

$= S(0,\ \ldots,\ 0, a_{\frac{l}{2}+2}^{(1)}(R_2) + \beta_{\frac{l}{2}+2}(R_2),\ \ldots,$

$a_l^{(1)}\left(R_{l/2}\right) + \beta_l\left(R_{l/2}\right))$

та

$D^*(R) = D^{(2)}\left(R_2,\ \ldots,\ R_{\frac{l}{2}}\right) f\left(y_1^{(1)}\right) =$

$= S(0,\ \ldots,\ 0, \beta_{\frac{l}{2}+2}(R_2),\ a_{\frac{l}{2}+3}^{(1)}(R_3) +$

$+\beta_{l/2+3}(R_3) + *,\ \ldots).$

4. Аліса відновлює R_2 із $\beta_{l/2+2}(R_2)$ за допомогою$\beta_{l/2+2}(R_2)^{-1}$.

Ітеративно повторюючи обчислення після $l/2$ кроків, Аліса одержує відновлення $R = (R_1,\ R_2,\ \ldots,\ R_{l/2})$ і повідомлення xвід y_1.

У праці [11] приведено практичний приклад, що підтверджує коректність обчислень. Алгоритм використовує узагальнену групу Сузукі $G = A_l(n, \theta)$ і логарифмічний підпис для типу $(r_1,\ \ldots,\ r_s)$над кінцевим полем F_q, $q = 2^n$. Припускаємо, що значення r_iприблизно дорівнюють $r_i = 2^{n/s}$. Раніше визначено, що розмір логарифмічного підпису має оцінку $V = ls|A_l(n,\ \theta)|^{1/ls}$. Для $q = 2^{64}$, $|A_l(n,\ \theta)| = 2^{512}$, $l = 8$ та

$s = 2^3,\ 2^4,\ 2^5$одержуємо відповідно з 64 – бітних рядків $V = 2^{14},\ 2^{11},\ 2^{10}$. Запропонована конструкція криптосистеми MST_3 на основі узагальненої групи Сузукі забезпечує потенційно більш вищу криптостійкість та оптимізує розмір ключових даних. Відмінність від відомої конструкції MST_3 полягає в ітеративному відновленні ключа у великій нормальній підгрупі узагальненої групи Сузукі.

СПИСОК ЛІТЕРАТУРИ

1. Gonzáles Vasco M. I. On minimal length factorizations of _nite groups / M. I. Gonzáles Vasco, M. Rotteler, R. Steinwandt // Experimental Mathematics. – 2003. – Vol. 12 (1).– P. 1–12.

2. Singhi N. Minimal logarithmic signatures for finite groups of Lie type / N. Singhi, N. Singhi, S. Magliveras // Designs, Codes and Cryptography. – 2010. – Vol. 55 (2). – P. 243–260.

3. Magliveras S. New approaches to designing public key cryptosystems using one-way functions and trap-doors in nite groups / S. Magliveras, D. Stinson, T. van Trung // Journal of Cryptology. – 2002. – Vol. 15. – P. 285–297.

4. Lempken W. A public key cryptosystem based on non-abelian _nite groups / W. Lempken, T. van Trung, S. S. Magliveras, W. Wei // Journal of Cryptology. – 2009. – Vol. 22 (1). – P. 62–74.

5. Goldreich O. Foundations of Cryptography: Basic Tools / O. Goldreich // Cambridge University Press. – 2001.

6. Nuss A. On group based public key cryptography [Electronic resource] : Phd thesis. – Access mode : http://nbn-resolving.de/urn:nbn:de:bsz:21-opus-63659.

7. Blackburn S. R. Cryptanalysis of the MST 3 public key cryptosystem / S. R. Blackburn, C. Cid, C. Mullan //

Journal of Mathematical Cryptology. – 2009. – Vol. 3 (4). – P. 321–338.

8. Weak keys in MST / J. Bohli, M. I. Gonzáles Vasco, C. J. M. Martínez, R. Steinwandt // Designs, Codes and Cryptography. – 2005. – Vol. 37 (3). – P. 509–524.

9. Caranti A. The round functions of cryptosystem PGM generate the symmetric group / A. Caranti, F. D. Volta // Designs, Codes and Cryptography. – 2006. – Vol. 38 (1). – P. 147–155.

10. Magliveras S. Algebraic Properties of Cryptosystem PGM / S. Magliveras, N. D. Memon // Journal of Cryptology. – 1992. – Vol. 5 (3). – P. 167–183.

11. Khalimov G. MST_3 Cryptosystem Based on a Generalized Suzuki 2-Groups [Electronic resource] / G. Khalimov, Y. Kotukh, S. Khalimova. – Access mode : http://ceur-ws.org/Vol-2711/paper1.pdf.

РОЗДІЛ 3
ПРОБЛЕМИ УПРАВЛІННЯ РИЗИКАМИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ В АВТОМАТИЗОВАНИХ СИСТЕМАХ

Є. А. Лавров, Я. І. Чибіряк

Проблема убезпечення автоматизованих систем у сучасному світі ускладняється відсутністю єдиної методичної бази, що дозволяє адекватно оцінювати загрози інформаційних ресурсів, а також ступінь захищеності систем в інформаційній сфері. Для виявлення загроз та управління захищеністю часто використовують апарат аналізу й управління ризиками [3, 6, 8].

Стандарт Великобританії BS 7799 [4] присвячений управлінню інформаційною безпекою організації. Він є одним із найавторитетніших документів у світі. На його базі розроблені міжнародний стандарт ISO/IEC/17799 і його вдосконалений варіант ISO/IEC/27002. Третя частина визначеного стандарту особливо важлива, тому що повністю присвячена питанням управління інформаційними ризиками.

Стандарт BS/7799-3 містить вступну частину, розділи щодо оцінювання та оброблення ризиків, безперервних дій для управління ними, а також додаток із прикладами активів, погроз, уразливостей, методів оцінювання ризиків у стандарті додержуються найбільшого поняття ризику, під яким розуміють комбінацію ймовірності події та її наслідків (вартості компрометованого ресурсу). Управління ризиком (risk management) сформульовано як скоординовані безперервні дії з управління й контролю ризиків в організації.

Зазначений документ допускає застосування як кількісних, так і якісних методів оцінювання ризиків але в ньому немає обґрунтування та рекомендацій щодо вибору математичного й методологічного апарату оцінювання ризиків інформаційної безпеки (ІБ).

Стандарт ISO/27001 [5] не містить конкретних заходів щодо захисту, визначаючи загальну стратегію управління безпекою інформації організації. Він розроблений, щоб надати модель для створення, упровадження, експлуатації, постійного контролю, аналізу, підтримки в робочому стані й покращання системи менеджменту захисту інформації (СМЗІ). Передбачено, що встановлення СМЗІ було стратегічним рішенням для організації.

Згідно з документом необхідно застосовувати процесний підхід до управління захистом інформації [5], тобто створення, упровадження, експлуатації, постійного контролю, аналізу, підтримки в робочому стані та покращання СМЗІ організації. Організація повинна виявити, чи змінилися ризики, й установити вимоги до запобіжних дій. Пріоритет запобіжних дій повинен бути визначеним на основі результатів оцінювання ризику.

Відповідно до стандарту ISO/17799 інформаційна безпека – механізм захисту, завдяки якому досягають таких характеристик, як [6]:

– конфіденційність: доступ до інформації лише авторизованих користувачів;

– цілісність: достовірність і повнота інформації й методів її оброблення;

– доступність: доступ до інформації та пов’язаних із нею активів авторизованих користувачів відповідно до необхідності.

Для інформаційної безпеки потрібна реалізація відповідного комплексу заходів з управління нею, що

можуть бути репрезентованими політиками, методами, процедурами, організаційними структурами й функціями програмного забезпечення.

Вимоги до інформаційної безпеки визначають за допомогою систематичного оцінювання ризиків. Рішення про витрати на заходи з управління інформаційною безпекою необхідно приймати з урахуванням можливого збитку, завданого бізнесу внаслідок порушень інформаційної безпеки.

Необхідність розгляду різних сучасних стандартів [9] зумовлена актуальністю проблеми пошуку та розроблення універсальної методики управління ризиками інформаційної безпеки. У галузі забезпечення інформаційної безпеки ситуація ускладнюється різноманітністю досліджуваних процесів. З огляду на це доцільне введення заходів ризику для автоматизованої системи, що мають узагальнений характер і потребують ретельного дослідження всіх організаційно-технічних параметрів автоматизованої системи.

У зазначених умовах актуальним є завдання швидкого оцінювання ризиків із можливістю налаштування на наявний обсяг відомих даних, що сприятиме розробленню заходів для системи забезпечення інформаційної безпеки (СЗІБ).

Для оперативного оцінювання ризиків на практиці здебільшого застосовують такі методи [1, 2, 7]:

– методи, що базуються на експертних знаннях;

– методи, що базуються на використанні нейронних сіток;

– пошукові методи;

– методи, що базуються на кластеризації.

Найбільш вивченими є методи першої та другої груп.

Такий підхід ефективний, якщо експерт володіє повними знаннями про систему забезпечення інформаційної безпеки, та передбачає:

– збирання інформації про систему з використанням знань фахівців;

– перетворення зібраної інформації на нечітку продукційну модель.

Нечітку продукційну модель (НПМ) можна репрезентувати так [2]:

$$(i)\colon Q;\ P;\ A \Rightarrow B\colon S;\ F;\ N,$$

де Q – сфера застосування використання нечіткої продукції;

P – умова активізації ядра нечіткої продукції;

A – умова ядра (антецедент);

B – висновок ядра (консеквент);

S – метод визначення кількісного значення ступеня істинності висновку ядра;

F – коефіцієнт упевненості нечіткої продукції;

N – післяумова продукційного правила.

Нечітке причинно-наслідкове відношення між антецедентом і консеквентом задають як нечітку продукцію:

ЯКЩО x є A, ТО y є У,

де X – область визначення антецедента;

A – нечітка множина, визначена на X;

$\mu_A(x) \in [0,1]$ – функція належності непарної множини A;

Y – область визначення консеквента;

B – нечітка множина, визначена на Y;

$\mu_A(x) \in [0,1]$ – функція належності нечіткої множини B.

Якщо відома функція належності нечіткої множини A – $\mu_A(x)$, для нечіткої множини В функцію належності визначають за правилом композиції

$$\mu_R(y) = \sup_{x \in X}\{T(\mu_A(x), \mu_R(x, y))\},$$

де sup – операція визначення верхньої межі множини елементів;

Т – операція Т-норми.

Для моделювання ризику інформаційної безпеки організації, зазвичай обчислення нечіткої імплікації, застосовують класичну нечітку імплікацію Л. Заде:

$$\mu_R(y) = \max\{min[\mu_A(x), \mu_B(y)], [1 - \mu_A(x)]\}.$$

Під час побудови нечіткої продукційної моделі оцінювання ризиків ІБ організації необхідно сформувати повний простір передумов $X = \{x_i\}, i = \overline{1, n}$ – факторів, що є джерелами ризику, та повний простір висновків $Y\{y_i\}, i = \overline{1, m}$ – показників ризику різних галузей інформаційної безпеки організації.

Лінгвістичні змінні (ЛЗ) характеризують фактори ризику. Їх описують терммножинами. У процесі аналізу факторів ризику експерти виявляють показники, що можуть бути джерелами ризику ІБ організації (табл. 3.1).

Для опису факторів ризику використовують такі терммножини, що визначають рівні факторів [3]:

Т1 = {Низький (Н), Високий (В)};

Т2 = {Низький (Н), Середній (С), Високий (В)};

Т3 = {Дуже низький (ДН), Низький (Н), Середній (С), Високий (В)};

Т4 = {Дуже низький (ДН), Низький (Н), Середній (С), Високий (В), Дуже високий (ДВ)}.

Таблиця 3.1 – Фактори ризику ІБ організації (фрагмент)

Позначення	**Лінгвістична змінна (ЛЗ) і її термножина**
x1	ЛЗ: Програмно-апаратний рівень захисту
	Н – задовільний, для забезпечення початкового рівня захисту; С – достатній, для базового інформаційного захисту; В – повністю відповідає рівню конфіденційності інформації
x2	ЛЗ: Рівень організаційного захисту
	Н – слабке планування й відсутність моніторингу вразливостей; С – планування та моніторинг уразливостей проводять нерегулярно; В – своєчасне планування й моніторинг уразливостей
x3	ЛЗ: Рівень правового захисту
	Н – уривчаста та неповна документація; З – документація є, але недостатньо детальна; В – документація повна й синхронізована
x4	ЛЗ: Мотивація джерела загроз
	ДужН – немає; Н – рідкісний прояв зацікавленості; С – цілком може зацікавити; В – наймовірніше, зацікавиться; ДужВ – обов’язково зацікавиться
x5	ЛЗ: Можливості джерела загроз
	ДужН – не має можливостей; Н – незначний рівень оснащеності джерела загроз; С – середній рівень оснащеності; В –високий рівень оснащеності; ДужВ – джерело загроз має значні можливості

Для задання лінгвістичних змінних, що характеризують показники ризику, використовують такі терммножини:

Т1 = {Низька очевидність ризику (НОР), Середня очевидність ризику (СОР), Висока очевидність ризику (ВОР)};

Т2 = {Дуже низька очевидність ризику (ДНОР), Низька очевидність ризику (НОР), Середня очевидність ризику (СОР), Висока очевидність ризику (ВОР), Дуже висока очевидність ризику (ДВОР)}.

Показники, що можуть характеризувати ризики ІБ організації, наведені в таблиці 3.2.

Таблиця 3.2 – Показники ризику ІБ організації (фрагмент)

Позначення	**Лінгвістична змінна (ЛЗ) і її опис**
y1	ЛЗ: Ризик зниження ефективності захисту
	Характеризує ймовірність зниження / збільшення ефективності захисту порівняльно з необхідною ефективністю для конкретного підприємства
y2	ЛЗ: Ризик виникнення потенційних загроз
	Характеризує ймовірність виникнення потенційних загроз для підприємства
y3	ЛЗ: Ризик матеріального збитку
	Характеризує ймовірність зазнання матеріального збитку підприємством унаслідок порушення параметрів його інформаційної безпеки
y4	ЛЗ: Ризик ІБ організації
	Інтегральний ризик, що характеризує забезпечення інформаційної безпеки підприємства

Взаємозв'язок між факторами (антецедентом) і показниками ризику (консеквентом) є бінарним нечітким відношенням на декартовому добутку відповідних нечітких множин, які задають як продукційні правила [2].

На рисунку 3.1 зображено нечітку модель із трьома входами й одним виходом, реалізовану у Fuzzy Logic Toolbox – пакеті MATLAB.

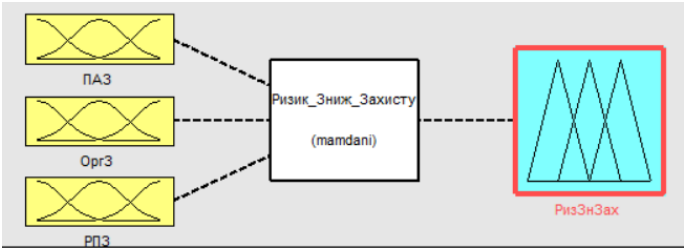


Рисунок 3.1 – Структура нечіткої моделі оцінювання ризиків ІБ

Як вхідні фактори ризику було використано змінні:

– x_1 – програмно-апаратний рівень захисту (ПАЗ);

– x_2 – рівень організаційного захисту (ОргЗ);

– x_3 – рівень правового захисту (РПЗ).

Змінна «ризик зниження ефективності захисту» (РизЗнЗах) відповідає досліджуваному показнику y.

Для математичного опису відповідних лінгвістичних змінних використано функції належності трикутного й трапецієподібного типів.

Обчислювальні експерименти з моделлю проводять способом уведення конкретних числових значень вхідних змінних у систему перегляду логічних правил (рис. 3.2).

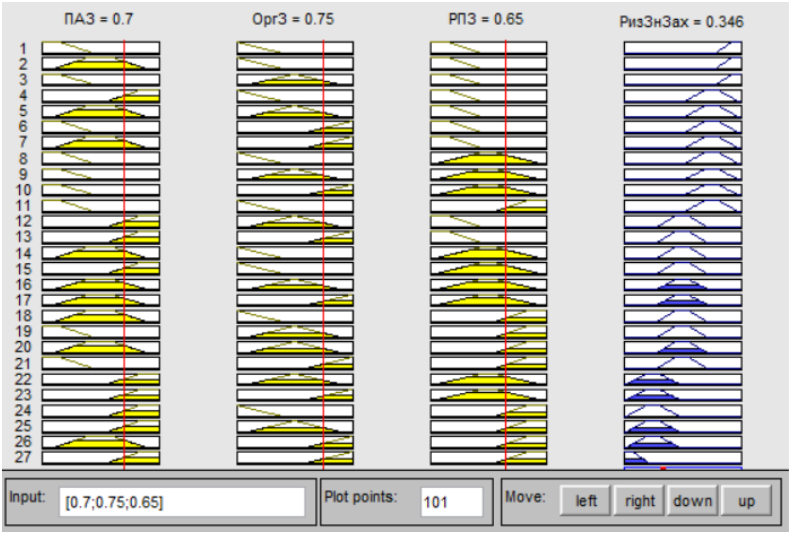


Рисунок 3.2 – Система логічних правил

За значень вхідних класифікаторів (рис. 3.2):

ПАЗ = 0,7 – високий, для базового інформаційного захисту (рівень В);

ОргЗ = 0,75 – планування й моніторинг уразливостей проводяться нерегулярно (рівень С);

РПЗ = 0,65 – документація є, але недостатньо детальна (рівень С).

Підсумкове значення класифікатора вихідної змінної РизЗнЗах 0,346, що свідчить про значення лінгвістичної змінної «ризик зниження ефективності захисту» НОР – низьку очевидність ризику.

На рисунку 3.3 зображено графік «крива виведення» залежності вихідної змінної РизЗнЗах (ризик зниження ефективності захисту) від вхідної змінної ПАЗ (програмно-апаратний рівень захисту) за фіксованих значень двох інших вхідних змінних для бази правил нечіткої моделі.

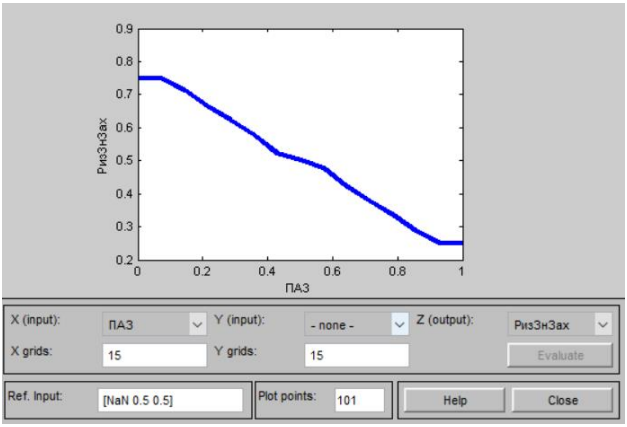


Рисунок 3.3 – Залежність РизЗнЗах від ПАЗ

На графіку можемо бачити зростання ризику зниження ефективності захисту організації в разі зменшення рівнів програмно-апаратного й організаційного захисту.

Отже, метод оцінювання ризиків інформаційної безпеки з використанням нечіткої логіки та інструментарію MATLAB дозволяє наочно репрезентувати стан системи захисту інформації, а також комплексно оцінити можливі загрози безпеки та інформаційні ризики.

Якщо, крім експертних оцінок, одержані результати вимірювань значень входів і виходів системи, розроблена модель повинна бути доповненою механізмом урахування цих зазначених даних [10].

Наведені вище обставини зумовлюють актуальність розроблення нечіткої самоналаштовувальної моделі оцінювання ризиків інформаційної безпеки (СОІБ). Для налаштування моделі, тобто оптимізації її параметрів, найчастіше застосовують метод [10], що базується на використанні нейронечітких мереж. Він пов’язаний із

перетворенням нечіткої моделі на нейронечітку систему (ННС) і застосуванням для налаштування параметрів моделі методів навчання мережі, що базуються на вимірюванні вхідних і вихідних даних системи.

На сьогодні метод вивчений найбільше й має такі переваги [10]:

– забезпечує можливість оптимізації (налаштування) параметрів функцій належності лінгвістичних змінних на основі результатів вимірювання вхідних і вихідних залежностей реальної системи;

– із його застосуванням можна коригувати нечіткі моделі, недостатньо точно сформовані експертами;

– дозволяє поширювати сформовані експертами нечіткі моделі на області досліджуваної системи, про які знання експертів обмежені.

Вищезазначені переваги пояснюють доцільність застосування методів, що базуються на використанні нейронечітких мереж, для налаштування нечіткої моделі оцінювання ризиків інформаційної безпеки організації.

Перетворення елементів блоку фазифікації передбачає перетворення кусково-лінійних функцій належності на фрагмент нейронної мережі.

Для налаштування параметрів функцій належності в процесі навчання мережі необхідно обчислити похідні вихідних значень блоку фазифікації за відповідними параметрами.

Результатом блоку фазифікації є обчислені значення ступеня належності вхідних значень нечітким множинам, кожне з яких представляє свою лінгвістичну область визначення. Перетворення елементів блоку бази правил передбачає подання умови правила як фрагмента нейронної мережі, водночас операції «І» та «АБО» можна виконувати з використанням Т- і S-норми, або за допомогою інших операторів.

Вхідними параметрами блоку дефазифікації є ступені активізації нечітких множин на виході моделі, для перетворення яких на конкретне число застосовують метод центра тяжіння.

Розроблену ННС будують на основі системи нейронечіткого виведення ANFIS (adaptive neuro-fuzzy inference system) [2, 9] за допомогою спеціалізованого пакета Neuro-Fuzzy Designer програмного засобу MATLAB [10].

На етапі фазифікації для терммножин вхідних даних (x1, x2, x3) і вихідної (y) лінгвістичних змінних, описаних у таблицях 3.1 та 3.2 були заданими функції належності трикутного типу:

– x_1 – програмно-апаратний рівень захисту (ПАЗ);
– x_2 – рівень організаційного захисту (ОргЗ);
– x_3 – рівень правового захисту (РПЗ);
– y – ризик зниження ефективності захисту (РизЗнЗах).

У результаті згенерована система нечіткого виведення, що містить 27 правил нечітких продукцій.

Навчання ННС базуються на навчальних вибірках, які є вектором значень рівнів факторів, що впливають на ризик (вхідні ЛЗ), і вектором значень рівня ризику ІБ (вихідних ЛЗ). Дані одержують у результаті узагальнення думок експертів предметної галузі способом застосування методу Дельфі в рамках підходу, запропонованого в праці [10]. Для формування навчальних наборів також можуть бути використаними дані систем виявлення вторгнень, антивірусних програм, міжмережевих екранів та інших систем, що входять до СЗІБ.

Пакет Neuro-Fuzzy Designer дозволяє виконувати навчання методом зворотного поширення похибки, основним призначенням якого є налагодження всіх шарів багатошарової структури способом трансформаційного

змінювання числових значень вагових коефіцієнтів проміжних шарів, і гібридним методом. Результати застосування методів навчання нечіткої мережі оцінювання ризиків інформаційної безпеки наведені в таблиці 3.3.

Згідно з даними зазначеної таблиці гібридний метод навчання дозволяє одержати кращі результати значення помилки мережі за меншу кількість епох. З огляду на це для налаштування параметрів функцій належності був вибраний гібридний метод.

Таблиця 3.3 – Застосування методів навчання нейронечіткої мережі

Метод навчання	**Значення похибки**	**Кількість епох[2]**
Метод зворотного поширення похибки	0,027 1	200
Гібридний метод	0,010 8	28

Графічне репрезентування залежності вихідної ЛЗ «ризик зниження ефективності захисту» від вхідних ЛЗ (програмно-апаратного рівня захисту й рівня організаційного захисту) свідчить про закономірне зростання величини ризику зниження ефективності захисту організації в разі зменшення рівнів програмно-апаратного та організаційної захисту (рис. 3.4).

Механізм оцінювання ризиків на основі ННС дає широкі можливості. Зокрема, його можна адаптувати до наявних моделей управління ризиками, а також модифікувати з урахуванням реальних умов політики інформаційної безпеки організації [1].

Сучасні автоматизовані системи функціонують в умовах мережевих атак і різноманітних конфліктів.

Розроблені системи вимог до управління інформаційною безпекою, що передбачають оцінювання відповідних ризиків від негативних впливів.

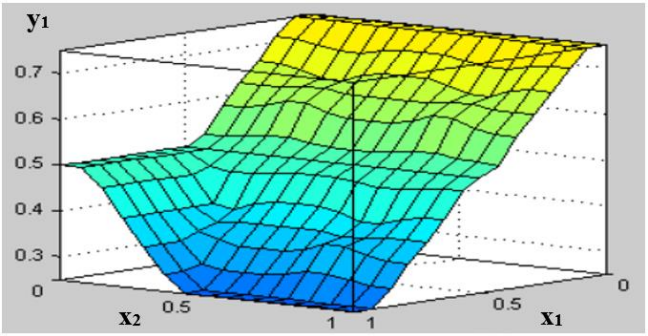


Рисунок 3.4 – Поверхня системи нечіткої моделі

Для оцінювання ризиків, що виникають в автоматизованій системі, додержуються системи стандартів, що регламентують методи оцінювання різноманітних ризиків та технології усунення вразливостей.

На сьогодні актуальними є завдання експрес-оцінювання ризиків на основі обмеженої кількості параметрів, за допомогою якого можна швидко визначати стан справ на підприємстві [7–9].

Зазначені експрес-методи оцінювання ризиків інформаційної безпеки автоматизованих систем орієнтовані на різні варіанти наявних даних про організаційно-технічні параметри автоматизованої системи.

Нечітка продукційна модель дозволяє істотно розширити можливості сучасних методів, зняти обмеження на кількість вхідних змінних та інтегрувати як якісні, так і кількісні підходи до оцінювання ризиків.

Адаптивна нейронечітка система дозволяє безперервно аналізувати ризики інформаційної безпеки, а результати нечіткого моделювання дають можливість ІТ-менеджерам визначати пріоритети ризиків (від «дуже високого» до «дуже низького»), розробляючи ефективні плани заходів щодо зниження впливу найнебезпечніших загроз.

СПИСОК ЛІТЕРАТУРИ

1. Burkov E. A. Analysis of Impact of Marginal Expert Assessments on Integrated Expert Assessment [Electronic resource] / E. A. Burkov // In Proceedings of 2020 23rd International Conference on Soft Computing and Measurements. Institute of Electrical and Electronics Engineers, 2020 – P. 14–17. – Access mode : https://doi.org/10.1109/SCM50615.2020.9198772.

2. Штовба С. Д. Методи оптимізації в середовищі MATLAB: лабораторний практикум : навч. посіб. / С. Д. Штовба. – Вінниця : Вінницький державний технічний університет, 2001. – 54 с.

3. Expert assessment s ystems to support decision-making for sustainable development of complex technological and socioeconomic facilities [Electronic resource] E. Lavrov, P. Paderno, E. Burkov, A. Volosiuk, V. D. Lung // In E3S Web of Conferences. – EDP Sciences 2020. – Vol. 166. – Access mode : https://doi.org/10.1051/e3sconf/202016611002.

4. BS 7799-3:2017 Information security management systems. Guidelines for information security risk management [Electronic resource]. − Access mode : https://shop.bsigroup.com/ProductDetail?pid=000000000030354572.

5. ISO 27001:2013 Risk Assessment and Treatment process [Electronic resource]. − Access mode : https://www.iso.org/obp/ui/#iso:std:iso-iec:27001:ed-2:v1:en.

6. ISO / IEC 17799:2005 Information technology. Security techniques [Electronic resource]. − Access mode : https://www.iso.org/ru/standard/39612.html.

7. Cybersecurity of distributed information systems, the minimization of damage caused by errors of operators during group activity [Electronic resource] / E. Lavrov, A. Tolbatov, N. Pasko, V. Tolbatov // 2nd International Conference on Advanced Information and Communication Technologies. Institute of Electrical and Electronics Engineers, 2017. – P. 83–87. – Access mode : https://doi.org/10.1109/AIACT.2017.8020071.

8. Information Technology for Modeling Human-machine Control Systems and Approach to Integration of Mathematical Models for Its Improvement [Electronic resource] / E. A. Lavrov, P. I. Paderno, E. A. Burkov, O. E. Siryk, N. B. Pasko // In Proceedings of International Conference on Soft Computing and Measurements. Institute of Electrical and Electronics Engineers, 2020. – P. 117–120. – Access mode : https://doi.org/10.1109/SCM50615. 2020.9198791.

9. Risk Management Guide for Information Technology Systems. – Gaithersburg : NIST, 2020. – 55 p.

10. Shtovba S. Fuzzy model tuning based on a training set with fuzzy model output values / S. Shtovba // Cybernetics and Systems Analysis. – 2007. – Vol. 43, № 3. – P. 334–340.

РОЗДІЛ 4
ТЕОРЕТИЧНІ АСПЕКТИ ПОБУДОВИ КОМПЛЕКСНОЇ СИСТЕМИ ЗАХИСТУ ІНФОРМАЦІЇ

М. С. Бабій, В. К. Ободяк

Специфіка предметних сфер комплексної системи захисту інформації (КСЗІ) відображається насамперед у математичних моделях об'єктів системи. Модель у математичній формі описує найбільш істотні властивості об'єктів, передусім це закони функціонування об'єктів, а також взаємозв'язки між окремими елементами об'єкта [1]. Реалізація моделі дозволяє одержати характеристики роботи проєктованої системи.

Залежно від методу дослідження виділяють аналітичні та імітаційні моделі. В аналітичних моделях використовують формальну логіку, теорію ймовірності, математичну статистику. Під час імітаційного моделювання на комп'ютері реалізують зміни стану реальної системи відповідно до реальних процесів.

Виділяють також детерміновані й статистичні моделі.

У загальному випадку математична модель описує залежність між вхідними параметрами і критеріальними показниками об'єкта дослідження. Елементами узагальненої моделі є такі:

− вхідна множина змінних параметрів *X*;

− вхідна множина постійних параметрів *Y*;

− математичний оператор *L*, що визначає операції над вхідними даними;

− множина вихідних даних *G (X, Y)*, що являє собою множину критеріальних функцій.

Ступінь адекватності математичної моделі об’єкта залежить від поставлення і правильності розв’язання задачі проєктування.

Множина X створює метричний простір пошуку розмірності n, де n − кількість параметрів.

Множина Y визначає зовнішні умови, тобто середовище, в якому буде працювати об’єкт. Зовнішні умови можуть містити:

− технічні параметри об’єкта, не змінювані в процесі функціонування об’єкта;

− фізичний вплив середовища.

Вихідні дані моделі створюють метричний простір критеріальних показників.

Схема проєктування моделі наведена на рисунку 4.1.

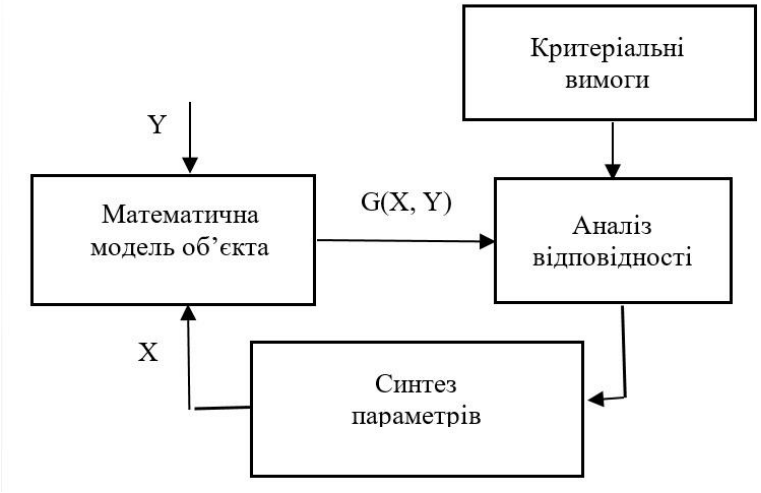


Рисунок 4.1 − Схема проєктування математичної моделі

Моделювання КСЗІ має ряд особливостей [2]:

− складність процесів функціонування КСЗІ;

− велика кількість структур КСЗІ і шляхів об’єднання елементів в єдину архітектуру;

− різноманітність складових елементів, підсистем і зв’язків між ними;

− складність функцій, наявних у системі;

− робота системи в умовах невизначеності;

− наявність критеріїв ефективності КСЗІ;

− наявність ієрархічної системи керування;

− розгалуженість і великий обсяг інформаційних потоків.

Вхідну інформацію для моделювання можна поділити на групи:

− нормативи, тактико-технічні характеристики;

− оперативна інформація про джерела загроз, нові методи захисту інформації;

− нормативна інформація, змінювана в процесі моделювання.

Основні цілі математичного моделювання:

− аналізування та оцінювання можливих значень параметрів функціонування системи;

− синтез, тобто проєктування системи, оптимальної за деяким показником або за їх сукупністю;

− пошук оптимальних керувальних впливів на параметри системи.

Розглянемо побудову моделі загального оцінювання загроз безпеці.

Основними завданнями комплексної системи захисту інформації є нейтралізація загроз і локалізація наслідків у випадку їх реалізації. Для ефективної протидії загрозам необхідно мати інформацію про об’єкти, на які можуть бути спрямовані загрози, обстановку, що може сприяти виникненню загроз, можливих порушників. Модель загроз повинна враховувати всі ці параметри.

Можливий варіант взаємодії окремих моделей наведено на рисунку 4.2.

Для опису моделей використано позначення:

$M_{ЗЗ}$ − модель загроз загальна;

O − множина об'єктів захисту;

C_B − привабливість загрози для порушника;

Q − збитки під час реалізації загрози;

$M_З$ − інтегральна модель загроз, що вміщує підмоделі комплексного рівня $M_{КР}$, сценарного рівня $M_{СР}$, рівня дій порушника $M_{РДП}$;

$M_П$ − модель порушника;

T − множина цілей;

$F_{КР}$, $F_{СР}$, $F_{РДП}$ − функції відповідних рівнів;

A − множина дій порушників;

$M_{КСЗІ}$ − формальна модель КСЗІ.

Загальну модель загроз використовують для відбору найбільш небезпечних загроз і відсіювання другорядних.

Відбір здійснюють на основі пріоритетів, які призначаються на основі можливих збитків від реалізації загрози.

Відсіяні загрози подають на вхід інтегрованої моделі загроз, де створюють модель порушника, описують можливі дії порушника і формують сценарії реалізації загроз. Інтегрована модель містить три рівні: комплексний, сценарний і рівень дій порушника.

Комплексний рівень параметризує сценарії реалізації загроз з урахуванням моделі $M_П$ і загалом вміщує множини цілей T, об'єктів захисту O і функцій цього рівня $F_{КР}$.

На сценарному рівні формують множину сценаріїв, які становлять послідовність дій порушника щодо вибраної мети.

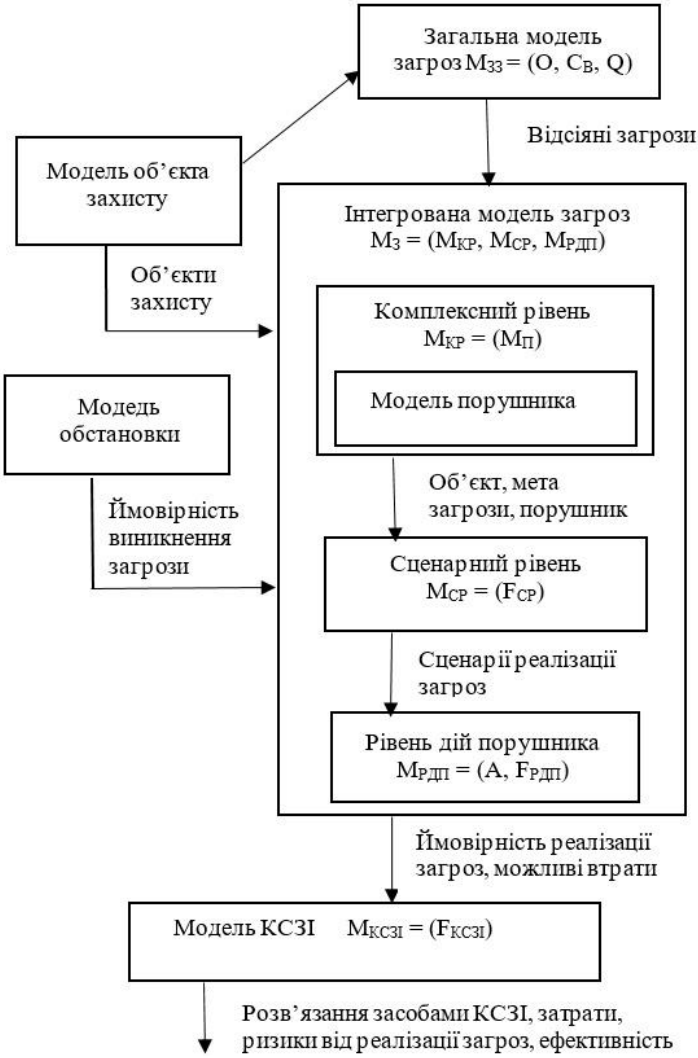


Рисунок 4.2 − Взаємозв'язок моделей об'єкта захисту, обстановки, загроз, порушника, КСЗІ

Сценарії формують шляхом перебирання всіх дій порушника: $T \times A \rightarrow S$, де A – множина дій порушника; S – множина сценаріїв.

Рівень дій порушника в загальному вигляді вміщує множину дій порушника і множину функцій цього рівня $F_{РДП}$.

Кожна дія порушника подається кортежем, який містить мету, час для успішної реалізації дій порушника, збитки під час реалізації, ймовірність виконання порушником даної дії, рекомендації щодо реагування на дію порушника засобами КСЗІ.

Кількість потенціально можливих сценаріїв може бути досить великою, тому необхідно проводити їх відбір. Найбільш ефективним є відбір за величиною ризику. Ризик R від реалізації сценарію обчислюють за формулою $R = Y\,P\,(B)$, де Y − сумарні збитки від дій даного сценарію, а $P\,(B)$ − ймовірність реалізації загрози B. Для ухвалення рішення про відбір розрахований ризик R порівнюють із прийнятним ризиком $R_{ПР}$.

На основі вищеописаних моделей та аналізу відповідних функцій, а також з урахуванням завдань, які стоять перед КСЗІ, може бути запропонована загальна функціональна модель КСЗІ (рис. 4.3).

Модель відображає процес забезпечення роботи системи в безпечному режимі та являє собою послідовність взаємозв’язаних функцій. За допомогою засобів моніторингу проводять сканування потоків даних для виявлення потенціальних загроз. Для класифікації загроз сховище даних містить інформацію про вже відомі загрози, потенційних порушників та об’єкти, вразливі до загроз. Після аналізування та оцінювання загроз вживають заходів із їх нейтралізації. Наведена схема працює в неперервному режимі.

Найбільшу увагу під час побудови комплексної системи захисту інформації потрібно приділити нормативним документам [3−8].

Розглянемо далі побудову моделі оцінювання ризиків за умови реалізації загроз безпеці.

Під ризиком реалізації загроз для безпеки об’єкта захисту розуміють імовірність виникнення ситуації, що призводить до порушення режиму функціонування об’єкта та економічних збитків.

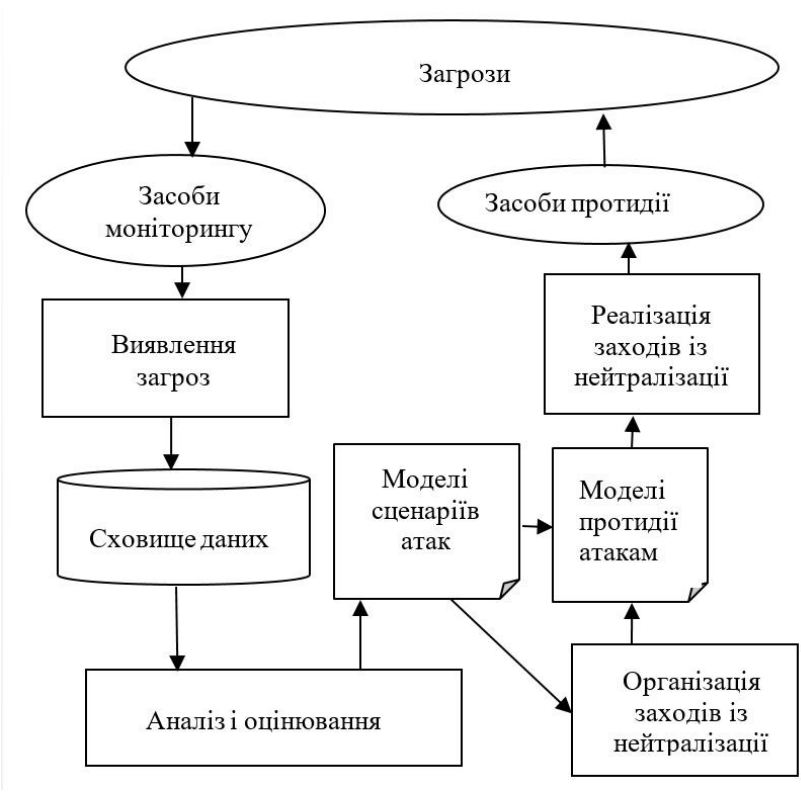


Рисунок 4.3 − Загальна функціональна модель КСЗІ

Значення ризику залежить від потенціальної можливості виникнення загрози та розміру наслідків її реалізації.

Виникненню ризику сприяють такі три чинники: людський, технічний і чинник середовища.

Людський чинник $R_Л$ погіршує стан захищеності системи через неадекватну поведінку персоналу. Вона може бути наслідком:

– великого обсягу робіт, покладених на співробітника;

– навмисних дій співробітника, спричинених економічними або соціально-політичними чинниками;

– некомпетентності персоналу.

Технічний чинник R_T можна визначити за формулою

$$R_T = k\sum_{i=1}^{N}(p_i v_i),$$

де k – важливість організаційної структури; N – кількість систем в структурі; p_i – ймовірність несанкціонованого доступу до системи; v_i – важливість системи в організаційній структурі. Величину p_i можна визначити через міцність бар'єрів захисту, ймовірність обходу цих бар'єрів та ймовірність $P_{відм}$ відмови пристроїв.

Величину $P_{відм}$ визначають за формулою

$$P_{відм}(t) = e^{-\lambda t},$$

де t – інтервал часу; λ – інтенсивність відмов обладнання даного типу.

Чинник інфраструктури зловмисника до можливостей керувального пристрою.

Оцінимо ефективність розроблення КСЗІ. У загальному випадку ефективність заходів із забезпечення безпеки визначають за формулою

$$E = \sum_{k=1}^{K}(Y_1^k - Y_2^k) - \sum_{m=1}^{M}(\Delta K_з - \Delta E_з)_m ,$$

де Y_1 – можливий збиток до впровадження КСЗІ;

Y_2 – можливий збиток після впровадження КСЗІ;

$\Delta K_з$ – капітальні затрати на засіб безпеки;

$\Delta E_з$ – експлуатаційні затрати на засіб безпеки;

k – вид збитку;

m – вид засобу безпеки.

Ефективність розроблення КСЗІ досягають завдяки мінімізації затрат шляхом зміни $\Delta K_з$ і $\Delta E_з$ на впровадження кожного засобу безпеки.

Ризик від реалізації загрози обчислюють за формулою $R = Y P(B)$, де Y – можливий збиток від загрози; $P(B)$ – ймовірність реалізації загрози.

Підвищення ефективності розроблення КСЗІ можна досягти трьома способами

1) $\sum_{k=1}^{K}(Y_1^k - Y_2^k) \to \max, \quad \sum_{m=1}^{M}(\Delta K_з - \Delta E_з)_m = const;$

2) $\sum_{k=1}^{K}(Y_1^k - Y_2^k) = const, \quad \sum_{m=1}^{M}(\Delta K_з - \Delta E_з)_m \to \min;$

3) $\sum_{k=1}^{K}(Y_1^k - Y_2^k) \to \max, \quad \sum_{m=1}^{M}(\Delta K_з - \Delta E_з)_m \to \min .$

Ймовірні втрати Y від реалізації загрози можна оцінити за допомогою формули

$$Y = Y^1 + Y^2 + Y^3 + Y^4 + Y^5 + Y^6 + Y^7,$$

де Y^1 – прямий збиток;

Y^2 – затрати на ліквідацію;

Y^3 – економічні затрати;

Y^4 – непрямий збиток;

Y^5 – екологічний збиток;

Y^6 – втрата працівників;

Y^7 – втрата інформаційних ресурсів.

Програму керування необхідно формувати так, щоб затрати на безпеку відповідали потенціальним загрозам. Від того, наскільки повно спрогнозовані загрози для об'єкта, залежить і його захищеність. З іншого боку реалізація деяких загроз може бути малоймовірною, або ж захист від неї може бути неможливим чи малоефективним. У цьому разі кошти будуть витрачені марно, що призведе до зниження показників ефективності розроблення КСЗІ. Отже, завдання оцінювання можливості реалізації загрози є надзвичайно важливими.

Звідси випливає необхідність вирішення завдання оптимізації ступеня захищеності об'єкта, а це б дозволяло домогтись максимальної ефективності від упровадження комплексної системи захисту. На перший погляд здається, що збільшення витрат на захист приводить до підвищення ефективності системи захисту. Але тут необхідно враховувати обмеження. Існує межа, за якою спроєктована система стає нерентабельною, тобто норма фінансового виграшу не буде виправдовувати кошти, інвестовані в систему захисту.

Для підвищення якості комплексної системи захисту інформації може бути задіяний апарат нечітких множин [9].

Рентабельність системи захисту може бути підвищена завдяки більш раціональному підходу до витрат на впровадження та експлуатацію, а також оптимальному розподілу коштів за простором загроз.

Іншим напрямком підвищення рентабельності є зонування системи захисту. Важливе значення в цьому разі має взаємне розміщення зон захисту, яке в цілому визначає структуру системи захисту.

На підставі аналізу існуючих методів зонування можна виділити три основні варіанти розміщення зон:

− структури з незалежними зонами;

− структури з частковим перекриттям зон;

− структури вкладених зон захисту.

Найбільш часто використовується перший варіант, який є відносно простим і дає можливість автономно реагувати на внутрішні й зовнішні загрози. Деяким недоліком в цьому разі є неможливість використання ресурсів із інших зон.

Якість протидії загрозам для цього варіанта оцінюється показниками ефективності захисту. Якщо загрози є статистично незалежними, то загальна ефективність E є лінійною комбінацією ефективностей E_j захисту окремих зон:

$$E = \sum_{j=1}^{M} \delta_j E_j .$$

Величина δ_j представляє відносну значущість зони з номером j і виражається формулою

$$\delta_j = \frac{\sum_{i=1}^{N_j} P_{ij}\delta_{ij}}{\sum_{j=1}^{M}\sum_{i=1}^{N_j} P_{ij}\delta_{ij}},$$

де M – кількість зон захисту в структурі КСЗІ;

N_j – загальна кількість загроз, які можуть реалізуватися в зоні з номером j;

P_{ij} – ймовірність реалізації загрози з номером i в зоні з номером j;

δ_{ij} – значущість зони захисту з номером j у відношенні загрози з номером i.

Використовуючи зонну структуру захисту, важливо слідувати таким рекомендаціям:

− для зон з вищою значимістю захист повинен бути більш ефективним, відповідно він буде потребувати більше витрат на його організацію;

− для досягнення високого загального рівня ефективності більша частина витрат повинна бути спрямована на захист зон із високим рівнем значущості.

Оптимізація розподілу ресурсів за зонами дозволяє економічно обґрунтувати виділення технічних і програмних засобів для кожної із зон. Як критерій оптимальності розподілу можна взяти суму витрат на систему захисту і збитків від реалізації загроз.

Позначимо через P_{ijk} ймовірність реалізації загрози з номером j у відношенні об'єкта з номером k за умови, що не використовується метод захисту з номером i. Збиток організації від реалізації загрози позначимо через L_{ijk}. Тоді математичне сподівання збитку

$$M_{ijk} = P_{ijk} L_{ijk}.$$

Формула набирає такого вигляду:

$M_{ij} = \sum_{k=1}^{K} P_{ijk} L_{ijk}$ − для всіх об'єктів захисту;

$M_{ik} = \sum_{j=1}^{M} P_{ijk} L_{ijk}$ − для всіх загроз;

$M_{jk} = \sum_{i=1}^{N_j} P_{ijk} L_{ijk}$ − у разі невикористання всіх методів захисту.

Повне математичне сподівання збитку від усіх загроз для всіх об'єктів у разі невикористання методів захисту буде відповідати максимально допустимим витратам Z_{max}:

$$Z_{\max} = M = \sum_{i,j,k} P_{ijk} L_{ijk}.$$

Функціонування системи захисту планується на деякий період, наприклад, на час існування об'єкта захисту, або на час існування загрози.

У випадку короткого періоду для зменшення витрат можуть бути невраховані загрози, що можуть виникати досить рідко. Але якщо така загроза виникне, то збитки можуть бути значними і система стає нерентабельною.

У випадку вибору великого періоду ймовірність виникнення загроз різного типу може змінюватися. Крім того, можуть з'явитися нові загрози, що в результаті зменшує ефективність захисту.

Таким чином, період функціонування також потребує оптимізації.

На підставі проведених досліджень можна зробити такі висновки. Сучасні КСЗІ мають складну структуру з великою кількістю елементів, зв'язків та інформаційних потоків між ними. Досягнення високої ефективності роботи КСЗІ неможливе без застосування теоретико-аналітичних методів дослідження. У цій роботі запропоновані та проаналізовані базові моделі об'єкта захисту, обстановки, загроз, порушника та КСЗІ; розроблена функціональна модель комплексної системи захисту інформації; наведені основні математичні залежності для розрахунку супутніх факторів і критеріїв ефективності.

Ураховуючи сучасний тренд на розвиток автоматизованих систем проєктування, в подальших роботах планується розширити математичний апарат, що використовується для проєктування КСЗІ.

СПИСОК ЛІТЕРАТУРИ

1. Saini D. K. Cyber Defense: Mathematical Modeling and Simulation / D. K. Saini // International Journal of Applied

Physics and Mathematics. – 2012, September. – Vol. 2, No. 5. − P. 312−315.

2. Проектування, введення в дію та супроводження КСЗІ : навчальний посібник / В. Д. Козюра, В. О. Хорошко, М. Є. Шелест та ін. – Ніжин : ФОП Лук'яненко В. В. ; ТПК «Орхідея», 2019. – 240 с.

3. НД ТЗІ 1.1-005-07. Захист інформації на об'єктах інформаційної діяльності. Створення комплексу технічного захисту інформації. Основні положення. Затверджено указом Адміністрації Державної служби спеціального зв'язку та захисту інформації України від 12.12.2007 № 232 [Електронний ресурс]. − Режим доступу : https://tzi.com.ua/downloads/3.1-001-07.pdf.

4. НД ТЗІ 2.5-004-99. Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу. Затверджено указом ДСТСЗІ СБ України від 28.04.1999 № 22 [Електронний ресурс]. − Режим доступу : https://tzi.ua/assets/files/%D0%9D%D0%94-%D0%A2%D0%97%D0%86-2.5-004-99.pdf.

5. НД ТЗІ 2.5-005-99. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу. Затверджено указом ДСТСЗІ СБ України від 28.04.1999 № 22 [Електронний ресурс]. − Режим доступу : https://tzi.ua/assets/files/%D0%9D%D0%94-%D0%A2%D0%97%D0%86-2.5-005--99.pdf.

6. НД ТЗІ 3.1-001-07. Захист інформації на об'єктах інформаційної діяльності. Створення комплексу технічного захисту інформації. Передпроєктні роботи. Затверджено указом Адміністрації Державної служби спеціального зв'язку та захисту інформації України від 12.12.2007 № 232 [Електронний ресурс]. − Режим доступу : https://tzi.com.ua/downloads/3.1-001-07.pdf.

7. НД ТЗІ 3.3-001-07. Захист інформації на об’єктах інформаційної діяльності. Створення комплексу технічного захисту інформації. Порядок розроблення та впровадження заходів із захисту інформації. Затверджено указом Адміністрації Державної служби спеціального зв’язку та захисту інформації України від 12.12.2007 № 232 [Електронний ресурс]. − Режим доступу : https://tzi.com.ua/downloads/3.3-001-07.pdf.

8. НД ТЗІ 3.7-003-2005. Порядок проведення робіт із створення комплексної системи захисту інформації в інформаційно-телекомунікаційній системі. Затверджено наказом ДСТСЗІ СБ України від 08.11.2005 № 125 [Електронний ресурс]. − Режим доступу : https://tzi.com.ua/downloads/3.7-003-2005.pdf.

9. Нечітка ієрархічна оцінка якості комплексних систем захисту інформації / І. В. Шелехов, Н. Л. Барченко, В. В. Кальченко, В. К. Ободяк // Радіоелектронні і комп’ютерні системи. − 2020. − № 4 (96). − С. 106−115.

РОЗДІЛ 5
КІБЕРБЕЗПЕКА ТА ТЕХНОЛОГІЯ WI-FI: ФІЗИЧНІ ТА ТЕХНІЧНІ ОСОБЛИВОСТІ СУЧАСНИХ СТАНДАРТІВ

В. В. Коваль, В. К. Ободяк, Б. О. Кузіков

У жовтні 2018 року групою Wi-Fi Alliance була репрезентована нова класифікація (назви) чинних і нових стандартів Wi-Fi [1].

Стандарти 802.11 перейменовані на більш прості і зрозумілі імена (табл. 5.1) [1].

Таблиця 5.1 – Модифікація імен стандарту 802.11

Стара назва стандарту	**Нова назва стандарту**
802.11b	Wi-Fi 1
802.11a	Wi-Fi 2
802.11g	Wi-Fi 3
802.11n	Wi-Fi 4
802.11ac	Wi-Fi 5
802.11ax	Wi-Fi 6

Така спрощена нумерація, на думку Wi-Fi Alliance, дозволить користувачам, які мають невеликий досвід роботи з обладнанням Wi-Fi, здійснити правильний підбір обладнання для забезпечення максимальної продуктивності в мережі Wi-Fi, наприклад, щоб сучасні гаджети з адаптером 802.11ac не використовувати в мережі стандарту 802.11n.

Порівняти номери стандартів тепер буде набагато простіше, а послідовна нумерація легко дозволить визначити, який із стандартів більш новий. У зв’язку з цим

на мережному обладнанні повинне з'явитися маркування стандарту Wi-Fi нового виду (рис. 5.1).

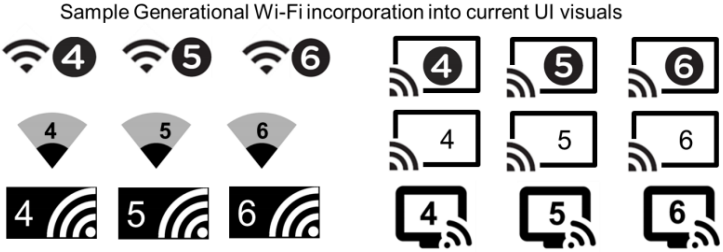


Рисунок 5.1 – Маркування стандарів Wi-Fi нового виду [1]

Сьогодні активно відбувається перехід побутових бездротових пристроїв (маршрутизатори, точки доступу, бездротові адаптери тощо) на стандарт бездротових мереж п'ятого покоління Wi-Fi 5 (IEEE 802.11ac). Розглянемо основні переваги та недоліки цього стандарту. Водночас основну увагу приділимо ключовим фізичним і технічним особливостям та проведемо порівняння з попереднім стандартом Wi-Fi 4 (IEEE 802.11n). Необхідно відзначити, що на сьогодні стандарти та технології Wi-Fi стрімко розвиваються і активно йде мова про стандарти 802.11ax та 802.11be [2]. Фінальна версія специфікації IEEE 802.11ac була ухвалена в січні 2014 року.

Основними перевагами стандарту 802.11ac є такі:

– робота бездротових пристроїв відбувається в діапазоні частот 5 ГГц;

– збільшення швидкості та продуктивності бездротової мережі передання даних;

– збільшення ширини каналів;

– збільшення кількості просторових потоків;

– використання нової технології модуляції сигналу;

– використання технології багатокористувацької MIMO (Multi-User MIMO);

підтримка технології формування спрямованого сигналу Beamforming.

Розглянемо більш детально переваги стандарту 802.11ac.

Робота бездротових пристроїв відбувається в діапазоні частот 5 ГГц

Стандарт бездротових мереж 802.11ac використовує тільки діапазон частот 5 ГГц (стандарт 802.11n може працювати як в діапазоні частот 2,4 ГГц, так і 5 ГГц). На сьогодні основна маса бездротових пристроїв працює в діапазоні 2,4 ГГц, тому сигнал у діапазоні 5 ГГц зазнає менше різних перешкод. Використання вільнішого радіоефіру, призводить до підвищення стабільності і швидкості з'єднання.

Збільшення швидкості та продуктивності бездротової мережі передання даних

Стандарт 802.11ac заявляє про максимальну теоретичної швидкості підключення до 7 Гбіт/с. Теоретично це має бути точка доступу або маршрутизатор із вісьмома антенами. У вільному продажу зараз трапляються пристрої зі швидкістю передання даних до 1,3 Гбіт/с. Домогтися істотного збільшення швидкості передання даних вдалося завдяки збільшенню ширини каналу до 80 МГц та збільшенню числа просторових потоків і підтримки нової модуляції 256-QAM.

Збільшення ширини каналів

Відповідно до стандарту 802.11ac ширина бездротового каналу для передання сигналу була збільшена до 80 МГц. Необхідно зазначити, що опціонально розширення ширини каналу можливо до 160 МГц. Дворазове збільшення ширини каналу (порівняно зі стандартом 802.11n, який використовує ширину каналу до 40 МГц) призводить до підвищення швидкості передання даних і поліпшення пропускних можливостей.

Збільшення кількості просторових потоків

Попередній стандарт 802.11n передбачає можливість використання до 4 просторових потоків, а в стандарті 802.11ac їхню кількість було збільшено до 8. Коли одночасно з різних антен відбувається передання радіосигналу, для уникнення колізії передання даних повинні застосовуватися роздільні просторові потоки (Spatial Streams).

Технологія MIMO (Multiple-Input Multiple-Output) забезпечує одночасний прийом і передання кількох потоків даних через кілька антен. Чим більше просторових потоків, тим більше потрібно антен для їхнього передання та прийому. Чим більше пристрій використовує антен для одночасної роботи передання та прийому, тим буде вище його максимальна швидкість передання даних.

Використання нової технології модуляції сигналу

Застосування в стандарті 802.11ac нової і більш продуктивної системи модуляції сигналу 256-QAM забезпечує приріст пропускної здатності в бездротовій мережі. Модуляція 256-QAM порівняно з 64-QAM приблизно на 25 % збільшує швидкість передання даних. Наприклад, на стандарті 802.11ac за ширини каналу 40 МГц, під час використання одного просторового потоку і модуляції 256-QAM максимальна швидкість у каналі становить 200 Мбіт/с, а на стандарті 802.11n за тих самих параметрів, але на модуляції 64-QAM становить 150 Мбіт/с. Знаючи ширину каналу, кількість просторових потоків і тип модуляції, що використовує пристрій, можна розрахувати максимально можливу теоретичну швидкість передання даних у кожному конкретному випадку.

У таблиці 5.2 наведені максимальні теоретично можливі швидкості передання даних (Data Rate) стандарту 802.11ac,залежно від різних параметрів:

- тип модуляції (Modulation);
- швидкість кодування (Coding Ratio);
- число просторових потоків (Spatial Stream);
- ширина каналу (20/40/80 / 160-MHz).

На одному просторовому потоці за ширини каналу 160 МГц і модуляції 256-QAM теоретично можливо отримати швидкість у 867 Мбіт/с. Тоді з урахуванням, що стандарт 802.11ac підтримує 8 просторових потоків по 867 Мбіт/с теоретично можливо отримати значення швидкості передання даних приблизно 7 Гбіт/с, що значно перевищує максимальну теоретично можливу швидкість на стандарті 802.11n (вона становить 600 Мбіт/с у разі використання 4 просторових потоків, кожен із яких працює на швидкості 150 Мбіт/с).

Підтримка технології MU-MIMO

Технологія MIMO, реалізована в стандарті 802.11n, забезпечує одночасну роботу передання/прийому даних між пристроями мережі. Але в конкретний момент часу тільки один пристрій може отримувати і відправляти дані, тоді як інші чекають своєї черги. Стандарт 802.11ac змінює підхід. У межах стандарту була реалізована технологія багатокористувацької MIMO – MU-MIMO (Multi-User Multiple-Input, Multiple-Output).

Технологія MU-MIMO (рис. 5.2) створює канал передання, у разі використання якого інші пристрої не чекають своєї черги. Пристрої з підтримкою MU-MIMO можуть забезпечувати одночасне передання чотирьох потоків даних (до чотирьох клієнтів). Це дозволило реалізувати більш ефективне використання бездротової мережі і скоротити час затримки (очікування на обслуговування), який виникає за значного збільшення кількості клієнтів у мережі.

Таблиця 5.2 – Швидкість передання даних

MCS index	Spatial Streams	Modulation type	Coding rate	Data rate (in Mbit/s)							
				20 MHz channels		40 MHz channels		80 MHz channels		160 MHz channels	
				800 ns	400 ns GI	800 ns GI	400 ns GI	800 ns GI	400 ns GI	800 ns GI	400 ns GI
0	1	BPSK	1/2	6.5	7.2	13.5	15	29.3	32.5	58.5	65
1	1	QPSK	1/2	13	14.4	27	30	58.5	65	117	130
2	1	QPSK	3/4	19.5	21.7	40.5	45	87.8	97.5	175.5	195
3	1	16-QAM	1/2	26	28.9	54	60	117	130	234	260
4	1	16-QAM	3/4	39	43.3	81	90	175.5	195	351	390
5	1	64-QAM	2/3	52	57.8	108	120	234	260	468	520
6	1	64-QAM	3/4	58.5	65	121.5	135	263.3	292.5	526.5	585
7	1	64-QAM	5/6	65	72.2	135	150	292.5	325	585	650
8	1	256-QAM	3/4	78	86.7	162	180	351	390	702	780
9	1	256-QAM	5/6	N/A	N/A	180	200	390	433.3	780	866.7
0	2	BPSK	1/2	13	14.4	27	30	58.5	65	117	130
1	2	QPSK	1/2	26	28.9	54	60	117	130	234	260
2	2	QPSK	3/4	39	43.3	81	90	175.5	195	351	390
3	2	16-QAM	1/2	52	57.8	108	120	234	260	468	520
4	2	16-QAM	3/4	78	86.7	162	180	351	390	702	780
5	2	64-QAM	2/3	104	115.6	216	240	468	520	936	1040
6	2	64-QAM	3/4	117	130.3	243	270	526.5	585	1053	1170
7	2	64-QAM	5/6	130	144.4	270	300	585	650	1170	1300
8	2	256-QAM	3/4	156	173.3	324	360	702	780	1404	1560
9	2	256-QAM	5/6	N/A	N/A	360	400	780	866.7	1560	1733.4
0	3	BPSK	1/2	19.5	21.7	40.5	45	87.8	97.5	175.5	195
1	3	QPSK	1/2	39	43.3	81	90	175.5	195	351	390
2	3	QPSK	3/4	58.5	65	121.5	135	263.3	292.5	526.5	585
3	3	16-QAM	1/2	78	86.7	162	180	351	390	702	780
4	3	16-QAM	3/4	117	130	243	270	526.5	585	1053	1170
5	3	64-QAM	2/3	156	173.3	324	360	702	780	1404	1560
6	3	64-QAM	3/4	175.5	195	364.5	405	N/A	N/A	1579.5	1755
7	3	64-QAM	5/6	195	216.7	405	450	877.5	975	1755	1950
8	3	256-QAM	3/4	234	260	486	540	1053	1170	2106	2340
9	3	256-QAM	5/6	260	288.9	540	600	1170	1300	2340	2600
0	4	BPSK	1/2	26	28.8	54	60	117.2	130	234	260
1	4	QPSK	1/2	52	57.6	108	120	234	260	468	520
2	4	QPSK	3/4	78	86.8	162	180	351.2	390	702	780
3	4	16-QAM	1/2	104	115.6	216	240	468	520	936	1040
4	4	16-QAM	3/4	156	173.2	324	360	702	780	1404	1560
5	4	64-QAM	2/3	208	231.2	432	480	936	1040	1872	2080
6	4	64-QAM	3/4	234	260	486	540	1053.2	1170	2106	2340
7	4	64-QAM	5/6	260	288.8	540	600	1170	1300	2340	2600
8	4	256-QAM	3/4	312	346.8	648	720	1404	1560	2808	3120
9	4	256-QAM	5/6	N/A	N/A	720	800	1560	1733.2	3120	3466.8

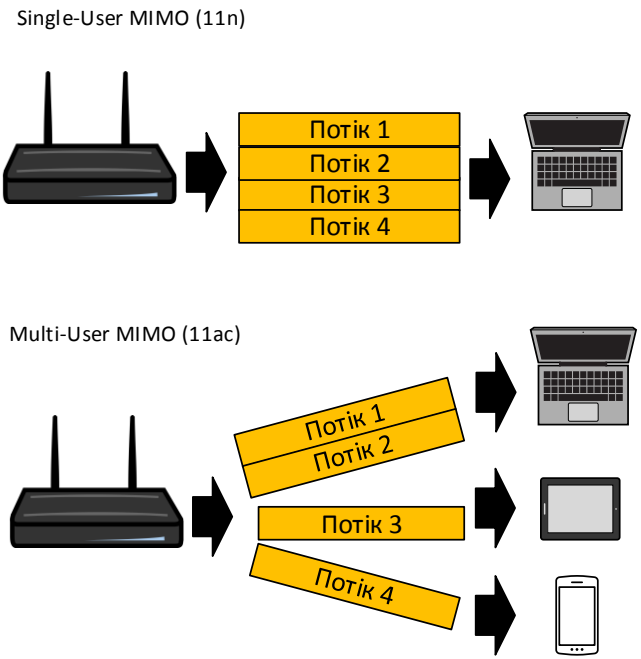


Рисунок 5.2 – Схематичне порівняння технологій Single-MIMO та Multi-MIMO

Підтримка технології формування спрямованого сигналу Beamforming

У стандарті 802.11ac реалізована підтримка технології формування спрямованого сигналу Beamforming Ця технологія також відома як технологія адаптивного формування діаграми спрямованості – Transmit Beamforming або Tx Beamforming.

Ця технологія розв'язує проблему падіння потужності сигналу, викликану його відбиттям від різних предметів і поверхонь. Технологію формування спрямованого сигналу можна було вже застосовувати в межах стандарту 802.11n, однак на той момент вона не

була стандартизована, і в разі використання пристроїв різних виробників вона зазвичай працювала некоректно.

Фізичний алгоритм роботи технології Beamforming такий:

– радіосигнали, які були прийняті від клієнтів, допомагають точці доступу визначити місце розташування клієнтів у просторі

– інформацію, отриману від клієнта, використовують для розрахунку і формування вузькоспрямованого сигналу.

У звичайному режимі роботи сигнал від приймача розповсюджується рівномірно у всі сторони, а за Beamforming спрямовується в чітко визначеному напрямку, що досягають за допомогою декількох антен.

Застосування технології Beamforming дозволяє більш ефективно використовувати смуги пропускання, що досить ефективно проявляє себе під час роботи з потоковою музикою, відео, іграми або додатками, які дуже чутливі до пропускної потужності та затримок у мережі. Також була реалізована сумісність пристроїв із підтримкою цієї технології. Тепер, якщо один пристрій має підтримку Beamforming, а в іншого вона відсутня, пристрої зможуть працювати, тоді як раніше це було неможливо.

Розвиток технології Beamforming є одним із перспективних напрямків підвищення швидкості передання інформації. Застосування різноманітних алгоритмів і модифікацій технології дозволяє покращити характеристики передання даних [3].

Переваги стандарту 802.11ac дозволяють створити надійну мережу з досить великою швидкістю передання даних. Це, зі свого боку, є привабливим для створення захищених систем на їхній основі. Але під час побудови мережі необхідно приділити особливу увагу врахуванню

фізичних особливостей розповсюдження електромагнітних хвиль у діапазонах 2,4 ГГц та 5 ГГц. Нехтування цим питання може призвести до значних проблем у плані безпеки та пошуку «примарних» порушників безпеки. Урахування цих особливостей може підвищити показник безпеки захищеності мережі у зв'язку з неможливістю фізичного під'єднання до неї. Це також дає можливість практично миттєво реагувати на появу сторонніх джерел сигналів у системі.

Усі наведені в таблиці 5.2 швидкості є теоретично максимально досяжними. Максимальні реальні швидкості для стандарту будуть нижче ніж, що зазначено на пристрої.

Ефективність обладнання в кожному випадку буде залежати від наявності інших бездротових пристроїв, конфігурації приміщення і інших чинників, що впливають на роботу мереж Wi-Fi. Орієнтовно, маршрутизатор із заявленою швидкістю бездротової мережі до 876 Мбіт/с зможе передавати інформацію не швидше ніж 400 Мбіт/с.

Основною проблемою якості бездротової мережі є те, що існує велика кількість спеціального та побутового обладнання, яке працює в діапазонах 2,4 ГГц та 5 ГГц. Це призводить до інтерференції та колізій, що погіршує якісні характеристики мережі.

У діапазоні частот 2,4 ГГц для бездротових мереж доступні 11 або 13 каналів шириною 20 МГц (802.11b/g/n) або 40 МГц (IEE 802.11n) з інтервалами 5 МГц між ними. Бездротовий пристрій, що використовує один із частотних каналів, створює значні перешкоди на сусідні канали. Наприклад, якщо точка доступу використовує канал 6, то вона надає сильні перешкоди на канали 5 і 7, а також більш слабші на канали 4 і 8. Для усунення взаємних перешкод між каналами необхідно, щоб їхні основні частоти відрізнялись одна від одної на 25 МГц.

На рисунку 5.3 схематично подані спектри 11 каналів. Однакові кольори позначають групи каналів, які не перекриваються: [1, 6, 11], [2, 7], [3, 8], [4, 9], [5, 10]. Рекомендуємо для бездротових мереж, які розташовані в межах однієї зони покриття, налаштовувати канали так, щоб вони не перекривались. У цьому разі не буде спостерігатись інтерференція та колізії.

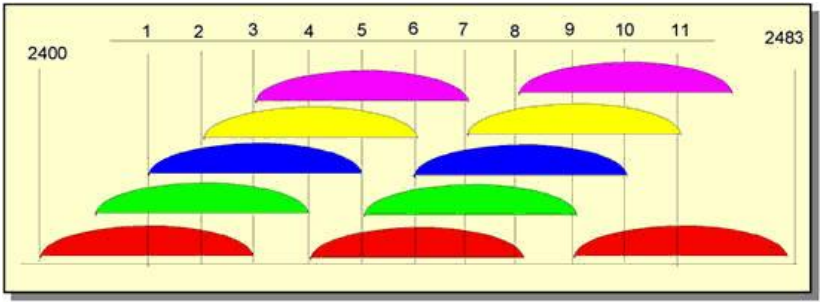


Рисунок 5.3 – Спектри 11 каналів
для стандарту 802.11n

Необхідно відзначити, що для стандарту 802.11n є такі номера каналів, які не перекриваються:

– для ширини каналу 20 МГц: 1, 6, 11;
– для ширини каналу 40 МГц: 3, 11.

У частотному діапазоні 5 ГГц є 23 канали, які не перекриваються, шириною по 20 МГц. Діапазон 5 ГГц складається тільки з каналів, які не перекриваються. Це пов’язано з тим, що на цій частоті перекриття створює колізії, за яких практично не можливо працювати.

У цьому частотному діапазоні можна використовувати не тільки ширину 20 МГц або 40 МГц, а і канали шириною 80 МГц.

Згідно з Рішенням НКРЗІ № 393 від 25.07.2017 «Про внесення змін у додатки до рішення НКРЗІ» від

12.01.2012 № 18 [4] в Україні дозволені канали в діапазоні 5 ГГц, наведені на рисунку 5.4.

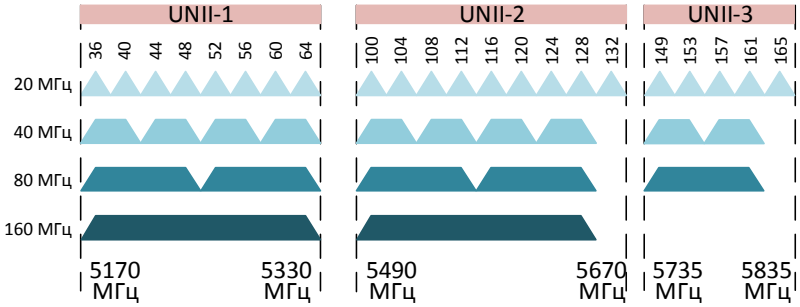


Рисунок 5.4 – Схема каналів у діапазоні 5 ГГц

Сітка центральних частот згідно з блоками каналів UNII-1, UNII-2, UNII-3 така:

1) для ширини смуги випромінювання 20 МГц – це канали 36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 149, 153, 157, 161, 165. Центральні частоти каналів шириною смуги випромінювання 20 МГц: 5180 МГц, 5200 МГц, 5220 МГц, 5240 МГц, 5260 МГц, 5280 МГц, 5300 МГц, 5320 МГц, 5500 МГц, 5520 МГц, 5540 МГц, 5560 МГц, 5580 МГц, 5600 МГц, 5620 МГц, 5640 МГц, 5660 МГц, 5745 МГц, 5765 МГц, 5785 МГц, 5805 МГц, 5825 МГц;

2) для ширини смуги випромінювання 40 МГц – це канали 38, 46, 54, 62, 102, 110, 118, 126, 151, 159. Центральні частоти каналів шириною смуги випромінювання 40 МГц: 5190 МГц, 5230 МГц, 5270 МГц, 5310 МГц, 5510 МГц, 5550 МГц, 5590 МГц, 5630 МГц, 5755 МГц, 5795 МГц;

3) для ширини смуги випромінювання 80 МГц – це канали 42, 58, 106, 122, 155. Центральні частоти каналів шириною смуги випромінювання 80 МГц: 5210 МГц, 5290 МГц, 5530 МГц, 5610 МГц, 5775 МГц;

4) для ширини смуги випромінювання 160 МГц – це канали 50, 114. Центральні частоти каналів шириною смуги випромінювання 160 МГц: 5300 МГц, 5570 МГц [4].

З рисунка 5.5 бачимо, що в разі підключення побутових роутерів і маршрутизаторів практично ніхто не проводить аналізу каналів на перекриття. Це викликає колізії в діапазоні 2,4 ГГц.

Завантаження діапазону 5 ГГц (рис. 5.6) досить низьке, тому питання завантаження каналів і колізій у цьому діапазоні сьогодні не актуальне.

Dynamic Frequency Selection

Завантаженість діапазону 5 ГГц на сьогодні є досить низька, але, незважаючи на проблеми завантаженості діапазону 2,4 ГГц, застосування технології DFS може в перспективі покращити стабільність бездротових мереж. При чому потрібно зосередитися не лише на визволенні каналів під роботу радарів та метеорологічного обладнання, а й під завантаження радіопростору навколо точки доступу. Якість розробленого ПЗ та алгоритмів його роботи буде досить сильно впливати на стабільність та надійність роботи бездротової мережі.

Потужність випромінювання

У деяких випадках на точці доступу рекомендують знизити потужність сигналу Wi-Fi до рівня 50–75 %. Це пояснюють тим, що використання занадто великої випромінюваної потужності сигналу Wi-Fi далеко не завжди гарантує, що мережа буде працювати стабільно і швидко. Якщо радіоефір, у якому працює точка доступу, сильно завантажений (рис. 5.5), то буде проявлятися вплив внутрішньоканальних і міжканальних перешкод. Такі перешкоди впливають на стабільність і швидкість мережі, унаслідок цього спостерігається різке збільшення рівню

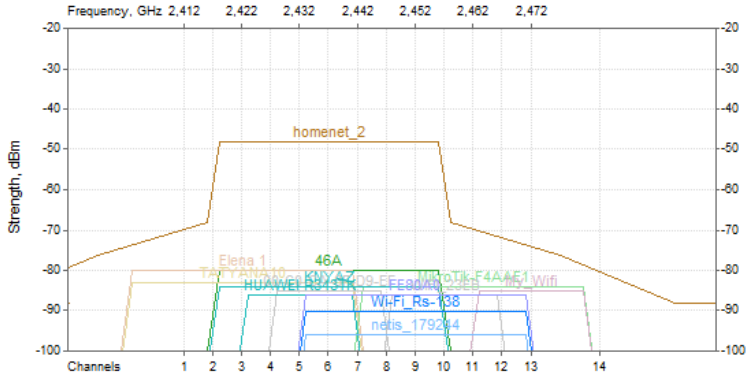


Рисунок 5.5 – Спектри реального завантаження каналів для стандарту 802.11n

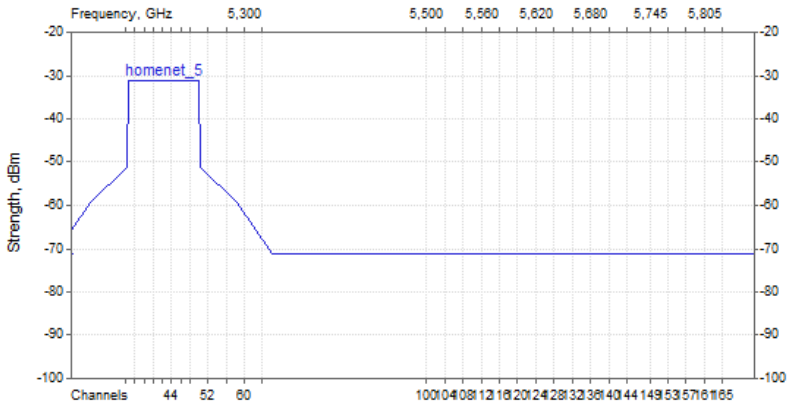


Рисунок 5.6 – Спектри реального завантаження каналів для стандарту 802.11ac

шуму, що призводить до низької стабільності зв'язку через постійне пересилання пакетів.

У цьому разі рекомендують знизити потужність передавачів у точках доступу.

Другою проблемою з потужністю випромінювання є те, що потужність передавача точки доступу в роутері зазвичай удвічі – тричі вище, ніж на клієнтських мобільних пристроях (ноутбук, смартфон, планшет).

У зоні покриття мережі можуть бути такі місця, де клієнт буде бачити точку доступу добре, а точка доступу буде бачити клієнта погано або взагалі не бачити. Останній випадок – це та ситуація, коли сигнал на клієнтському пристрої є, а зв'язку немає. Тобто в каналі зв'язку виникає асиметрія, пов'язана з різними значеннями потужностей і чутливостей приймачів.

Для забезпечення стабільного рівня сигналу необхідне досягнення якомога більш симетричного з'єднання між клієнтським пристроєм і точкою доступу. Для цього може бути необхідне зниження потужності передавача в точці доступу.

Завантаженість діапазону частот іншими приладами.

Крім бездротового обладнання на роботу мережі досить сильно впливають пристрої, функціонування яких може призводити до електромагнітного випромінювання.

Впливають на роботу бездротових мереж пристрої Bluetooth, бездротові клавіатури і миші, які працюють у частотному діапазоні 2,4 ГГц або 5 ГГц, мікрохвильові печі, електромотори, бездротові динаміки, бездротові телефони та інші бездротові пристрої, зовнішні джерела електромагнітного випромінювання.

Перешкоди

Перешкоди (стіни, стелі, меблі, металеві двері тощо), розміщені між Wi-Fi-пристроями, можуть

відбивати або поглинати радіосигнали, що призводить до часткової або повної втрати сигналу. У містах із багатоповерховою забудовою основною перешкодою для радіосигналу є будівлі. Наявність капітальних стін (бетон + арматура), листового металу, штукатурки на стінах, сталевих каркасів тощо впливає на якість сигналу та може значно погіршувати роботу Wi-Fi-пристроїв.

Усередині приміщення створювати перешкоди радіосигналу також можуть дзеркала і тоновані вікна, теплоізоляція, алюмінієвий профіль. У таблиці 5.3 наведені орієнтовні втрати ефективності сигналу Wi-Fi під час проходження через різні середовища. Дані наведені для мережі, що працює в частотному діапазоні 2,4 ГГц.

Таблиця 5.3 – Втрати ефективності сигналу під час проходження через різні перешкоди

Перешкода	**Додаткові втрати dB**	**Ефективна відстань, %**
Відкритий простір	0	100
Вікно без тонування (без металізованого покриття)	3	70
Вікно з тонуванням (з металізованим покриттям)	5–8	50
Дерев’яна стіна	10	30
Міжкімнатна стіна (15,2 см)	15–20	15
Капітальна стіна (30,5 см)	20–25	10
Бетонна підлога (стеля)	15–25	10–15
Монолітне залізобетонне перекриття	20–25	10

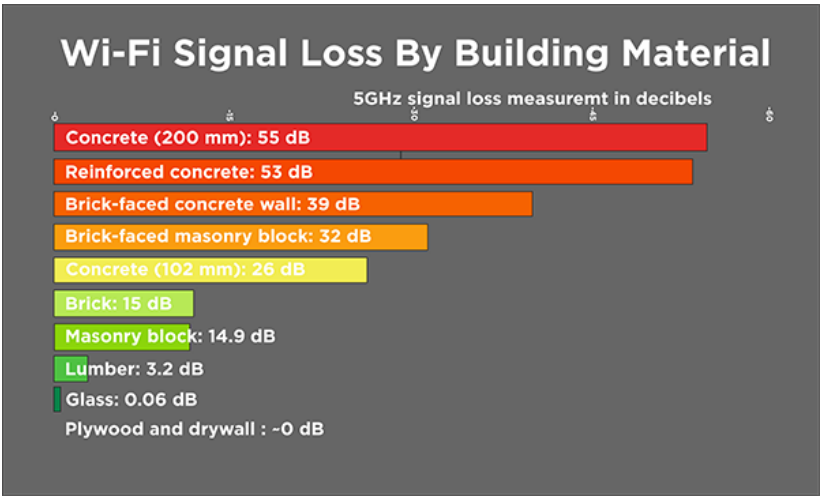


Рисунок 5.7 – Втрати сигналу під час проходження різних перешкод [5]

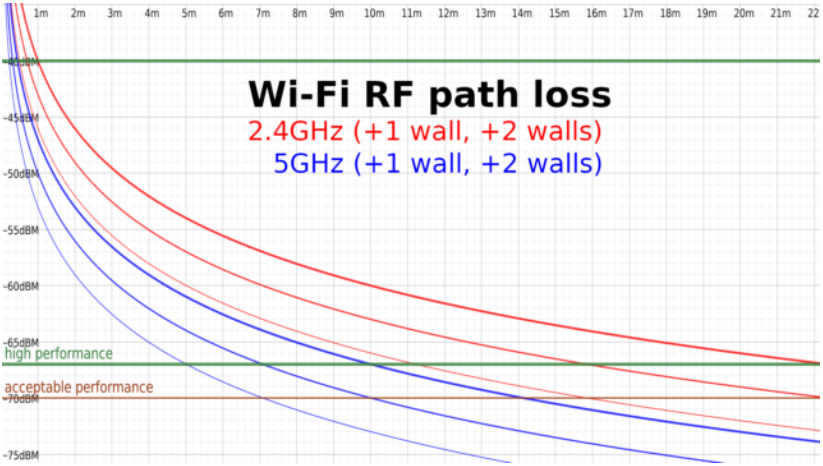


Рисунок 5.8 – Порівняльні залежності загасання 2,4 ГГц та 5 ГГц [6]

На рисунку 5.7 наведені орієнтовні втрати сигналу діапазону 5 ГГц під час проходження різних матеріалів.

Порівняння таблиці 5.3 та рисунка 5.7 демонструє, що для діапазону 5 ГГц зниження якості сигналу та його радіуса відбувається набагато швидше, ніж для діапазону 2,4 ГГц.

Для якісного порівняння загасання сигналів необхідно проводити дослідження обох стандартів за однакових умов.

Аналіз рисунка 5.8 свідчить, що загасання діапазону 5 ГГц відбувається досить сильно. Це обов'язково потрібно враховувати під час побудови та модернізації бездротових мереж. Радіус якісного сигналу під час переходу на новий стандарт зменшується вдвічі.

Рекомендації

Розглянутий стандарт 802.11ac (Wi-Fi 5) порівняно з 802.11n (Wi-Fi 4) має великі переваги та позитивні тенденції для збільшення швидкості передавання сигналу. Це дозволить значно розширити сферу застосування стандарту.

Істотним недоліком 802.11ac (Wi-Fi 5) є досить сильне загасання та більша чутливість електромагнітних хвиль до перешкод діапазону 5 ГГц порівняно з діапазоном 2,4 ГГц.

За коректного підходу основний недолік можна використати для збільшення кібербезпеки бездротової мережі: без застосування додаткових матеріалів практично до нуля гасити бездротову мережу побутовими матеріалами та перешкодами.

СПИСОК ЛІТЕРАТУРИ

1. Generational Wi-Fi® User Guide October 2018 [Електронний ресурс]. – Режим доступу : https://www.wi-fi.org/download.php?file=/sites/default/files/private/Generational_Wi-Fi_User_Guide_20181005.pdf .

2. Lopez-Perez D. IEEE 802.11be Extremely High Throughput: The Next Generation of Wi-Fi Technology Beyond 802.11ax / A. Garcia-Rodriguez, L. Galati-Giordano, M. Kasslin // IEEE Communications Magazine. – 2019, September. – Vol. 57, No. 9. – P. 113–119. – DOI: 10.1109/MCOM.001.1900338.

3. Huang H. Fast Beamforming Design via Deep Learning / Y. Peng, J. Yang, W. Xia // IEEE Transactions on Vehicular Technology. – 2020, Jan. – Vol. 69, No. 1. – P. 1065–1069. – DOI: 10.1109/TVT.2019.2949122.

4. Про схвалення узагальнених умов застосування радіоелектронних засобів та випромінювальних пристроїв: Рішення № 18 від 12.01.2012 [Електронний ресурс]. – Режим доступу : https://nkrzi.gov.ua/index.php?r=site/index&pg=38&id=805&language=uk.

5. Fresnel zone & loss [Електронний ресурс]. – Режим доступу : https://interline.pl/Information-and-Tips/FRESNEL-ZONE-LOSS.

6. The Ars Technica semi-scientific guide to Wi-Fi Access Point placement loss [Електронний ресурс]. – Режим доступу : https://arstechnica.com/gadgets/2020/02/the-ars-technica-semi-scientific-guide-to-wi-fi-access-point-placement.

РОЗДІЛ 6
ТИПОВІ ПОМИЛКИ В ОБ'ЄКТНО-ОРІЄНТОВАНОМУ ПРОГРАМУВАННІ ТА ЇХ ВИПРАВЛЕННЯ ДЛЯ ДОСЯГНЕННЯ БЕЗПЕКИ JAVA-ДОДАТКІВ

В. А. Колесніков

У цьому розділі обговорюємо типові помилки під час написання об'єктно-орієнтованих Java-додатків. Такі помилки призводять до небезпечного коду, який легко зламати та важко підтримувати. Наводимо найкращі практики та керівні принципи інжинірингу програмного забезпечення, щоб уникнути багатьох таких помилок. Незважаючи на те, що обговорення відбувається в загальному сенсі і тому наведені підходи можуть бути застосованими до багатьох об'єктно-орієнтованих мов, усі наведені в цьому розділі приклади написані мовою Java в середовищі Eclipse. Основа для всіх наведених прикладів є реальною. Усі приклади спостерігались у різних формах і взяті з особистого досвіду з галузі програмного забезпечення та академічного досвіду. Цей розділ буде корисним для студентів та тих, хто вивчає об'єктно-орієнтоване програмування, а також для викладачів, які викладають курси з об'єктно-орієнтованого програмування, щоб переконатись, що вони вказують своїм учням на погану практику та напрямки подолання такої практики.

Об'єктно-орієнтований підхід ґрунтується на трьох китах – класах, успадкуванні та поліморфізмі. Обмежимо наше обговорення лише класами та об'єктами, а решту залишимо для подальшої роботи. Читаючи цей матеріал, важливо пам'ятати про основні причини, що призвели до створення об'єктно-орієнтованої парадигми. Як і сьогодні,

оскільки розробники програмного забезпечення витрачали більшу частину часу на роботу з наявним кодом, повторне використання та легкість підтримки коду були одними з основних причин необхідності пошуку кращої альтернативи процедурному програмуванню.

Далі коротко переглянемо основну об'єктно-орієнтовану термінологію, яку використовуємо в цьому розділі. *Класи* – це нові типи даних, що описують *об'єкти,* які можна створити на основі цих класів. *Об'єкти,* або *екземпляри,* є змінними цих типів даних. Класи визначають дані, що містяться в об'єктах цього класу (ці дані називаються *атрибутами,* або *полями*) та дії, які можуть виконувати об'єкти цього класу (ці дії називаються *методами,* або *поведінкою*). Тобто атрибути зберігають дані про об'єкт, а методи описують операції, які об'єкт може виконувати. Тоді зазначають, що класи та їхні об'єкти *інкапсулюють* у собі атрибути та методи (рис. 6.1). Тобто все, що потрібно класу, міститься всередині цього класу.

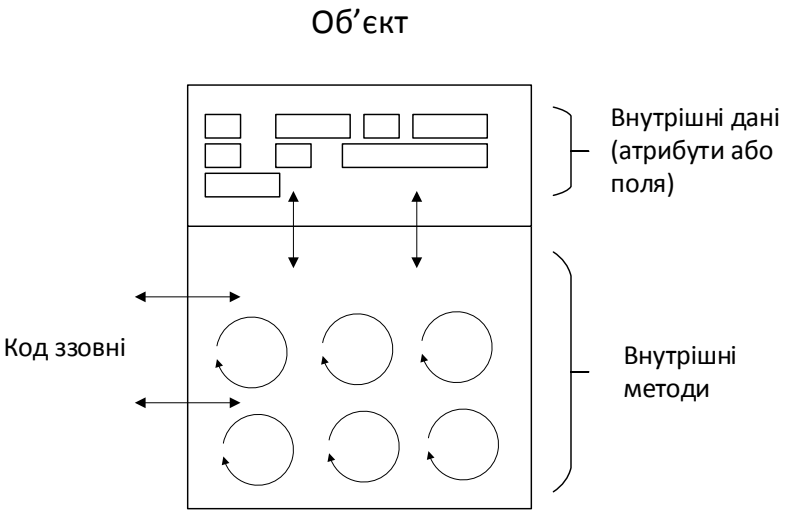


Рисунок 6.1 – Об'єкт, атрибути та методи

Об’єкти можуть спілкуватися між собою, але їм, зазвичай, не дозволяють знати деталі реалізації інших об’єктів, деталі реалізації приховані всередині об’єкта. На рисунку 6.1 атрибути наведені в окремій частині об’єкта, до якої мають доступ лише внутрішні методи. Зовнішній код зазвичай має доступ лише до внутрішніх методів, які, зі свого боку, можуть працювати з внутрішніми даними. Це відомо як *приховування інформації*. Концепція приховування інформації є важливим аспектом програмної інженерії та безпечного програмування.

Далі перелічимо типові помилки, які допускають в об’єктно-орієнтованому програмуванні, обговоримо деякі з них більш детально та пояснимо, як уникнути їх, втілюючи на практиці принципи безпечного програмування та програмної інженерії. Цей список аж ніяк не вичерпний. Деякі з цих помилок і керівні принципи щодо їхнього уникнення обговорюють в [1–5]. Усі матеріали, наведені в цьому розділі, ґрунтуються на широкому досвіді роботи в академічних колах та галузі програмного забезпечення.

Загальні проблеми, на які варто звернути увагу:

1. Створення класу для обчислення формули.

2. Додавання непотрібних атрибутів класу, які не описують стан об’єкта.

3. Використання застарілих атрибутів.

4. Використання публічних атрибутів класу.

5. Базування коду, який використовує клас, на внутрішній реалізації класу.

6. Уникання використання гетерів і сетерів.

7. Відсутність перевірки параметрів на валідність у сетерах.

8. Дублювання коду в сетерах і конструкторах замість виклику сетерів у конструкторах.

9. Виконання введення користувачем у методах класу.

10. Створення об’єкта в невалідному стані.

Розглянемо три згадані проблеми більш докладно.

Створення класу для обчислення формули

Коли потрібен результат обчислення формули, використання класу для подання формули як на рисунку 6.2, очевидно, є надмірним і зловживанням об’єктно-орієнтованим підходом. Натомість для виконання необхідного обчислення потрібно використовувати функцію.

```
1 // class to calculate a formula
2 // bad practice
3 public class Formula {
4     private int a;
5     private int b;
6     private int c;
7
8     // Constructor
9     public Formula(int a, int b, int c) {
10        this.a = a;
11        this.b = b;
12        this.c = c;
13    }
14
15    // returns the result of the calculation - the sum of attributes
16    public int getResult() {
17        return a + b + c;
18    }
19 }
```

Рисунок 6.2 – Приклад поганої практики – клас *Formula* для подання формули для обчислення

Як приклад поганої практики цієї проблеми на рисунку 6.2 наведений клас *Formula*, який використовуємо для обчислення суми трьох чисел. Три числа подані як три атрибути класу. Конструктор використовуємо для

ініціалізації трьох атрибутів. Для повернення результату обчислення використовуємо метод *getResult()*.

Драйвер для класу *Formula* та результати виконання програми наведені на рисунку 3. Очевидно, такий підхід до обчислення результату формули є надмірним. Єдиною корисною частиною коду на рисунку 6.2 є метод *getResult()*, наведений у рядках 16–18, який робить фактичний розрахунок.

```
1 // This program is a driver for testing a class
2
3 public class Driver {
4     public static void main(String[] args) {
5         int a = 1;
6         int b = 2;
7         int c = 3;
8
9         // create a formula object
10        Formula myFormula = new Formula(a, b, c);
11
12        // get the result
13        System.out.println("The sum of " + a + ", " + b + " and " +
14            c + " is " + myFormula.getResult());
15    }
16 }
```

Вивід програми:

```
The sum of 1, 2 and 3 is 6
```

Рисунок 6.3 – Драйвер для класу *Formula* з рисунка 6.2 та вивід програми

Кращий підхід – це створити функцію, яка прийме необхідну інформацію через аргументи, виконає обчислення і поверне результат обчислення, як наведено на рисунку 6.4. Додатковий клас тоді не потрібен і немає необхідності створювати об’єкти.

```
1 // This program is a driver for testing a class
2 // good practice
3
4 public class Driver {
5     public static void main(String[] args) {
6         int a = 1;
7         int b = 2;
8         int c = 3;
9
10        // get the result
11        System.out.println("The sum of " + a + ", " + b + " and " +
12            c + " is " + getSum(a, b, c));
13    }
14
15    // returns the result of the calculation - the sum of attributes
16    public static int getSum(int a, int b, int c) {
17        return a + b + c;
18    }
19 }
```

Вивід програми:

```
The sum of 1, 2 and 3 is 6
```

Рисунок 6.4 – Використання функції для обчислення формули

Мотивацією поганому прикладу, про який тут ішлося, могло бути бажання продемонструвати перший і простий приклад класу для студентів. Проблема полягає в тому, що такий приклад є контпродуктивним під час викладання теми про об'єктно-орієнтоване програмування і суперечить реальній причині, чому був розроблений об'єктно-орієнтований підхід загалом.

Використання застарілих атрибутів

Щоб обговорити цю проблему, використаємо клас *Rectangle* із двома атрибутами *length* та *width*, а також відповідними гетерами та сетерами і конструктором, як наведено на рисунку 6.5.

```
1 // class Rectangle
2 public class Rectangle {
3     private double length;
4     private double width;
5
6     // Constructor
7     public Rectangle(double len, double w) {
8         setLength(len);
9         setWidth(w);
10    }
11
12    // getLength() method returns the value of the length attribute
13    public double getLength() {
14        return length;
15    }
16
17    // getWidth() method returns the value of the width attribute
18    public double getWidth() {
19        return width;
20    }
21
```

Рисунок 6.5 – Клас *Rectangle* (неповна версія)

Якщо потрібно надати класу *Rectangle* можливість повідомляти користувачам його об’єктів площу цих об’єктів, одним із варіантів є додавання атрибута *area* до класу *Rectangle* і збереження відповідного значення площі в ньому, як наведено на рисунку 6.6. Проблема з таким підходом полягає в тому, що значення атрибута *area* може «застаріти». Коли значення атрибута залежить від інших даних і значення атрибута не змінюється, коли ці дані змінюються, відзначають, що значення атрибута стає застарілим. У цьому разі стан об’єкта стає недійсним або нестабільним. Тобто якщо площа прямокутника зберігається в атрибуті класу, значення цього атрибута стає недійсним, коли змінюється значення або атрибута *length*, або атрибута *width*.

```
1 // class Rectangle
2 public class Rectangle {
3     private double length;
4     private double width;
5     private double area;
6
7     // Constructor
8     public Rectangle(double len, double w) {
9         setLength(len);
10        setWidth(w);
11
12        area = len * width;
13    }
14
15    // getLength() method returns the value of the length attribute
16    public double getLength() {
17        return length;
18    }
19
20    // getWidth() method returns the value of the width attribute
21    public double getWidth() {
22        return width;
23    }
24
25    // getArea() method returns the area of the rectangle
26    public double getArea() {
27        return area;
28    }
29
```

Рисунок 6.6 – Приклад поганої практики – клас *Rectangle* з атрибутом *area* (неповна версія)

Такого підходу варто уникати. Зберігання значення, яке є результатом обчислення в атрибуті класу – це погана практика. Більш коректним підходом у такому разі є надання класу методу, який повертає інформацію, що є результатом підрахунку, а потім виклик цього методу, коли ця інформація потрібна. Такий підхід гарантує, що обчислення виконують із використанням дійсних значень його компонентів і завжди отримують правильний результат обчислення.

Щоб клас *Rectangle* мав можливість повертати інформацію про площу своїм користувачам, додаємо до класу метод *getArea()*, який обчислює та повертає площу прямокутника. Площа прямокутника обчислюється множенням довжини прямокутника на його ширину. Значення довжини та ширини зберігаються у відповідних атрибутах класу *Rectangle*. Цей підхід наведено на рисунку 6.7.

```
1 // class Rectangle
2 public class Rectangle {
3     private double length;
4     private double width;
5
6     // Constructor
7     public Rectangle(double len, double w) {
8         setLength(len);
9         setWidth(w);
10    }
11
12    // getLength() method returns the value of the length attribute
13    public double getLength() {
14        return length;
15    }
16
17    // getWidth() method returns the value of the width attribute
18    public double getWidth() {
19        return width;
20    }
21
22    // getArea() method returns the area of the rectangle
23    public double getArea() {
24        return length * width;
25    }
26
```

Рисунок 6.7 – Приклад більш коректної практики – клас *Rectangle* з методом *getArea()* (неповна версія)

Виконання введення інформації користувачем у методах класу

Остання проблема, яку розглянемо, стосується отримання інформації, введеної користувачем, для ініціалізації атрибутів класу в одному з методів класу. Як добра практика введення даних користувачем не повинно відбуватися в методах класу, така інформація повинна бути отримана в драйвері, де потім можуть бути створені об'єкти та задіяний конструктор, який використовуємо для ініціалізації стану об'єктів.

Приклад отримання інформації від користувача для ініціалізації атрибутів класу одним із методів класу наведений для класу *Rectangle* на рисунку 6.8, драйвер для класу *Rectangle* з рисунка 6.8 наведений на рисунку 6.9.

У цьому прикладі клас *Rectangle* має два атрибути: *length* і *width*. Метод *input()* використовуємо для отримання інформації від користувача для ініціалізації двох атрибутів. Метод *output()* відображає інформацію про об'єкт. У драйвері спочатку створюємо об'єкт *box* типу *Rectangle*, а потім викликаємо його метод *input()* для ініціалізації об'єкта. Після ініціалізації об'єкт можна використовувати.

Мотивація написання такого коду полягає в тому, щоб мати якомога менше коду та написати програму якомога швидше. Крім того, основна програма має вигляд меншої. Ще однією мотивацією є те, що цей код є простим прикладом для введення поняття класів на початку курсу з об'єктно-орієнтованого програмування.

```java
import java.util.Scanner;

// class Rectangle
public class Rectangle {
    private double length;
    private double width;

    // user input method
    public void input() {
        // Create a Scanner object to read input.
        Scanner keyboard = new Scanner(System.in);

        // Get the length
        System.out.print("Enter length: ");
        length = keyboard.nextDouble();

        // Get the width
        System.out.print("Enter width: ");
        width = keyboard.nextDouble();
    }

    // output method
    public void output() {
        System.out.println("Rectangle with length of " + length +
            " and width of " + width);
    }
}
```

Рисунок 6.8 – Приклад поганої практики – клас *Rectangle* з методом для введення інформації від користувача

Така мотивація є некоректною з декількох причин. Нагадуємо, що основними причинами використання об’єктно-орієнтованого підходу є повторне використання коду та обслуговування коду. Концепція *write once use many times* порушується, оскільки клас *Rectangle* більше не можна використовувати повторно в іншому контексті.

```
1 // This program is a driver for testing a class
2
3 public class Driver {
4     public static void main(String[] args) {
5         Rectangle box = new Rectangle();
6
7         box.input();
8         box.output();
9     }
10 }
```

Рисунок 6.9 – Драйвер для класу *Rectangle* з рисунка 6.8

Розглянемо варіант, коли потрібно створити об’єкти класу *Rectangle*, не отримуючи інформації від користувача, наприклад, для якогось моделювання. Для цього доведеться створити ще один клас *Rectangle* з нуля, оскільки клас *Rectangle* на рисунку 6.8 не підходить для нового контексту.

Інша проблема виникає, коли нам потрібно змінити спосіб уведення інформації користувачем. У цьому разі доведеться внести зміни до класу *Rectangle*. Це вимагатиме повторного тестування класу та відновлення бібліотек, у яких цей клас з’являється. Усі інші проєкти, що залежать від такого класу, можливо, доведеться змінити також, перевірити їх і перекомпілювати.

Жодна з цих проблем не виникає, якщо дотримуватися належної практики програмної інженерії, коли код для введення користувачем не розміщується в методі класу. Такий підхід наведено на рисунку 6.10, а драйвер – на рисунку 6.11. На рисунку 6.10 клас *Rectangle* має два атрибути *length* та *width*, як і раніше. Але цього разу клас також має гетери та сетери для кожного атрибута. Весь код для введення користувачем інформації міститься в драйвері, як і повинно бути.

```
1 // class Rectangle
2 public class Rectangle {
3     private double length;
4     private double width;
5
6     // Constructor
7     public Rectangle(double len, double w) {
8         setLength(len);
9         setWidth(w);
10    }
11
12    // setLength() method sets the length attribute of the class
13    public void setLength(double len) {
14        if (len > 0)
15            length = len;
16        else {
17            System.out.println(
18                "Trying to set the length to an invalid value of " +
19                len + ".\n" +
20                "The length of a rectangle " +
21                "should be a positive value.\n" +
22                "The program will now end.");
23            System.exit(1);
24        }
25    }
26
27    // setWidth() method sets the width attribute of the class
28    public void setWidth(double w) {
29        if (w > 0)
30            width = w;
31        else {
32            System.out.println(
33                "Trying to set the width to an invalid value of " +
34                w + ".\n" +
35                "The width of a rectangle " +
36                "should be a positive value.\n" +
37                "The program will now end.");
38            System.exit(1);
39        }
40    }
41
42    // getLength() method returns the value of the length attribute
43    public double getLength() {
44        return length;
45    }
46
47    // getWidth() method returns the value of the width attribute
48    public double getWidth() {
49        return width;
50    }
51
```

Рисунок 6.10 – Клас *Rectangle* (частина 1 з 2)

```
52      // getArea() method returns the area of the rectangle
53      public double getArea() {
54          return length * width;
55      }
56
57      // this method returns a string representing the state
58      // of a Rectangle object
59      public String toString() {
60          // Create a string describing the rectangle
61          String str = "Rectangle with length of " + length +
62              " and width of " + width;
63
64          // Return the string
65          return str;
66      }
67  }
```

Рисунок 6.10 – Клас *Rectangle* (частина 2 з 2)

```
 1 import java.util.Scanner;
 2
 3 // This program is a driver for testing a class
 4
 5 public class Driver {
 6     public static void main(String[] args) {
 7         // Create a Scanner object to read input.
 8         Scanner keyboard = new Scanner(System.in);
 9
10         // Get the length
11         System.out.print("Enter length: ");
12         double length = keyboard.nextDouble();
13
14         // Get the width
15         System.out.print("Enter width: ");
16         double width = keyboard.nextDouble();
17
18         Rectangle box = new Rectangle(length, width);
19
20         System.out.println(box);
21     }
22 }
```

Рисунок 6.11 – Драйвер для класу *Rectangle* з рисунка 6.10

У разі застосування вищенаведеного підходу можна легко повторно використати клас *Rectangle* з рисунка 6.10 у будь-якому іншому контексті. Для цього створюємо об'єкти класу *Rectangle*, використовуючи конструктор класу. Об'єкти, створені таким способом, перебувають у стабільному стані відразу після їх створення і, отже, можуть бути використані відразу. Будь-яка зміна способу введення інформації користувачем не впливає на клас *Rectangle*.

Можлива така мотивація для використання коду, наведеного на рисунках 6.8 і 6.9, – показати простий приклад класу на початку курсу з об'єктно-орієнтованого програмування. Але такий підхід також не витримує жодної критики. Студентів не варто знайомити з поганою практикою як способом навести приклади важливим поняттям. Це створює шкідливі звички в коді студентів, оскільки вони використовуватимуть такі приклади, як шаблони для власного коду. Концепції програмування завжди потрібно вводити на прикладах, які відповідають належній практиці програмування. Варто також наводити приклади поганої практики, але лише як приклади.

Висновки

Розглянуто кілька типових помилок, які допускають в об'єктно-орієнтованому програмуванні. Також розглянуті принципи належної практики, яких необхідно дотримуватися, а саме: уникати створення класу для обчислення формули, використання застарілих атрибутів та введення інформації користувачем у методах класу. Уникнення таких помилок дає можливість більшої безпеки Java-додатків. Такий код набагато важче зламати. З погляду програмної інженерії дотримання належної практики дає той результат, що програмний код простіше повторно використовувати та підтримувати.

СПИСОК ЛІТЕРАТУРИ

1. Pressman R. S. Software Engineering / R. S. Pressman. – 8 ed. – McGraw-Hill, 2015. – ISBN 978-0-07-802212-8.

2. Bishop M. Computer Security: Art and Science / M. Bishop. – 2 ed. – Addison-Wesley Professional, 2019. – ISBN 978-0-321-71233-2.

3. Bishop M. Teaching secure programming / M. Bishop // IEEE Security & Privacy. – 2005, Sept. – Oct. – Vol. 3, No. 5. – P. 54–56. – DOI: 10.1109/MSP.2005.133.

4. Taylor B. Threading secure coding principles and risk analysis into the undergraduate computer science and information systems curriculum / B. Taylor // InfoSecCD ’06 : Proceedings of the 3rd annual conference on Information security curriculum development, September, 2006. – P. 24–29. – DOI: 10.1145/1231047.1231053.

5. Winters T. Software Engineering at Google: Lessons Learned from Programming Over Time / T. Winters. – O’Reilly, 2020. – Feb. 28.

РОЗДІЛ 7
КОМПЛЕКСНИЙ ПІДХІД ДО ЗАХИСТУ СКБД У КОНТЕКСТІ ДИСЦИПЛІНИ «ЗАХИЩЕНІ БАЗИ ДАНИХ ТА ІНФОРМАЦІЙНІ СИСТЕМИ»

Б. О. Кузіков

Дисципліна «Захищені бази даних та інформаційні системи» займає одне з центральних місць у навчальній програмі студентів спеціальності 125 «Кібербезпека» третього курсу. Місією дисципліни є узагальнення матеріалу попередньо вивчених дисциплін, його структурування та підготовка до написання бакалаврської роботи. Виконання практичних і теоретичних завдань дисципліни повинні забезпечити студентові формування практичних навичок та вмінь із:

– проєктування бази даних;

– розроблення чи застосування готового прикладного забезпечення на базі систем керування базами даних;

– використання бази даних на основі конкретних бізнес-вимог до інформаційної системи;

– аналізування аспектів інформаційних систем, важливих для забезпечення безпеки даних та сервісів.

Дисципліна викладається впродовж 5-го та 6-го семестрів навчання, пов’язаних загальною понятійною базою і логікою викладання матеріалу. У 5-му семестрі акцентується на базових питаннях розроблення та використання інформаційних систем. Матеріал відповідає програмі курсів «SQL Fundamentals» та «Introduction to Relational Database and SQL» іноземних ЗВО. Основною метою цього етапу є набуття базових практичних навичок за такими напрямками:

– структурування та відділення бізнес-вимог замовника, їх формалізація у вигляді Data Flow-діаграм;

– проєктування та нормалізація структури сховища даних на прикладі реляційних баз даних, документування структури сховища за допомогою Entity-Relationship-діаграм;

– використання мови SQL у межах стандарту SQL:2003 (включаючи ієрархічні запити).

Вище перелічене забезпечується через набір відповідних лекційних матеріалів, практичних завдань та контрольної роботи. Водночас ключове місце займає саме контрольна робота, яка проводить студента через основні етапи проєктування і розроблення сховища даних інформаційної системи на основі наперед заданих типових сценаріїв використання такої системи та обмежень щодо можливих засобів їх реалізації.

Ключовим під час вивчення дисципліни є 6-й семестр, який розкриває різноманітність сценаріїв та засобів, пов'язаних із питаннями експлуатації й безпеки інформаційних систем і СКБД як їх складової частини, що є ключовим під час вивчення дисципліни. Розгляд зосереджується на типовому рішенні зі трирівневої архітектури з тонким клієнтом – браузером (див. рис. 7.1), але не обмежується нею. Нижче наведені теми, прив'язані до етапів взаємодії користувача із СКБД.

З'єднання з БД

Взаємодія із СКБД починається з установлення з'єднання з базою даних із подальшим відкриттям користувацької сесії. Кожен з етапів – установлення з'єднання, авторизація, ініціація сесії – є критичним із точки зору безпеки та управління ресурсами.

Одним із ключових питань у цьому разі є співвідношення фізичних користувачів та користувачів СКБД. Варіанти з наскрізним використанням користувача є типовими для моделей із товстим клієнтом у

контрольованому середовищі. Прикладом останнього підходу може бути користувач *postgres*, який використовується для запускання служб і за замовчуванням використовує однорангову (в термінах postgresql – «peer») авторизацію операційною системою у разі локального доступу без використання мережевого з'єднання.

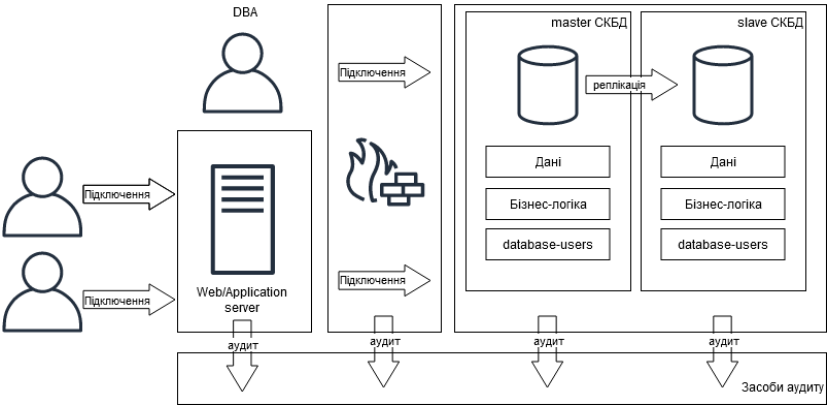


Рисунок 7.1 − Загальна схема інформаційної системи

Більш поширеним варіантом є використання єдиного користувача, за допомогою якого сервер додатків (*application server*) виконує дії від імені різних користувачів бізнес-логіки додатка. Такий підхід дозволяє значно зекономити ресурси та пришвидшити виконання операцій за рахунок використання пулу з'єднань. Водночас потрібно розділяти користувача, який може виконувати обмежений перелік операцій, що відповідають бізнес-логіці додатка, та адміністративних користувачів, які можуть виконувати сервісні операції на кшталт створення нових структур даних чи проведення розширеного аудиту.

Продовженням попереднього питання є перелік способів доступу до сервера СКБД із додатка. Вся взаємодія жорстко обмежена контрольованим API бізнес-логіки чи користувачі можуть реалізовувати власні запити до БД. Останній варіант навіть за умови використання засобів обмеження прав користувача з боку бізнес-логіки вимагає більш глибокого розмежування прав на доступ об'єктів усередині БД засобами самої СКБД.

Питання адміністрування БД повсякчас вимагає застосування користувачів із розширеним переліком прав та/або безпосереднім доступом до сервера бази даних. Водночас виникає питання регламентування такого доступу, виділення СКБД як окремого сегмента мережі з відповідними обмеженнями доступності із зовнішньої мережі тощо. Показовим у цьому сенсі є звіт про вразливість CVE-2019-9193, в якому зазначається, що штатна, задокументована в Postgresql, можливість виконувати команди в ОС від імені користувача є критичною вразливістю.

Одним із типових елементів під час доступі до БД є використання pooling/proxy, що забезпечують консолідацію кількох з'єднань від сервера додатків до меншої кількості з'єднань із СКБД. Дисципліна розглядає цей клас програмного забезпечення на прикладі pgPool та pgBouncer. Зазначається, що на цьому етапі комерційне ПЗ може виконувати додаткові функції мережевого екрана, балансувальника навантаження, перемикання підключень на резервний сервер у разі недоступності основного, виявлення SQL-injection тощо.

Виходячи з перелічених факторів, серед основних порад для цього розділу будуть:

1. Використовуйте рішення класу брандмауера бази даних. Додатковий захист шару, як мінімум, підвищує

прозорість того, що відбувається в СКБД, як максимум, ви можете забезпечити додатковий захист даних.

2. Використовуйте парольні політики. Їх застосування залежить від того, як вибудована ваша архітектура. У будь-якому разі одного пароля в конфігураційному файлі вебдодатків, що підключається до СКБД, мало для захисту.

3. Збагачуйте контекст сесій необхідною інформацією. Під час аналізування дій користувачів ми повинні чітко розуміти, хто виконував певну операцію.

4. Мережеві з'єднання є одним із векторів атаки. Тому з'єднання повинне використовувати шифрування (наприклад, через SSL) або СКБД повинне виділятися як окремий VLAN.

Моніторинг та аудит дій

До цілей, що повинні бути покриті засобами моніторингу, відносять:

– фізичне обладнання та операційну систему хоста СКБД (показники пам'яті, завантаженості CPU, довжина черги I/O тощо). Прикладом систем, що дозволяють вирішити це питання, є atop (локально), Zabbix (централізоване збирання даних на рівні підприємства), DataDog (Monitoring as a Service);

– безпосередньо СКБД, як програмне забезпечення (час виконання запитів, використання індексів, кешів (показник hit/miss), довжина черги оброблення запитів, кількість відкритих з'єднань). Засоби для цієї категорії специфічні для конкретної СКБД. У рамках дисципліни розглядається моніторинг на прикладі postgresql через запити до БД (таблиці pg_stat_activity, pg_stat_all_tables тощо; [6] проводяться аналогії з vsql, vsession у СКБД Oracle), налаштування журналювання (розширення

pg_stat_statements), зовнішніх сервісів (*пропрієтарний* pganalyze та його вільний self-hosted-аналог pgHero);

– моніторинг бізнес-метрик. До цього розділу можна віднести як відстежування роботи СКБД у контексті виконання запитів (у дисципліні розглядається на прикладі NewRelic), так і розширений аудит дій користувачів (на прикладі pgAudit та реалізації власних тригерів із виконанням відповідної лабораторної роботи);

– моніторинг допоміжного програмного забезпечення (стану реплікації, довжини черги на db proxy, спрацьовувань database firewall тощо).

Для закріплення теоретичних навичок із цього розділу студентам пропонуються:

– практичну роботу «Тригери та функції» – реалізація засобів аудиту та версіонування даних власними силами. СКБД для виконання практики визначається відповідно до варіанта. У подальшому матеріал закріплюється у вимогах аудиту даних у завданні до курсової роботи. Окремим питанням у лабораторній роботі розглядається використання тригерів для забезпечення цілісності даних у тих припадках, якщо це неможливо реалізувати засобами підтримання цілісності за посиланням (*referral integrity*);

– практичну роботу «Частково унікальні дані (partial distinct)» – написання запитів, що дозволяють визначити дані, які були актуальними на певний момент часу і змінені в подальшому. До таких даних можна віднести записи в журналі аудиту, історичні дані. Проводять аналогії з технологією Oracle Flashback, зазначають її технічне підґрунтя та обмеження. Окремим питанням розглядають використання подібних запитів у розрізі засобів підтримання консистентності даних (наприклад, додавання обмеження UNIQUE, якщо існують дані, що його порушують).

Окрім безсумнівних переваг, які мають засоби моніторингу, вони створюють потенційний канал для витоку секретних чи чутливих даних. Наприклад, журнали функціонування СКБД PostgreSQL можуть містити таку інформацію:

1) шаблонний опис події, наприклад факт перезапуску сервера чи створення контрольної точки. Така інформація не є конфіденційною;

2) дані, пов'язані з подіями авторизації. Такі записи можуть містити імена чи паролі користувачів;

3) помилки синтаксичного аналізу. Текст запитів може містити різну інформацію, включаючи конфіденційну. Прикладом може бути запис:

```
ERROR: syntax error at or near “postgres” at character 1
STATEMENT: postgres://user:password@host/db_name
```

4) запити до БД. Текст запиту може містити, крім загальних конструкцій, дані які параметризують запит. водночас зазначені параметри можуть належати до чутливої інформації;

5) параметри для підстановки. У разі використання попередньо підготовлених запитів (prepared statement) журнал може містити інформацію щодо того, з якими саме параметрами вони виконувалися. Частина з цієї інформації може бути конфіденційною;

6) дані таблиць. У разі помилок виконання запитів з оператором COPY частина даних може потрапляти до журналу. Потенційно ця інформація може бути конфіденційною;

7) помилки, пов'язані з операційною системою та зовнішнім ПЗ. Текст помилок може містити дані щодо розміщення файлів, дані щодо топології мережі, налаштування окремих команд (наприклад, параметри авторизації в зовнішніх системах);

8) інші повідомлення, що не були віднесені до жодного з попередніх класів. Потенційно ці дані можуть бути секретними.

Зазначені особливості свідчать, що чутливі дані можуть бути ненавмисно передані назовні СКБД, що значно ускладнює експлуатаційне підтримування та оптимізацію. Виходами з цієї ситуації можуть бути обмеження підстав журналювання, фізичний захист файлів на рівні операційної системи, локальне оброблення журналів для моніторингу та виявлення інцидентів, попереднє оброблення для виключення чутливої інформації під час передавання даних до зовнішніх систем моніторингу/аудиту. Прикладом останнього підходу може бути підсистема фільтрації журналів, розроблена для сервісу моніторингу PgAnalyze [7]. Відповідно до налаштувань підсистема маскує чутливі дані символами «X» із боку СКБД, що дозволяє зберегти прийнятний компроміс між наявністю достатньої кількості інформації, необхідної для пошуку джерел (*root causes*) проблем, та збереженням конфіденційності.

Обмеження доступу до даних

Наведемо перелік технологій, використовуваних для захисту даних та доступу до них у комерційних СКБД і з відкритим кодом.

1. Вибіркове керування доступом (*Discretionary access control*, DAC) – розмежування прав із застосуванням матриць доступу. До цього підходу відносять застосування *списків контролю доступу (Access Control List*, ACL) та розмежування прав користувачів на основі ролей (*Role-Based Access Control*, RBAC). Застосування підходів із цього переліку є достатнім при збереженні конференційної інформації. Залежно від ухвалених архітектурних рішень зазначені підходи є

достатніми під час виконання курсової роботи. В окремих практичних роботах розглядається застосування базового синтаксису SQL (GRANT/REVOKE, системні та об'єктні привілеї, перегляд привілеїв засобами SQL, концепція користувачів та ролей, схеми даних) та застосування цих знань для організації розмежування прав користувачів залежно від їх функціональних ролей у контекстів архітектурних рішень курсової роботи. Інші варіанти з переліку розглядаються лише в рамках лекційної частини курсу.

2. Мандатне керування доступом (*Mandatory access control*, MAC) передбачає розмежування прав доступу на основі ієрархії міток доступу («для службового використання», «таємно», «цілком таємно» тощо).

3. Обмеження видимості даних за рядками (*Row level security*, RLS) – налаштування прав користувачів в залежно від специфічних властивостей рядка (як-от власник об'єкта). Завдання курсової роботи передбачає обов'язкове застосування RLS. Але, за бажанням студента, їх реалізація може бути перенесена на рівень сервера додатків.

4. Редагування даних, що відображаються / передаються, – можливість залежно від прав користувача динамічною змінювати актуальні дані окремих колонок на безпечні відповідники. Здебільшого ці можливості винесені за межі СКБД та реалізуються додатковим програмним забезпеченням з класу Database proxy/firewall. Інший приклад застосування такого підходу вже було наведено вище – фільтрація даних під час записування в журнал.

5. Шифрування та обфускація процедур і функцій – це окремі інструменти та засоби, які з роблять код складним для розуміння та зворотного розроблення (*reverse engineering)*. Оброблений таким чином код майже

непридатний для зміни чи рефакторингу. Проте це чи не єдиний спосіб приховувати логіку ліцензійних обмежень чи авторизації на рівні СКБД.

6. Обмеження доступу до файлів на рівні файлової системи. Дії окремих користувачів можуть бути обмежені через налаштування власників, груп та прав доступу на окремі частини файлової системи.

7. Очищення пам'яті. Під час оброблення запиту інформація може дублюватися й тимчасово зберігатись у різних форматах. Наприклад, для підтримання конкурентної роботи користувачів інформація на диску не видаляється, а лише позначається видаленою і буде зберігатися до проведення процедури очищення (*vacuum*). До альтернативних прикладів можна віднести кешування даних в оперативній пам'яті (*shared buffers*), старі копії файлів під час видалення / заміні об'єктів, старі WAL-журнали на диску. Залежно від характеру інформації, що обробляється, в перелічених випадках можна покластися на вбудовані механізми захисту (права доступу до файлів, ізоляція пам'яті процесів, SELinux). Якщо цих механізмів недостатньо – використовують шифрування.

8. Шифрування даних. Залежно від вимог до конфіденційності інформації база даних чи її частина може бути зашифрована. У курсі тема розглядається на прикладі розширення pgCrypto для Postgresql. Він дозволяє зберігати вибрані поля в зашифрованому вигляді. Це корисно, якщо цінність становлять лише деякі дані (імена чи адреси користувачів, номери кредитних карток тощо). Щоб прочитати зашифровані поля, клієнт передає ключ дешифрування, сервер розшифровує дані та видає їх клієнтам. У курсі розглядаються як детерміновані, так і випадкові алгоритми шифруванням, а також підходи з наскрізним шифруванням (на прикладі технології MS SQL Server «Always Encrypted»).

Варто враховувати, що шифрування сильно впливає на продуктивність запитів. Тому окремим питанням є дослідження впливу застосованих підходів та засобів на продуктивність додатка. Для деталізації цього питання під час лекції у форматі демонстрації надводиться використання утиліти pgBench, що дозволяє порівняти показники швидкодії, використання CPU та пам'яті сервера.

Таким чином, курс «Захищені бази даних та інформаційні системи» дозволяє одержати достатній обсяг знань та набути навичок для розв'язання практичних задач захисту даних, оцінювання захищеності системи та доцільності використання наявного програмного забезпечення і організаційних підходів для розв'язування означених задач.

СПИСОК ЛІТЕРАТУРИ

1. Створення та обробка баз даних [Електронний ресурс] : навч. посібник для студ. техн. спец. вищ. навч. закл. / Л. С. Глоба, М. Ю. Тернова, Р. Л. Новогрудська, О. С. Штогрина. – Київ : НТУ України «КПІ», 2013. – 477 с. – Режим доступу : its.kpi.ua/subjects/21/Documents/Навчальний посібник.pdf.

2. Martin Kleppmann Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems Paperback. – O'Reilly Media, 2016. – 400 p.

3. Open Web Application Security Project [Electronic resource]. – Access mode : www.owasp.org.

4. National Vulnerability Database. CVE-2019-9193 Detail [Electronic resource]. – Access mode : https://nvd.nist.gov/vuln/detail/CVE-2019-9193.

5. Lukas Fittl The Most Important Events to Monitor in Your Postgres Logs [Electronic resource]. – Access mode : https://pganalyze.com/ebooks/monitoring-postgres-logs.

6. Postgresql Documentation. The Statistics Collector [Electronic resource]. – Access mode : https://www.postgresql.org/docs/12/monitoring-stats.html.

7. Introducing Log Insights: Realtime Analysis of Postgres Logs pganalyze [Electronic resource]. – Access mode : https://pganalyze.com/blog/introducing-log-insights.

РОЗДІЛ 8
СИСТЕМА СТАНДАРТІВ КОМПЛЕКСНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ ТА УПРАВЛІННЯ ІНФОРМАЦІЙНОЮ БЕЗПЕКОЮ

Н. Л. Барченко, В. К. Ободяк

8.1. Міжнародні стандарти та стандарти зарубіжних країн

Міжнародні стандарти у сфері інформаційної безпеки (ІБ) є орієнтиром під час побудови управління інформаційною безпекою (СУІБ).

Вони також допомагають вирішувати пов'язані з цією діяльністю завдання всіх рівнів: стратегічних, тактичних, операційних.

Кожному фахівцю з ІБ потрібно ознайомитися з найбільш відомими стандартами в цій сфері ІБ, а також навчитися застосовувати їх на практиці.

Вивчення стандартів дає можливість дізнатися про таке:

- термінологію у сфері ІБ;
- загальні підходи до побудови систем ІБ;
- загальноприйняті процеси ІБ;
- конкретні заходи захисту, використовувані в ІБ;
- роль фахівців під час побудови процесів ІБ.

Міжнародні стандарти ІБ постійно вдосконалюються та вбирають кращий досвід фахівців-практиків.

Загальновизнані стандарти міжнародного рівня дозволяють фахівцям спілкуватися між собою і з внутрішніми підрозділами компанії однією мовою з використанням усталених термінів та визначень. Це допомагає обґрунтовувати необхідність тих чи інших

заходів ІБ за допомогою тих формулювань, які зрозумілі для бізнесу.

Також варто відзначити, що ІБ завжди йде поряд з інформаційними технологіями (ІТ).

Тому для підвищення ефективності роботи дуже важливо вміти комунікувати з фахівцями ІТ. У цьому ІБ-фахівцю допомагає вивчення таких стандартів, як ISO, ITIL, COBIT тощо.

ISO

Міжнародна організація стандартизації, ISO (англ. International Organization for Standardization; фр. Organisation internationale de normalisation) – міжнародна організація, що займається випуском стандартів [1].

Цією організацією було розроблено низку стандартів щодо інформаційної безпеки. Найвідомішими з яких є ISO/IEC 27000 та ISO 15408.

Стандарти серії ISO/IEC 27000 розроблені для регулювання управлінням інформаційною безпекою [2]. Найвідоміший стандарт серії – ISO/IEC 27001:2013 визначає аспекти менеджменту інформаційної безпеки і містить кращі практики з вибудовування процесів для підвищення ефективності управління ІБ.

Стандарт декларує ризик-орієнтований підхід, який дозволяє вибрати необхідні заходи та засоби захисту, що найкраще відповідають потребам бізнесу. За результатами впровадження стандарту ISO/IEC 27001:2013 компанія може пройти сертифікацію. ISO/IEC 27001:2013 як модель управління якістю бере за основу Цикл Демінга – Шухарта PDCA (англ. «Plan – Do – Check – Act» – «планування –дія – перевірка – коригування»), що означає безперервне вдосконалення ІБ-процесів.

Стандарт ISO 15408 присвячений загальним критеріям оцінювання безпеки інформаційних технологій (Common Criteria for Information Technology Security Evaluation) [3].

COBIT

COBIT (Control Objectives for Information and Related Technology – «Завдання інформаційних і суміжних технологій») являє собою пакет документів, що складається з понад 40 міжнародних та національних стандартів і настанов у сфері управління IT, аудиту та IT-безпеки [4].

Міжнародна асоціація аудиту і контролю за інформаційними системами (ISACA) провела аналіз та оцінювання практик управління інформаційними системами. Результатом став COBIT, який поєднав найкраще з міжнародних технічних стандартів, стандартів управління якістю, аудиторської діяльності, а також із практичних вимог і досвіду.

ITIL

Бібліотека інфраструктури інформаційних технологій, або ITIL (The IT Infrastructure Library), – це набір публікацій (бібліотека), що описує загальні принципи ефективного використання IT-сервісів. Була розроблена Центральним агентством з обчислювальної техніки і телекомунікацій (Central Computer and Telecommunications Agency, CCTA) у Великобританії. Перед CCTA постало завдання структурування всіх існуючих методів успішного використання IT-ресурсів і розроблення способів їх якісного застосування. Таким чином, бібліотека ITIL покликана оптимізувати набір процесів, спрямованих на забезпечення високої якості IT-послуг та підвищення рівня наданих послуг.

Результатом застосування ITIL в організації має стати підвищення конкурентоспроможності компанії в цілому [5].

Бібліотека ITIL містить окрему книгу з управління IT-безпекою, що має назву Security Management. Вона допомагає гармонійно інтегрувати процес управління IT-безпекою в загальну систему управління інформаційними технологіями в компанії. У матеріалах ITIL не прописані конкретні вимоги до засобів захисту, а лише дається опис загальної організації безпечної роботи IT-сервісів.

У бібліотеці можна знайти як основні принципи вибудовування самого процесу управління ІБ, так і ключові рекомендації з підтримки СУІБ – Системи управління інформаційною безпекою (Information Security Management System, ISMS).

NIST

NIST (National Institute of Standards and Technology) – це американський національний інститут стандартизації, аналог вітчизняного Держстандарту. До складу інституту входить Центр з комп'ютерної безпеки, який публікує з початку 90-х років XX ст. Стандарти, а також детальні роз'яснення та рекомендації в галузі ІБ [6, 7].

Для рекомендацій у сфері ІБ (Special Publications) в NIST виділили спеціальну серію з кодом 800. Серія містить документи, що описують підходи до управління інформаційною безпекою, і висвітлює технічні питання її забезпечення (забезпечення безпеки мобільних пристроїв, захист хмарних обчислень, вимоги до аутентифікації, віддаленого доступу).

Однією з найвідоміших публікацій є стандарт NIST SP 800-53 «Security and Privacy Controls for Federal Information Systems and Organizations», що містить опис

засобів реалізації вимог ІБ і рекомендації щодо їх застосування.

Ще одним відомим у фахівців документом є NIST 800-30, що визначає підходи до організації діяльності з управління ризиками ІБ. Цей документ часто розглядають разом з ISO/IEC 27005:2018 «Information technology – Security techniques – Information security risk management», який присвячений управлінню ризиками ІБ.

8.2. Стандарти щодо управління ІБ

Найбільш поширеним і загальновизнаним збірником рекомендацій у сфері захисту інформації є стандарт ISO/IEC 27001 «Information technology – Security techniques – Information security management systems – Requirements». Стандарт ISO / IEC 27001 вважається найбільш концептуальним і комплексним. Його історія почалася в 80-х роках минулого століття, коли Центр комп'ютерної безпеки Департаменту торгівлі і промисловості Великобританії опублікував рекомендації DTI CCSC User's Code of Practice [8].

У 1993 році документ був доопрацьований та опублікований Британським інститутом стандартів (British Standards Institute, BSI) під назвою «Code of Practice for Information Security Management» [9]. Результатом подальшого доопрацювання документа BSI став виданий у 1995 році британський національний стандарт BS 7799: 1995, що містив актуалізований перелік рекомендованих для застосування в організаціях заходів захисту інформації.

Однак вибір оптимальних для конкретної організації заходів захисту інформації залишався за рамками стандарту. Для вирішення цієї проблеми в 1998 році BSI розробив стандарт-додаток –

BS 7799. Part 2: 1998. Саме його можна вважати прямим попередником стандарту ISO / IEC 27001.

Важливою подією в подальшій історії розвитку стандартів BS 7799 стало їх визнання з боку Міжнародної організації зі стандартизації (International Organization for Standardization, ISO) і Міжнародної електротехнічної комісії (International Electrotechnical Commission, IEC). У 2000 році технічним комітетом, створеним під егідою цих організацій, був прийнятий стандарт ISO/IEC 17799:2000, що є розвитком стандарту BS 7799-1. У 2005 році аналогічну процедуру пройшов стандарт BS 7799. Part 2: 1998, який одержав назву ISO/IEC 27001. З цього часу всі міжнародні стандарти менеджменту інформаційної безпеки, що випускаються під патронажем ISO/IEC, входять до серії 27000. Так, у 2007 році оновлена версія стандарту ISO/IEC 17799:2000 одержала назву ISO/IEC 27002 (рис. 8.1). Стандарти серії можна поділити на кілька груп (рис. 8.2).

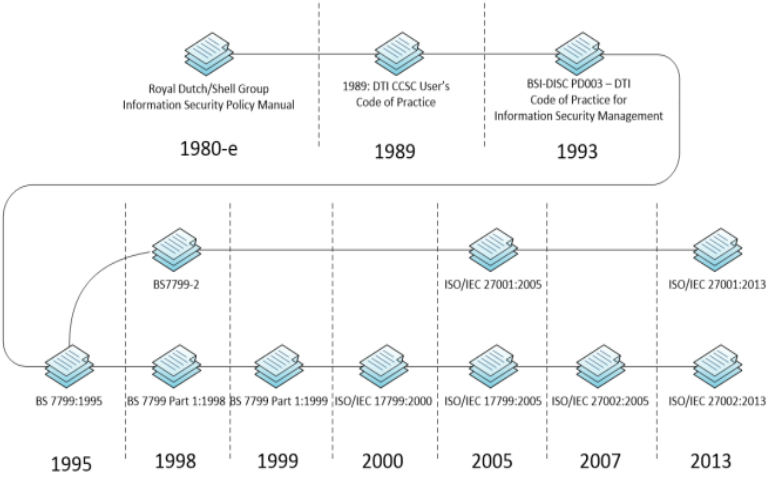


Рисунок 8.1 – Хронологія розвитку стандарту ISO/IEC 27001 [8]

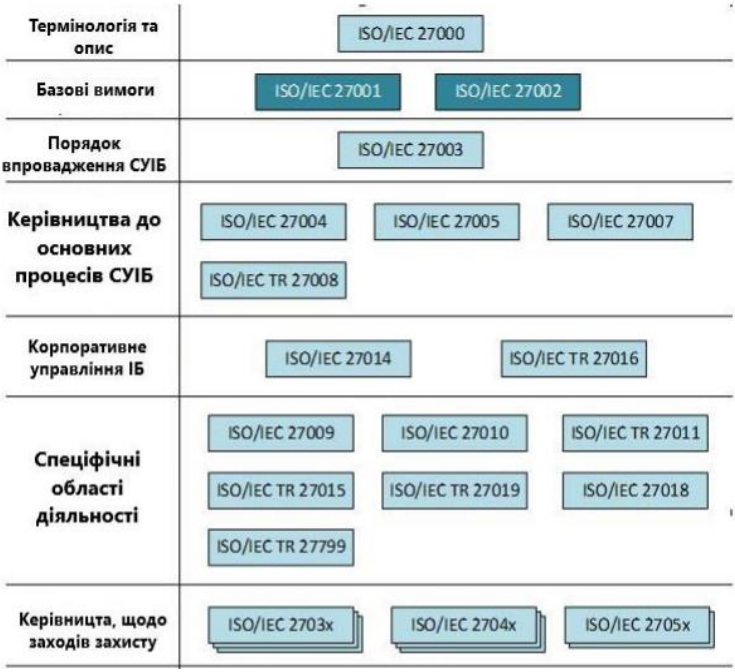


Рисунок 8.2 – Стандарти серії [8]

Базові стандарти ISO/IEC 27001 та ISO/IEC 27002 були доповнені такими документами:

– стандартом ISO/IEC 27000, що описує термінологію і загальний підхід усієї серії стандартів;

– стандартом ISO/IEC 27003 до вказівок щодо порядку впровадження СУІБ;

– стандартами з окремих процесів СУІБ: вимірювання ефективності, ризик-менеджменту, аудиту;

– стандартами за напрямками стратегічного управління ІБ та економікою СУІБ;

– стандартами в специфічних сферах діяльності: телекомунікаційні послуги; фінансові операції; оброблення персональних даних у хмарних сервісах;

паливно-енергетичному комплексі, спільнотах інформаційного обміну, організаціях охорони здоров'я;

– детальними вимогами до заходів захисту інформації, зокрема з управління інцидентами, мережевої безпеки; посібниками з інтеграції СУІБ із системами ІТ-менеджменту (ISO 20000) і системами забезпечення безперервності діяльності (зокрема ISO 22301);

– стандартами ISO/IEC 27006 та ISO/IEC 27021, що описують вимоги до експертів та аудиторів СУІБ.

ISO/IEC 27001 сумісний з іншими стандартами систем менеджменту якості, такими як ISO 9001, ISO 14000, ISO 31000, ISO/IEC 38500, ISO/IEC 20000, ISO/IEC 22301 тощо. Це дозволяє використовувати єдиний підхід і принципи, загальну термінологію.

Перелік стандартів серії ISO/IEC 27000 містить близько 60 найменувань – від стандарту ISO/IEC 27001 до стандарту ISO/IEC 27799. Вони містять вимоги до СУІБ (ISO/IEC 27001) і вимоги до органів сертифікації (ISO/IEC 27006), які здійснюють сертифікацію на відповідність ISO/IEC 27001, а також додаткові вимоги, пов'язані з впровадженням СУІБ в конкретних галузях (ISO/IEC 27009).

Інші стандарти містять рекомендації з різних аспектів упровадження СУІБ. Вони регламентують загальний процес, а також рекомендації для конкретних галузей. Класифікація для основних стандартів має вигляд:

1) загальні принципи, термінологія: 27000;

2) установлення вимог: 27001, 27006;

3) загальні рекомендації: 27002, 27003, 27004, 27005, 27007;

4) рекомендації для спеціальних сфер: 27011.

Переваги застосування Міжнародних стандартів ISO 27000: забезпечення безперервності бізнес-процесів,

забезпечення комплексного контролю рівня захисту інформації, мінімізація ризиків, забезпечення цілісності, конфіденційності та доступності критичних інформаційних ресурсів інформаційно-комунікаційних систем (ІКС) та мереж, зниження витрат на інформаційну безпеку.

8.3. Використання стандартів для проєктування та оцінювання комплексних систем захисту інформації

Загальні критерії оцінювання безпеки інформаційних технологій

Упродовж останніх років у законодавстві України відбулися зміни, пов'язані із захистом інформації в ІКС та кіберзахистом об'єктів критичної інфраструктури. Так, згідно із Законом України [18] було дозволено використовувати міжнародний підхід у сфері забезпечення захисту інформації. Згідно із Законом України [19] єдиним методом кіберзахисту об'єктів критичної інфраструктури та систем, де обробляється державна таємниця і службова інформація, є побудова комплексних систем захисту інформації (КСЗІ).

Найбільш поширеним варіантом оцінювання якості КСЗІ для складних ІКС є застосування міжнародного стандарту ISO 15408. Положення зазначеного стандарту використовують для оцінювання захищеності інформаційної системи з точки зору повноти реалізованих у ній функцій безпеки і надійності реалізації цих функцій. Стандарт є відкритим, що дозволяє урядам різних держав використовувати цей стандарт, доповнювати його та нормативно закріплювати його вимоги [3].

Стандарт складається з декількох частин, що визначають Загальні критерії оцінювання безпеки інформаційних технологій:

– Вступу та загальної моделі (Introduction and general model), що містить єдині критерії оцінювання безпеки ІТ-систем на програмно-апаратному рівні та визначає повний перелік об'єктів аналізу і вимог до них, не загострюючи уваги на методах створення, управління та оцінюванні системи безпеки;

– функціональних компонентів безпеки (Security functional components model), що містять вимоги до функціональності засобів захисту, які можуть бути використані під час аналізу захищеності для оцінювання повноти реалізованих функцій безпеки;

– компонентів довіри до безпеки (Security assurance components), що містять обґрунтування загроз, політик і вимог.

Стандарт визначає компоненти довіри до безпеки, каталогізує набори компонентів і класів довіри, містить оцінні рівні довіри, критерії оцінювання профілів захищеності і завдань безпеки.

На базі цього стандарту було розроблене Положення про державну експертизу в сфері технічного захисту інформації [20].

Положення встановлює загальні вимоги до процесу проведення державної експертизи, розподілення обов'язків між суб'єктами взаємовідносин, терміни дії дозвільних документів тощо.

У праці [21] зазначено, що мають місце концептуальні проблеми щодо оцінювання захисту інформації в ІКС (зокрема, й кіберзахисту критичних систем), підключених до мережі «Інтернет». Це пов'язано з тим, що немає підходів, які б дозволили поєднати тестування на проникнення з існуючою українською нормативною базою.

Тому була запропонована модель оцінювання захищеності ІКС державних органів із використанням

методів тестування на проникнення та апарату нечіткої логіки для одержання висновку щодо стану захищеності цільової інформаційної, комунікаційної системи [21].

Модель оцінювання захищеності ІКС

У нормативному документі НД ТЗІ 2.5-004-99 [22], визначений перелік стандартних профілів захищеності (ПЗ). Ці профілі визначають необхідні рівні послуг безпеки для забезпечення захисту інформації.

Проведений аналіз нормативного документа НД ТЗІ 2.5-004-99 дав можливість описати предметну область завдання ПЗ структурною формулою:

$$P = < X, Y, V, B, E >, \quad (1)$$

де X – множина критеріїв конфіденційності;

Y – множина критеріїв цілісності;

V – множина критеріїв доступності;

B – множина критеріїв спостережуваності;

E – інтегральний показник відповідності ПЗ вимогам.

У цьому разі

$$X = < \{x_i, Valx_i\} \,|\, i \in (\overline{1,5}) >, \quad (2)$$

де x_i – множина критеріїв оцінювання конфіденційності;

$Valx_i$ – допустимі значення критеріїв оцінювання конфіденційності.

Так, наприклад, згідно з НД ТЗІ 2.5-004-99 множина критеріїв конфіденційності X складається з таких елементів: x_1 – довірча конфіденційність, x_2 – адміністративна конфіденційність, x_3 – повторне використання об’єктів, x_4 – аналіз прихованих каналів, x_5 – конфіденційність при обміні.

Крім того,

$$Y = < \{y_i, Valy_i\} \,|\, i \in (\overline{1,4}) >, \quad (3)$$

де y_i – множина критеріїв оцінювання цілісності (y_1 – довірча цілісність, y_2 – адміністративна цілісність, y_3 – відкат, y_4 – цілісність при обміні);

$Valy_i$ – допустимі значення критеріїв оцінки цілісності;

$$V = < \{ v_i, Valv_i \} \Big| i \in \overline{(1,4)} > , \qquad (4)$$

де v_i – множина критеріїв оцінки доступності (v_1 – використання ресурсів, v_2 – стійкість до відмов, v_3 – гаряча заміна, v_4 – відновлення);

$Valv_i$ – допустимі значення критеріїв оцінювання доступності;

$$B = < \{ b_i, Valb_i \} \Big| i \in \overline{(1,9)} > , \qquad (5)$$

де b_i – множина критеріїв оцінювання спостережуваності (b_1 – реєстрація, b_2 – ідентифікація, b_3 – достовірний канал, b_4 – розподіл обов'язків, b_5 – цілісність комплекту засобів захисту, b_6 – самотестування, b_7 – ідентифікація при обміні, b_8 – автентифікація відправника, b_9 – автентифікація отримувача);

$Valb_i$ – допустимі значення критеріїв оцінювання спостережуваності.

Критерій

$$E \in \{ e_1, e_2, e_3 \},$$

де e1 = «не відповідає», e2 = «частково відповідає», e3 = «відповідає», визначає інтегральну оцінку захищеності.

У кожному конкретному випадку критерії відбирають на підставі аналізу моделі загроз, моделі порушника, політики безпеки та зазначають у технічному завданні на створення КСЗІ.

Оскільки критерії мають ієрархічну структуру та під час оцінювання використовують якісну шкалу вимірювання, то були застосовані методи логічного виведення для нечітких ієрархічних систем [23, 24]. У

рамках цих методів загальна схема розв'язування задачі оцінювання складається з таких дій:

а) перевірка необхідних умов;

б) тестування на проникнення;

в) оцінювання локальних критеріїв за принципом термометра [23];

г) процедура нечіткого логічного виведення [24];

д) підбиття підсумків оцінювання.

Ієрархію критеріїв оцінювання зображено на рисунку 8.3 у вигляді дерева логічного виведення. Цій ієрархії відповідає система співвідношень

$$E = f_E(X, Y, V, B), \tag{6}$$

$$X = f_X(x_1, x_2, x_3, x_4, x_5), \tag{7}$$

$$Y = f_Y(y_1, y_2, y_3, y_4), \tag{8}$$

$$V = fv\,(v_1, v_2, v_3, v_4), \tag{9}$$

$$B = f_B(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9). \tag{10}$$

Нечіткі логічні рівняння дозволяють оцінювати інтегральний показник E для фіксованих значень локальних показників. На першому кроці відбувається нечітке виведення для проміжних вершин локальних показників. На другому кроці чіткі значення цих змінних передаються в нечітку систему наступного рівня ієрархії.

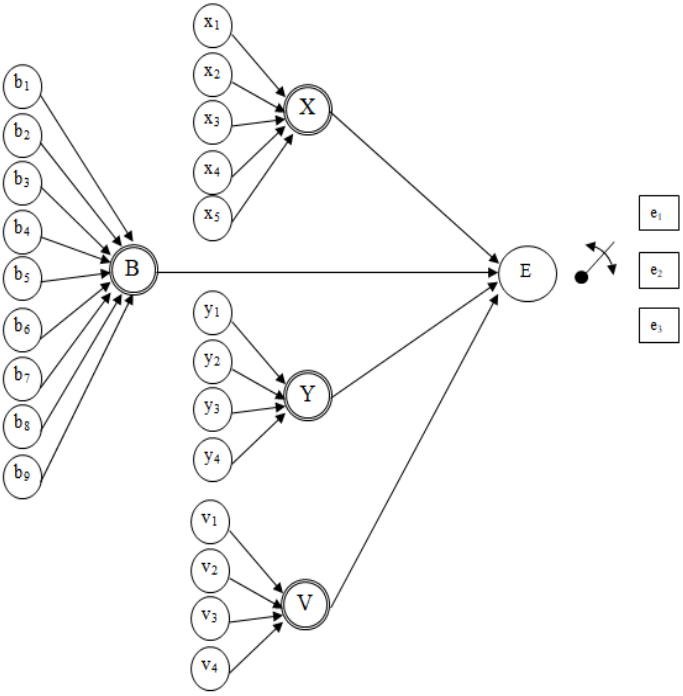


Рисунок 8.3 – Схема критеріїв оцінювання

Алгоритм нечіткого логічного виведення [23] має такий вигляд:

1. Фіксується вектор значень вхідних змінних.

2. Визначається значення функцій належності термів-оцінок вхідних змінних.

3. Обчислюються функції належності термів-оцінок вихідної величини, що відповідає вектору значень вхідних змінних.

4. Визначається оцінка, функція належності якої максимальна.

Лінгвістичні змінні x_i, y_i, v_i, b_i оцінюються нечіткими термами: Нв – не відповідає, Чв – частково відповідає, Вв

– відповідає. За співвідношеннями (6) – (10) формуються матриці знань. Умовне висловлювання в рядках матриці формується з нечітких значень вхідних та вихідних змінних.

Практична значущість розробленої моделі полягає в тому, що вона дозволяє удосконалити процес оцінювання профіля захищеності. Подальші напрямки дослідження будуть спрямовані на дослідження можливості інтеграції алгоритмів навчання до розробленої моделі.

Висновок

Таким чином, комплексне застосування системи стандартів забезпечує контроль рівня захисту інформації, сприяє мінімізації ризиків та забезпеченню цілісності, конфіденційності та доступності критичних інформаційних ресурсів ІКС.

СПИСОК ЛІТЕРАТУРИ

1. International Organization for Standardization [Electronic resource]. – Access mode : https://www.iso.org.

2. Popular standards: ISO/IEC 27001 INFORMATION SECURITY MANAGEMENT [Electronic resource]. – Access mode : https://www.iso.org/isoiec-27001-information-security.html.

3. ISO/IEC 15408-1:2009. Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model [Electronic resource]. – Access mode : https://www.iso.org/standard/50341.html.

4. COBIT 2019 Framework. Introduction and Methodology [Electronic resource]. – Access mode : https://community.mis.temple.edu/mis5203sec003spring2020/files/2019/01/COBIT-2019-Framework-Introduction-and-Methodology_res_eng_1118.pdf.

5. IT Infrastructure Library (ITIL) [Electronic resource]. – Access mode : https://www.ibm.com/cloud/learn/it-infrastructure-library.

6. Framework for Improving Critical Infrastructure Cybersecurity https [Electronic resource]. – Access mode : //nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf.

7. CYBERSECURITY FRAMEWORK [Electronic resource]. – Access mode : https://www.nist.gov/cyberframework.

8. Павленко В. В. Методика проведення комплексного аудиту системи управління інформаційної безпеки : магістерська праця / В. В. Павленко. – Київ, 2018. – 93 с.

9. BSI [Electronic resource]. – Access mode : https://www.bsigroup.com/.

10. ISO/IEC 27000:2018 / Information technology – Security techniques – Information security management systems – Overview and vocabulary. – 2018. – 27 с.

11. ISO/IEC 27001:2013 / Information technology – Security techniques – Information security management systems – Requirements. – 2013. – 23 с.

12. ISO/IEC 27006:2015 / Information technology – Security techniques – Requirements for bodies providing audit and certification of information security management systems – 2015. – 35 с.

13. ISO/IEC 27002:2013 / Information technology – Security techniques – Code of practice for information security management. – 2013.

14. ISO/IEC 27003:2017 / Information technology – Security techniques – Information security management systems – Guidance. – 2017. – 45 с.

15. ISO/IEC 27004:2016 / Information technology – Security techniques – Information security management –

Monitoring, measurement, analysis and evaluation. – 2016. – 58 с.

16. ISO/IEC 27005:2018 / Information technology – Security techniques – Information security risk management. – 2018. – 56 с.

17. ISO/IEC 27011:2016 / Information technology – Security techniques – Code of practice for Information security controls based on ISO/IEC 27002 for telecommunications organizations. – 2016. – 31 с.

18. Про захист інформації в інформаційно-телекомунікаційних системах [Електронний ресурс] : Закон України від 05.07.1994 р. № 80/94-ВР. – Режим доступу : https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80#Text. – 26.11.2020 р.

19. Про затвердження Загальних вимог до кіберзахисту об'єктів критичної інфраструктури [Електронний ресурс] : Постанова Кабінету Міністрів України від 19.06.2019 р. № 518. – Режим доступу : https://zakon.rada.gov.ua/laws/show/518-2019-%D0%BF#Text – 26.11.2020 р.

20. Про затвердження Положення про державну експертизу в сфері технічного захисту інформації [Електронний ресурс] : Наказ від 16.05.2007 р. № 93. – Режим доступу : https://zakon.rada.gov.ua/laws/show/z0820-07#Text. – 26.11.2020 р.

21. Нечітка ієрархічна оцінка якості комплексних систем захисту інформації / І. В. Шелехов, Н. Л. Барченко, В. В. Кальченко, В. К. Ободяк // Радіоелектронні і комп'ютерні системи. – 2020. – № 4 (96). – С. 106–115.

22. НД ТЗІ 2.5-004-99. Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу [Електронний ресурс] – Режим доступу : http://www.dut.edu.ua/ua/lib/2/category/925/view/1032.

23. Organizational Approach to the Ergonomic Examination of E-Learning Modules/ E. Lavrov, O. Kupenko, T. Lavryk, N. Barchenko // Informatics in education. – 2013. – No.12 (1). – P. 107–124. – DOI: 10.15388/infedu.013.08.

24. Ротштейн О. Моделювання та оптимізація надійності багатовимірних алгоритмічних процесів / О. П. Ротштейн, С. Д Штовба, О. М Козачко. – Вінниця : УНІВЕРСУМ-Вінниця, 2007. – 212 с.

РОЗДІЛ 9
ОРГАНІЗАЦІЯ ЗМІШАНОГО НАВЧАННЯ ДИСЦИПЛІН СПЕЦІАЛЬНОСТІ «КІБЕРБЕЗПЕКА»

О. А. Шовкопляс

9.1. Інноваційність екосистеми онлайн-навчання Сумського державного університету

Глобальною тенденцією університетів світу є розширення своїх освітніх послуг із навчання впродовж усього життя. Одним із важливих питань, із яким сьогодні стикаються сучасні заклади вищої освіти, є трансформований попит у сучасних студентів. Молодь бажає будувати свою індивідуальну траєкторію навчання, обираючи як академічні, так і неакадемічні курси; як в автономному режимі, так і в режимі онлайн.

Особливої актуальності це питання набуло в ІТ-сфері. Фахівці з інформаційної та кібернетичної безпеки сьогодні мають надзвичайний попит.

Основним завданням тут є забезпечення повної інтеграції між різними освітніми програмами, формами навчання і результатами, одержаними студентами на різних етапах навчання. Поглиблення процесів інтеграції повинне відбуватися на всіх рівнях: технологічному, контентному, управлінському, що й складає університетську екосистему онлайн-навчання.

І тут виявляється такий проблемний досвід: для багатьох університетів вартість розроблення електронного середовища для забезпечення навчально-наукової діяльності з нуля є непідйомною, особливо у країнах із недостатнім рівнем соціально-економічного розвитку.

Інформаційний простір Сумського державного університету – сучасна платформа, що об’єднує традиційні уявлення про комунікації та новітні віртуальні

можливості. Користувачами екосистеми онлайн-навчання СумДУ є й нинішні, й майбутні студенти українських ЗВО, а також їх випускники. Навчальні ресурси СумДУ допомагають викладачам університету в їх повсякденній практиці та істотно заощаджують університетські бюджети. Найближчою перспективою є поширення напрацювань СумДУ в освітню діяльність інших навчальних закладів України чи інших країн з аналогічною моделлю фінансування освіти на рівні держави.

Із 2002 року функціонує й розвивається в університеті потужна система дистанційного навчання. За цей період напрацьовані інноваційні педагогічні рішення в системі електронного навчання, проведена практична апробація різних методичних та організаційних моделей електронного навчання, розроблені понад 1 000 онлайн-курсів, власні платформи, середовища й онлайн-ресурси, розгалужена мережа центрів для забезпечення всіх видів онлайн-навчання. Починаючи з 2009 року застосування технологій електронного навчання активно виходить за рамки дистанційної форми навчання. Потужна розбудова *e*-learning почалася в університеті з 2011 року з прийняттям «Концепції розбудови єдиного освітнього середовища *e*-learning у СумДУ». Багатофункціональна система *e*-learning забезпечує навчально-методичну та навчально-організаційну діяльність за всіма формами і напрямами навчання, зокрема, надає доступ кожному студентові до відкритих навчальних матеріалів у зручний для нього час із будь-якого місця, забезпечує його індивідуальну освітню траєкторію, сприяє розвитку системи додаткової освіти, підтримує широке впровадження принципів академічної мобільності студентів тощо [1]. У 2016 році починається пілотне впровадження змішаного навчання в освітню діяльність

університету, формування відповідного освітнього середовища, інтеграція змішаного навчання з освітніми програмами.

Єдине освітнє середовище *e*-learning СумДУ (рис. 9.1) продовжує розбудовуватися. Більшість робіт зі створення цілісного комплексу програмно-технічних засобів різного навчального призначення виконується силами власних ІТ-підрозділів [2].

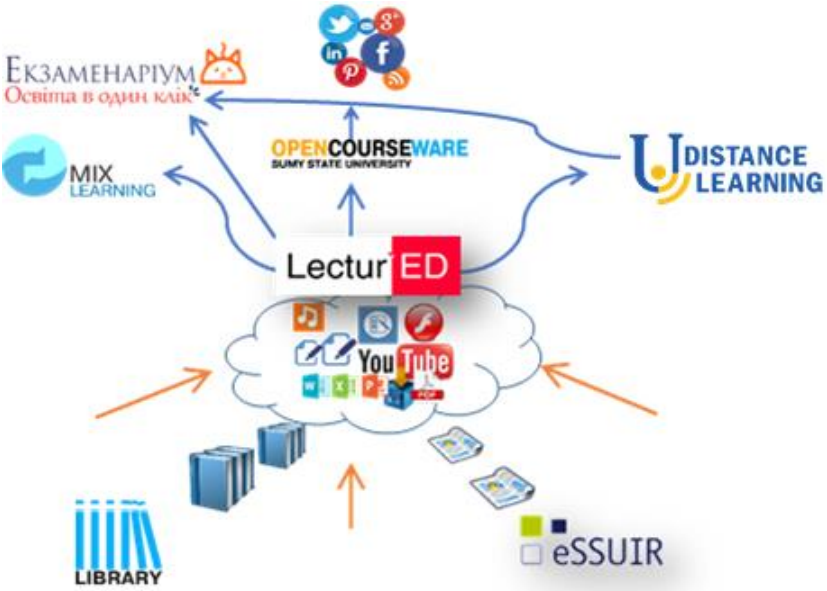


Рисунок 9.1 – Екосистема навчальних ресурсів СумДУ

На сьогодні система *e*-learning складається з таких програмно-інформаційних середовищ [3]:

– автоматизованої системи дистанційного навчання Salamstein https://dl.sumdu.edu.ua/;

– конструктора навчально-методичних матеріалів Lectur.ED https://elearning.sumdu.edu.ua/;

– відкритого освітнього ресурсу OCW https://ocw.sumdu.edu.ua/;

– платформи відкритих онлайн-курсів Екзаменаріум https://examenarium.sumdu.edu.ua/;

– платформи змішаного навчання MIX https://mix.sumdu.edu.ua/;

– електронного каталогу бібліотеки https://library.sumdu.edu.ua/;

– інституційного репозитарію https://essuir.sumdu.edu.ua/.

9.2. Міжпредметні зв'язки в системі професійної підготовки фахівців із кібербезпеки

Інформаційне суспільство розвивається швидкими темпами й потребує комплексного підходу до захисту інформації. Тому підготовка кваліфікованих фахівців із кібернетичної безпеки останнім часом набула гострої актуальності [4]. Набуття студентами теоретичних і практичних навичок під час вивчення дисциплін професійної підготовки можливе лише за умови попереднього здобуття якісної базової математичної освіти.

Міжпредметні зв'язки забезпечують принцип систематичності в процесі навчання і знаходять відображення в узгодженості навчальних програм. Розуміння міжпредметних зв'язків дозволяє студентові скласти чітке уявлення про роль та місце конкретної дисципліни в системі підготовки.

У таблиці 9.1 продемонстрована інтеграція математичної та фахової підготовки бакалаврів спеціальності 125 «Кібербезпека» на прикладі дисципліни «Математичні методи дослідження операцій» (ММДО). Вона належить до циклу дисциплін професійної підготовки, її вивчають студенти спеціальності

«Кібербезпека» на другому курсі. Упродовж вивчення дисципліни студенти опановують базові принципи моделювання систем і процесів, навчаються будувати математичні моделі задач інформаційної та кібернетичної безпеки, застосовувати методи знаходження оптимальних характеристик функціонування досліджуваних систем, а також надавати якісну інтерпретацію одержаних результатів.

Таблиця 9.1 – Основні види міжпредметних зв'язків на прикладі дисципліни ММДО

Опорні (попередні) дисципліни, поняття і категорії яких є базовими для даного курсу, використовуються в ньому	**Паралельні (супутні)** дисципліни, в яких вивчається та сама сфера реальності або професійної діяльності, використовується подібний категоріальний і методологічний апарат тощо	**Подальші (перспективні)** дисципліни, для яких курс, що вивчається, буде опорним, або курси більш спеціальної сфери вивчення
«Вища математика», «Теорія ймовірностей, ймовірнісні процеси та математична статистика», «Організація та обробка електронної інформації»	«Спеціальні розділи математики», «Алгоритми захисту інформації»	«Системи та засоби криптоаналізу»

Для математичного моделювання сучасних систем захисту інформації використовують методи символьного, графічного моделювання або їх комбінацію. Порівняння аналітичного, чисельного та інших розв'язків однієї й тієї самої задачі сприяє всебічному аналізу процесів. Моделі зазвичай є наближеним математичним описом процесів функціонування досліджуваних систем. Їх розрізняють як за характером, так і за ступенем складності. Залежно від специфіки зв'язків моделі поділяють на детерміністичні та імовірнісні (стохастичні). Як ті, так і інші зазвичай містять цільову функцію, яку потрібно оптимізувати, й деяку сукупність обмежень.

Основними розділами навчальної дисципліни ММДО є «Лінійне програмування», «Нелінійне і динамічне програмування», «Чисельні методи оптимізації», «Дискретне та стохастичне програмування».

Передумовою для вивчення ММДО є одержання студентами базових знань та набуття практичних навичок з опорних дисциплін: елементи лінійної алгебри, аналітичної геометрії, теорії розв'язання систем лінійних рівнянь і нерівностей, диференціальне та інтегральне числення («Вища математика»), основи теорії ймовірностей та математичної статистики, випадкові величини, закон великих чисел («Теорія ймовірностей, ймовірнісні процеси та математична статистика»), підготовка електронних документів засобами офісного програмного забезпечення, набуття навичок розв'язування задач оптимізації за допомогою Microsoft Excel, навичок роботи в таких математичних пакетах, як MathCad, MatLab («Організація та обробка електронної інформації») тощо. Вміння ж розробляти й аналізувати моделі складних систем потрібні для реалізації мети подальшої дисципліни «Системи та засоби криптоаналізу» – формування здатності застосовувати та модифікувати існуючі методи

криптоаналізу під час розв'язування задач захисту інформації.

Урахування взаємозв'язків між дисциплінами дозволяє викладачам створювати навчальний контент із застосуванням інформації про раніше одержані знання студентів і планувати подання матеріалу не ізольовано, а з урахуванням потреб подальших дисциплін.

9.3. Реалізація концепції змішаного навчання в Сумському державному університеті

Змішане навчання реалізується під керівництвом викладача під час безпосередньої взаємодії в аудиторії та опосередкованої роботи в онлайн-середовищі. У цьому разі застосування електронного навчання не замінює повністю аудиторних занять із викладачем, а лише розширює їх можливості шляхом упровадження сучасних засобів та технологій. Характерною рисою змішаного навчання є певний самоконтроль студентом свого часу, темпу та обсягів вивчення навчального матеріалу.

Упровадження будь-яких новітніх технологій завжди супроводжується необхідністю вирішення широкого кола проблем, як технічних, так і організаційно-методичних. Змішане навчання передбачає зовсім інший погляд на навчальний процес порівняно з традиційними формами. Поряд із викладацьким досвідом, наявністю матеріально-технічної та інформаційної бази для створення онлайн-курсу потрібна інтеграція зусиль цілого комплексу спеціалістів різних напрямків – науковців, методистів, психологів, адміністраторів, програмістів тощо.

Змішане навчання надає більше простору для забезпечення умов досягнення високих навчальних результатів, що обумовлено:

– використанням інформаційних сервісів університету;

– розміщенням на навчальних платформах якісного навчально-методичного забезпечення;

– використанням інструментів дистанційної роботи [5];

– постійною взаємодією студентів між собою і з викладачами безпосередньо та онлайн;

– цілеспрямованим опрацюванням освітніх інформаційних ресурсів.

За допомогою інформаційно-комунікаційних технологій можна по-новому подавати зміст навчального матеріалу, регулювати форми і темп навчання, що сприятиме підвищенню якості навчання.

Із огляду на можливі типи взаємодії суб'єктів навчального процесу (студент – викладач, студент – студент, студент – контент) визначені такі категорії для складових навчальної діяльності, поєднання яких характеризує змішане навчання:

– категорія 1 – усі види навчальної діяльності, що передбачають *безпосередню* взаємодію суб'єктів між собою в аудиторії чи за її межами, а також самостійне опрацювання навчального контенту без застосування онлайн-технологій;

– категорія 2 – всі види навчальної діяльності, що передбачають *опосередковану* взаємодію суб'єктів між собою в аудиторії чи за її межами у віртуальному онлайн-середовищі, а також інтерактивну взаємодію студентів із навчальним онлайн-контентом.

Відповідно до зазначених категорій визначений перелік видів навчальної діяльності студентів, передбачених у змішаному навчанні [6].

Необхідними складовими забезпечення змішаного навчання за навчальною дисципліною є:

– методична модель, що передбачає спільне застосування навчальної діяльності за обома категоріями;

– відповідне обладнання, аудиторії та навчально-методичне забезпечення навчальної діяльності за категорією 1;

– ресурси для забезпечення навчальної діяльності за категорією 2;

– навчально-методичне забезпечення дисципліни, доступне за вебпосиланням;

– віртуальне середовище для доступу до контенту та для взаємодії суб'єктів навчального процесу.

Механізм реалізації концепції змішаного навчання як процесу передбачає наявність комфортного освітнього інформаційного середовища, і такою системою комунікацій між викладачами та студентами в університеті є власна платформа змішаного навчання MIX.

Для вивчення та поширення успішного досвіду поєднання традиційних технологій навчання з технологіями *e*-learning у навчальному процесі університет започаткував експеримент із розроблення та апробації університетської моделі змішаного навчання (Експеримент). Термін проведення Експерименту – 2017–2018 та 2018–2019 навчальні роки. Основною метою є підвищення ефективності аудиторної та самостійної роботи студентів, оптимальне поєднання класичних педагогічних підходів із технологіями *e*-learning у навчальному процесі для дисциплін різного спрямування та напрацювання нових рішень.

У рамках експериментальних дисциплін студенти висловили свою думку щодо змішаного навчання [7]. На запитання про те, яка модель навчання вам сподобалася, більше ніж 65 % студентів відповіли змішана, 18 % – традиційна, і 17 % – не змогли визначитися.

Результати фінального опитування свідчать, що запропонована модель змішаного навчання розвинула в студентів навички самостійного планування та організації діяльності, відповідальності й самостійності, сприяла поглибленню вмінь, активному залученню до освітнього процесу. Серед переваг самостійної роботи з онлайн-матеріалами (вибір кількох відповідей) студенти зазначили зручність доступу до матеріалів (70,6 %), можливість самостійно планувати час (58 %), можливість неодноразово звертатися до одного й того самого матеріалу для кращого засвоєння (55,5 %), комфортність навчання в спокійних умовах (48,7 %), зручність для навчання за індивідуальним планом (45,4 %), можливість опрацювання додаткового матеріалу з певної тематики (27%), розкриття потенціалу кожного студента відповідно до його індивідуальних особливостей (22 %), можливість особистого зростання (13,5 %). Основним недоліком студенти вважають технічні проблеми.

Сьогодні *e*-learning в університеті – це потужне освітнє середовище, яке навіть за кардинальних змін у житті людства спроможне забезпечити не лише повноцінний навчальний процес, а й його вдосконалення.

9.4. Педагогічні рішення в системі змішаного навчання

Для створення і розміщення колекцій навчальних матеріалів залежно від цільової аудиторії слухачів та мети онлайн-курсів використовують різні авторські середовища (підрозділ 9.1, рис. 9.1). Організація змішаного навчання відбувається на університетській платформі MIX. Навчальний контент розробляється згідно з Вимогами до навчальних матеріалів дистанційної форми навчання та критеріями їх оцінювання в університеті [8].

Теоретичний матеріал викладається в повнотекстових лекціях, які можна доповнювати стислим конспектом, презентаціями, відео- та аудіоматеріалами. Навчальні об'єкти для набуття практичних навичок і вмінь, а також контролю знань подані тестами, практичними завданнями, тренажерами, віртуальними лабораторними роботами, завданнями для дискусій та обговорень, завданнями для спільної роботи. Наприкінці курсу наводиться глосарій і завдання для підсумкового контролю знань.

Лекції супроводжуються прикладами розв'язування задач, ілюстративним матеріалом, ключовими термінами, бібліографічними посиланнями. Стислий конспект містить обов'язковий мінімум інформації для підсумкового контролю знань. Його можна реалізувати як окрему складову після повнотекстової лекції, або як структуровану форму, де кожний змістовний елемент матеріалу можна «розгорнути» з метою деталізації, або ж зробити налаштування для використання лише повнотекстової або стислої версії конспекту. Це дає можливість викладачеві зробити певні акценти на основних елементах курсу. Студенти краще сприймають та засвоюють матеріал, а також мають можливість більш ефективно підготуватися до підсумкового контролю знань. Ключові терміни полегшують навігацію на html-сторінці та забезпечують пошук необхідної інформації. Лекційний матеріал містить навчальний контент, передбачений змістом дисципліни. Для поглибленого вивчення студентові надається перелік основної та рекомендованої літератури.

Важливою складовою електронного навчання є *тестування*, розглядати яке можна і як засіб для забезпечення об'єктивного оцінювання результатів навчальної діяльності студента, і як інструмент для

самоперевірки. Тестову базу викладачі створюють за допомогою редактора навчальних об'єктів Studio та/або конструктора навчальних матеріалів Lectur.ED із подальшим використанням на навчальних платформах дистанційного та змішаного навчання.

Блок самоперевірки передбачає перелік питань та тестові завдання до змістовної теми / модуля курсу. Тестові завдання дозволяють студентові одержати зворотний зв'язок про рівень і повноту засвоєних ним знань. Основна мета тестів полягає насамперед у тому, щоб організувати цілеспрямоване осмислення основних теоретичних положень та їх практичне застосування, і лише потім – в оцінюванні знань. Тести формують таким чином, щоб охопити весь основний навчальний контент. Для тестування використовують відкриті тестові завдання та закриті (вибір одного або кількох правильних варіантів відповіді, заповнення пропусків, установлення відповідностей, упорядкування, тести «правильно – неправильно» тощо).

Приклади авторських розробок тестових завдань (Шендрик В. В., Парфененко Ю. В., Шовкопляс О. А.) для дисципліни ММДО продемонстровані на рисунках 9.2–9.3.

Для задачі пошуку точок екстремуму функції $F = x_1^2 + x_2^2$ за умови $x_1 + x_2 = 5$ функція Лагранжа матиме такий вигляд:

- $L(x_1, x_2, \lambda) = x_1^2 + x_2^2 + \lambda(5 - x_1 - x_2)$
- $L(x_1, x_2, \lambda) = x_1^2 + x_2^2 + \lambda(x_1 + x_2)$
- $L(x_1, x_2, \lambda) = x_1^2 + x_2^2 - 5\lambda - x_1 - x_2$
- $L(x_1, x_2, \lambda) = x_1^2 + x_2^2 + \lambda(5 - x_1 + x_2)$

Рисунок 9.2 – Приклад тестового завдання з вибором однієї правильної відповіді

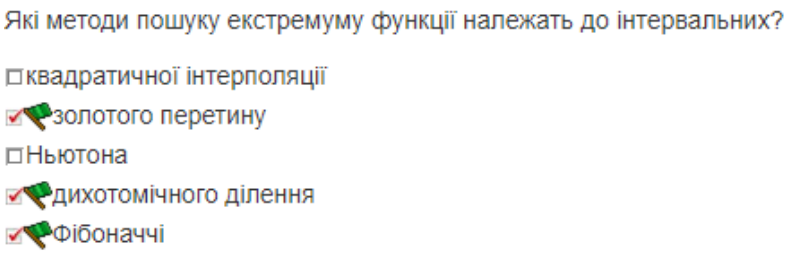


Рисунок 9.3 – Приклад тестового завдання з вибором кількох правильних відповідей

Система дозволяє налаштовувати такі параметри тестування: відеоспостереження, прохідний бал, кількість сеансів тестування, кількість спроб виправлення відповіді у кожному сеансі тестування, кількість тестових завдань для одночасного перегляду на екрані. Є можливість показувати / не показувати студентові його відповіді, а також відкривати правильні відповіді, розбивати тестові питання на блоки, виводити на екран питання випадково або в заданому порядку. Варіанти відповідей тестового завдання автоматично перемішуються. Налаштування можна зробити і для типу оцінювання, обравши відповідну модель (бали за тест з урахуванням балів за завдання, зменшення кількості балів за наступні спроби та інші).

Після опрацювання теоретичного матеріалу, роботи з тестами студент продовжує одержувати професійні знання, працюючи з навчальними об’єктами, які формують практичні навички. Студенти виконують *практичні завдання* і надсилають викладачеві на перевірку. Звіти можуть мати різний формат: стандартні електронні документи, шаблони для введення обмеженої кількості інформації, аудіовідповіді тощо.

Під *інтерактивним практичним завданням* розуміють електронний засіб навчання, що базується на імітації об’єктів і процесів реального світу та застосовується для організації набуття тими, хто

навчається, досвіду використання способів та засобів управління цими об’єктами і процесами, підготовки студентів до здійснення відповідних дій у реальних ситуаціях. Насамперед тренажери використовують для задач, які мають визначений алгоритм розв’язування. У цьому разі студент повинен здійснити певну кількість кроків, на кожному з яких він одержує інформацію про правильність виконаних дій. У тренажері зазвичай надається орієнтир на розв’язування задачі, що реалізується у вигляді вказівок на те, які дії потрібно виконувати, наводяться допоміжні запитання (рис. 9.4).

Рисунок 9.4 – Тренажер «Транспортна задача. Метод мінімального тарифу» (автор Літвіненко О. А.)

Демонстраційні версії тренажерів, відеоролики з прикладами розв’язування завдань та проблемними ситуаціями дозволяють викладачеві наочно й доступно

подати матеріал, а студентові – засвоїти його з метою подальшого застосування в професійній діяльності.

СумДУ проводить пошук педагогічних рішень із розроблення та впровадження інтерактивних практичних завдань. Практика розроблення тренажерів викладачами університету свідчить про необхідність застосування двох різних методичних підходів. Якщо розв'язування задачі передбачає виконання чіткої послідовності заданих операцій з однозначним варіантом ухвалення рішення на кожному етапі, є можливість створити абсолютно автономний у роботі тренажер, результати якого система зараховує автоматично без участі викладача.

Особливого підходу потребує розроблення сценаріїв тренажерів для вирішування проблемних ситуацій творчого характеру. В таких випадках зазвичай існує багато суб'єктивних факторів, що впливають на кінцевий результат. Оскільки однозначної відповіді бути не може, аналіз та оцінювання розумової діяльності студента можна здійснити лише за участі викладача. Дослідницька група розробляє напівавтоматичні тренажери, де лише частина етапів зараховуються автоматично, а остаточну перевірку і висновок щодо рівня виконання завдання робить викладач.

Дискусійним положенням є принципи роботи студентів із тренажерами. Насамперед для формування навичок розв'язування задач необхідно, щоб виконувалася вимога повторюваності, тобто неодноразового виконання заданої вправи. У цьому разі тренажер використовується як навчальний засіб. Також доцільно враховувати зроблені помилки для висновку про досягнення певного рівня успішності. Можна вважати, що студент опанував опрацьований вид роботи, якщо кількість допущених помилок не перевищує деякого наперед заданого рівня. З іншого боку, нараховуючи результуючі бали, можна

поставити умову – максимальна кількість балів за даний вид роботи нараховується лише після певної кількості успішних спроб, або ж поступово за кожною спробою (мінімально достатню кількість спроб визначає викладач).

Одним з аспектів застосування тренажерів в навчальному процесі є врахування міжпредметних зв'язків, якщо раніше вивчений матеріал може бути корисним для розв'язування задач інших дисциплін. В опорній дисципліні тренажер є навчальним або контролювальним засобом, а в паралельній або перспективній – обчислювальним засобом для розв'язування певних професійних завдань. Результати дослідження міжпредметних зв'язків наведені в праці [9]. На рисунку 9.5 проілюстровано інтеграцію поглядів на функції тренажерів.

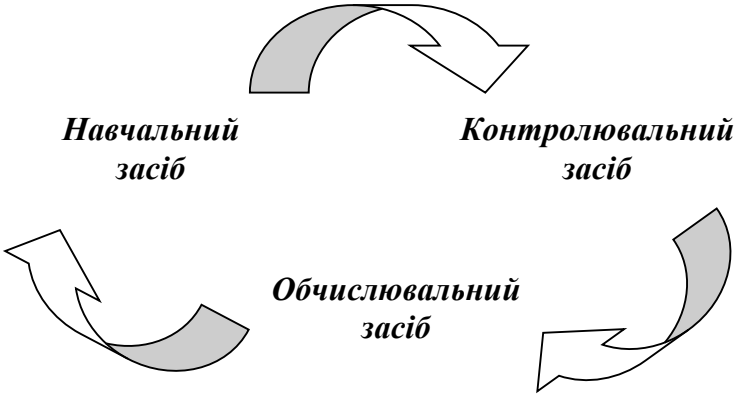


Рисунок 9.5 – Функції тренажерів

Використання такого коммунікативного методу навчання, як *дискусія*, а також різних форм *спільної роботи* сприяють розвитку професійного мислення,

зокрема, формуванню базових навичок медіаграмотності, інформаційної безпеки та кібергігієни.

Платформа змішаного навчання MIX передбачає організацію *взаємоперевірки робіт*. Цей вид навчальної діяльності значно посилює оцінювання самостійної роботи студентів. Організація взаємоперевірки в масових відкритих онлайн-курсах (МВОК) є обов'язковою умовою використання практичних завдань, бо вони перевіряються вручну і потребують колосальних витрат часу. На відміну від МВОК взаємоперевірка в онлайн-курсах для академічних груп має інші першочергові завдання [10].

Виважені критерії та правильно організована взаємоперевірка активізує діяльність студентів, сприяє розвитку вмінь аналізувати, порівнювати, узагальнювати, підвищуючи ефективність онлайн-навчання в цілому. Здійснення взаємоперевірки поглиблює знання студентів: щоб проаналізувати роботу одногрупника, студент повинен спочатку розібратися з теоретичним матеріалом, виконати своє завдання. А потім, як зазначають самі студенти, перевіряючи інші роботи, можна побачити й альтернативні варіанти виконання одного й того самого завдання, і знайти не лише чужі, а й свої помилки, і таким чином переосмислити власні дії (рис. 9.6). Проведене опитування в рамках вивчення дисципліни ММДО свідчить, що не всім студентам спочатку вдавалося бути об'єктивними, але в кожній наступній спробі суб'єктивна складова зменшувалася, з'являлося більше відповідальності. Надалі планується використовувати взаємоперевірку і як форму контролю засвоєння знань, і як корисний інструмент для публічного захисту курсових робіт.

На рисунку 9.7 наведений фрагмент сторінки навчального курсу з результатами взаємоперевірки: у першому стовпчику можна бачити, який студент виконав

цю роботу, в другому – оцінки за перевірені ним роботи інших студентів із поясненнями. У третій графі викладач оцінює якість взаємоперевірки та змістовність коментарів.

Рисунок 9.6 – Результати опитування студентів

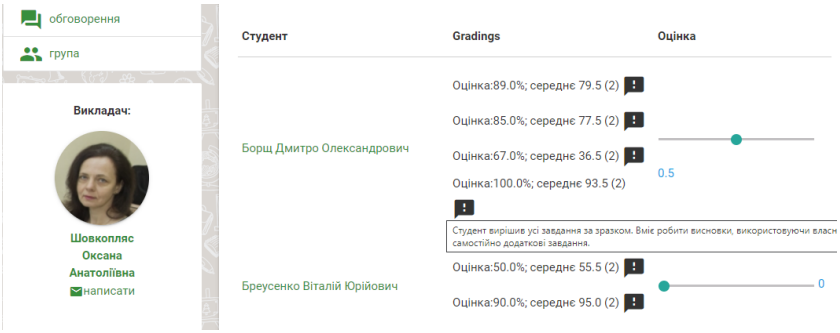


Рисунок 9.7 – Результати за завданням «Взаємоперевірка лабораторної роботи»

Навчальні бали за всі види активності студентів у курсі заносяться до підсумкової таблиці (рис. 9.8).

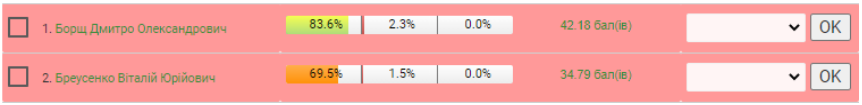


Рисунок 9.8 – Сумативна оцінка навчальної діяльності

З уведенням в Україні з 12 березня 2020 року карантину з протидії поширенню коронавірусної інфекції викладачі СумДУ, як і всі освітяни, прийняли нові виклики. Найбільшу результативність організації навчального процесу забезпечила власно розроблена платформа змішаного навчання MIX. У період карантину кількість дисциплін на навчальній платформі збільшилася в рази. Використання навчальної платформи забезпечило студентам послідовність і безперервність навчання, наочність та доступність, практичне застосування, підтримку.

Онлайн-заняття проводили за розкладом із використанням таких сервісів для організації відеоконференцій:

– Google Meet (https://meet.google.com/);
– Zoom (https://zoom.us/);
– Microsoft Teams (http://365.sumdu.edu.ua/).

Викладачі розробляли і розміщували навчально-методичні матеріали, створювали віртуальні класи, приєднували до них студентські групи, мали широкі можливості з організації повної взаємодії зі студентами. Більшої оперативності щодо підтримки студентів можна досягнути поєднанням різних засобів комунікації, наприклад, навчальної платформи MIX і месенджерів – Telegram, Viber тощо. Об’єднання студентів в одну групу дає можливість викладачеві миттєво реагувати на запити студентів, створювати дискусії та обговорення, відповідати на поширені питання, аналізувати – проблемні, надавати посилання на онлайн-заняття, повідомляти оперативну інформацію та виконувати безліч

інших корисних операцій. Постійний зв'язок зі студентами дозволяє викладачеві вибудовувати довірливі відносини і бути неформальним лідером [11].

Узагальнені рекомендації МОН України щодо впровадження змішаного навчання в закладах фахової передвищої та вищої освіти [12] допоможуть викладачам оптимально поєднати методики й технології для викладання своїх навчальних дисциплін, окреслити очікувані результати навчання та програмні.

Висновки

У роботі пропонується визначити як основну ознаку змішаного навчання його реалізацію під керівництвом викладача в умовах, коли навчальна діяльність студентів поєднує безпосереднє спілкування в аудиторії із самостійним опрацюванням матеріалів, зокрема, в опосередкованому віртуальному онлайн-середовищі. Використання інформаційних технологій у навчальному процесі дає більш широкі можливості для забезпечення зворотного зв'язку і підвищення ефективності навчання.

Особливо уважного ставлення потребує організація змішаного навчання за спеціальністю 125 «Кібербезпека». Установлення міжпредметних зв'язків навчальних дисциплін кафедри комп'ютерних наук дозволяє студентам органічно поєднувати базову й професійну освіту, усвідомлювати важливість професії кіберзахисника.

Навчання, починаючись у 2016 році як змішане, з 2020 року у зв'язку з карантинними обмеженнями продовжилося в університеті як дистанційне. Авторські навчальні платформи, постійна організаційна, методична та технологічна підтримка викладачів, онлайн-курси як збалансовані колекції різноманітних навчальних об'єктів, правильно організовані взаємодії «викладач – студент»,

«студент – студент», «студент – контент» дозволили не лише гідно продовжувати навчальний процес в умовах карантину, а й удосконалювати його. Набутий унікальний досвід сприятиме подальшій розбудові університетської моделі змішаного навчання, підвищенню рівня ефективності безпосередньої та опосередкованої взаємодії між суб'єктами навчального процесу.

СПИСОК ЛІТЕРАТУРИ

1. Концепція розбудови єдиного освітнього середовища *e*-learning в СумДУ [Електронний ресурс] : Протокол вченої ради // Реєстр нормативної бази Сумського державного університету. – Режим доступу : https://normative.sumdu.edu.ua/?task=getfile&tmpl=component&id=800c72f4-f364-e411-afcd-001a4be6d04a&kind=1.

2. Застосування електронного навчання для підготовки й підвищення кваліфікації фахівців ІТ-галузі у вищих навчальних закладах : монографія / за заг. ред. А. В. Васильєва. – Суми : СумДУ, 2013. – 138 с.

3. ІТ-забезпечення діяльності інноваційного університету: досвід українського вишу : монографія / за заг. ред. А. В. Васильєва. – Суми : СумДУ, 2016. – 173 с.

4. Доктрина інформаційної безпеки України : Указ Президента України від 25.02.2017 р. № 47/2017 // Офіційний вісник Президента України. – 2017. – № 5. – С. 15. – Ст. 102.

5. Tech against Coronavirus [Electronic resourse]. – Access mode : https://techagainstcoronavirus.com/.

6. Про змішане навчання з окремих дисциплін [Електронний ресурс] : Наказ ректора № 0537-І від 19.07.2018 р. // Реєстр нормативної бази Сумського державного університету. – Режим доступу : https://normative.sumdu.edu.ua/?task=getfile&tmpl=component&id=de2eef67-0c8c-e811-8607-001a4be6d04a&kind=1.

7. Особливості впровадження змішаного навчання у Сумському державному університеті / О. А. Шовкопляс, Ю. О. Зубань, О. О. Базиль та ін. // Матеріали Міжнародної науково-практичної конференції «Змішане навчання – інновація XXI сторіччя» (Харків, 29–30 листопада 2018 р.). – Харків : НТУ «ХПИ», 2018. – С. 114–120.

8. Вимоги до навчально-методичних матеріалів дистанційної форми навчання та критерії їх оцінювання [Електронний ресурс] // Реєстр нормативної бази Сумського державного університету. – Режим доступу : https://normative.sumdu.edu.ua/?task=getfile&tmpl=component&id=ee2d07c5-f464-e411-afcd-001a4be6d04a&kind=1.

9. Моделювання навчального процесу вивчення економіко-математичних дисциплін з використанням комп'ютерних технологій : звіт про НДР (остаточний) / кер. О. А. Шовкопляс. – Суми : СумДУ, 2015. – 53 с.

10. Модель організації змішаного навчання у вищому навчальному закладі : звіт про НДР (остаточний) / кер. О. А. Шовкопляс. – Суми : СумДУ, 2020. – 91 с.

11. Шовкопляс О. А. Забезпечення навчальної діяльності студентів Сумського державного університету у дистанційному режимі / О. А. Шовкопляс, О. О. Базиль // Екстрене дистанційне навчання в Україні : колективна монографія / за ред.: В. М. Кухаренка, В. В. Бондаренка. Харків : Вид-во КП «Міська друкарня», 2020. – С. 326–341.

12. Рекомендації МОН України щодо впровадження змішаного навчання у закладах фахової передвищої та вищої освіти [Електронний ресурс]. – Режим доступу : https://mon.gov.ua/ua/osvita/visha-osvita/rekomendacij-shodo-vprovadzhennya-zmishanogo-navchannya-u-zakladah-fahovoyi-peredvishoyi-ta-vishoyi-osviti.

РОЗДІЛ 10
ДОСЛІДЖЕННЯ ВЕБУРАЗЛИВОСТЕЙ: МЕТОДИ ВИЯВЛЕННЯ Й ЗАПОБІГАННЯ

Т. В. Лаврик, З. І. Маслова

Вступ

Нині значна частина користувачів для економії свого часу вважають кращим користуватися послугами та купувати товари онлайн, дізнавшись усю необхідну інформацію через сайти компаній. З огляду на це всі компанії, що надають послуги, створюють вебдодатки, які є обличчям кожної компанії, суттєво впливаючи на її репутацію й функціонування. У зв'язку з цим, компанія зацікавлена підтримувати свої вебдодатки на високому технологічному рівні, зокрема забезпечувати захист від кібератак.

Сучасні вебдодатки складаються з двох частин: серверної та клієнтської. Серверна частина використовує різні технології та мови програмування (.Net, PHP, Python і Ruby), а інтерфейсна частина – це клієнтська частина, що запускається у веббраузері користувача з різними мовами програмування, такими як JavaScript і CSS / HTML. Ці дві частини пов'язані через протоколи HTTP або HTTPS, асинхронний XML (AJAX) і JavaScript [1]. Типову архітектуру серверної й клієнтської частин вебдодатка наведено на рисунку 10.1.

Серед користувачів глобальної мережі «Інтернет» найпопулярнішими вебдодатками є блоги, соціальні мережі, онлайнові магазини та банки, що дуже часто стають мішенню зловмисників [2].

Будь-яке слабке місце або помилка в програмному коді чи налаштуваннях – уразливість вебдодатка, що може бути використаною зловмисниками [3].

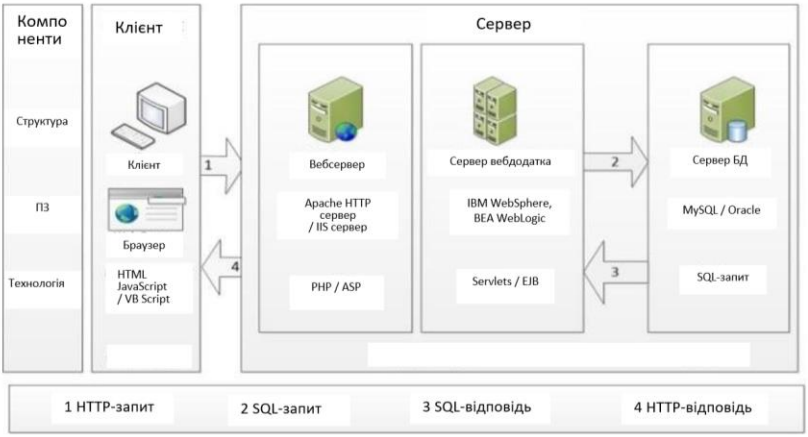


Рисунок 10.1 – Типова клієнт-серверна архітектура вебдодатка

Згідно зі статистикою, одержаною експертами компанії «Positive Technologies» щодо аналізу захищеності вебдодатків у 2019 році, частка додатків, що містять уразливості високого рівня ризику, досягла 50 % і продовжує зростати [4].

До найбільш критичних уразливостей належать міжсайтовий скриптинг (XSS), SQL-ін'єкції (SQLi) і міжсайтова підробка запитів (CSRF), унесених до списку 10 основних вебуразливостей за версією OWASP [5]. Здебільшого саме через уразливості у вебдодатках зловмисники одержують доступ до важливої конфіденційної інформації. Отже, вебдодаток потребує заходів для його безпечного функціонування.

Досвід іноземних та українських фахівців з інформаційної й кібербезпеки свідчить про необхідність застосування різних методів своєчасного виявлення та запобігання вебуразливостям. Одним із таких методів є тестування безпеки вебдодатка для запобігання й

мінімізації вразливостей. Проте для цього потрібно, щоб тестувальники мали відповідний досвід [6].

Отже, проведемо дослідження різних підходів для виявлення та запобігання вразливостям у вебдодатках.

Аналіз вебуразливостей

Відповідно до стандарту ISO / IEC 29147:2018 термін «уразливість» означає слабке місце в програмному забезпеченні, апаратному забезпеченні або онлайн-сервісі [7]. Уразливості, виявлені зловмисниками в коді вебдодатка, можуть призвести до значних втрат під час його експлуатації [4, 8]. Шкідливий код, упроваджений зловмисником за допомогою вхідних даних, поширюється в додатку через такі недоліки програмування, як неправильні перевірка вхідних даних, неправильні механізми автентифікації й авторизації, управління інформацією про сеанс та інші помилки реалізації, що порушують передбачувану функціональність програмного коду [8, 9].

Зі збільшенням кількості різних технологій за останні декілька років здійснювалися численні спроби ввести найбільш зручну й загальну класифікацію можливих уразливостей, щоб поділити їх за категоріями. Проте ті вразливості, що пов’язані з помилками програмування та можуть впливати на безпеку системи, не вдавалося однозначно класифікувати, тому що кожна з них може належати до декількох категорій або класів.

У таблиці 10.1 наведено стандарти класифікації категорій і класів вебуразливостей, що призначені допомогти фахівцям з інформаційної та кібербезпеки й розробникам програмного забезпечення виявити поширені проблеми безпеки у вебдодатках. Водночас ці

стандарти вже реалізовано в багатьох інструментах оцінювання безпеки вебдодатків.

Таблиця 10.1 – Стандарти класифікації вебуразливостей

Стандарт безпеки	Посилання на ресурс
Common Weakness Enumeration (CWE)	https://cwe.mitre.org/data/index.html
OWASP TOP 10	https://owasp.org/www-project-top-ten/
WASC Threat Classification	http://projects.webappsec.org/f/WASC-TC-v2_0.pdf

Для більш детальнішого розгляду зупинимося на окремих класах і категоріях вебуразливостей, наведених у таблиці 10.2. Їх поділено на три класи: некоректну автентифікацію, помилки під час уведення даних і некоректне управління сеансом. Також додатково виділяють чотири категорії вебуразливостей: маніпулювання запитами, ін'єкції клієнтської частини, упровадження шляху й управління сеансом.

Досить часто вебдодатки надають можливість користувачам уводити дані, тому одна з основних проблем безпеки вебдодатків полягає саме в некоректній перевірці даних, уведених користувачем. Такі дані надходять у вебдодаток через відповідні точки входу ($ _GET на мові PHP).

Зловмисники можуть використовувати програмні недоліки й помилки вебдодатка, наприклад за допомогою запитів MySQL.

Більшість атак реалізують під час уведення необхідних даних разом із метаданими, такими як And, OR [10].

Таблиця 10.2 – Класи й категорії вебуразливостей

Клас вебуразливості	**Категорія вебуразливості**	**Огляд**
Некоректна перевірка введення	*Маніпуляція запитами*	Вебуразливості, пов'язані зі структурою, що використовується для зберігання інформації
	Ін'єкції клієнтської частини	Вебуразливості, пов'язані зі зловмисним кодом, упроваджуваним на стороні клієнта, та оброблюваним на стороні сервера
	Уразливості впровадження файлів й шляхів	Вебуразливості, що маніпулюють відносним шляхом, переспрямовуючи в інше місце
Некоректне управління сеансом	*Управління сеансом*	Вебуразливості, за яких несанкціоновані вказівки передаються від клієнта, якому вебдодаток довіряє
Некоректні автентифікація та авторизація	*Логічна помилка*	Вебуразливості, якими можна маніпулювати за допомогою коду вебдодатка, зокрема його змінювання

Клас 1. Некоректна перевірка введення

Вебдодаток повинен належно перевіряти або/й дезінфікувати дані, уведені користувачем, перед їх використанням на вебсерверах. Зазвичай веброзробники застосовують методи очищення даних за допомогою фільтрації [2, 11].

1. *Маніпуляція запитами* – це вебуразливість, пов'язана зі структурами, що зберігають дані, наприклад базами даних, яка полягає в упровадженні шкідливого коду, що маніпулює запитами й змінює їх.

За допомогою вебуразливостей такого типу зловмисник легко маніпулює параметрами, уведеними користувачем. У результаті зловмисник одержує можливість змінити синтаксис запиту. Якщо перевірка уведених параметрів не здійснюється належним чином, то зловмисно інфіковані параметри потрапляють до надійного вебдодатка. Як наслідок небезпечна та ненадійна інформація потрапляє до вебдодатка й завдає шкоди його безпеці. Отже, відсутність або некоректна перевірка даних, уведених користувачем, є основною причиною вразливості ін'єкцій.

Виділяють різні типи подібних вебуразливостей, такі як SQL-, LDAP- і NoSQL-ін'єкції. Вони пов'язані з побудовою фільтрів і запитів. Найвідомішою експлуатованою вебуразливістю вважають SQL-ін'єкцію [2, 11].

2. *Ін'єкції клієнтської частини* дозволяють зловмисникові проникати до вебдодатка з клієнтської частини, виконувати шкідливий код у браузерах користувача без запиту сервера. У цій категорії є різні вразливості, такі як міжсайтовий скриптинг (Cross-site Scripting), віддалене виконання коду (Remote Code Execution) та імейл-ін'єкція [12].

3. *Уразливості впровадження файлів і шляхів.* Уразливості цієї категорії надають зловмисникові можливість управляти доступом до записів вебдодатків, структури їх документів та URL-адрес. Наявність таких уразливостей дозволяє йому підключати локальні файли з виведенням для читання на серверній частині або реалізувати віддалене виконання коду на сервері, що він атакує, зі свого сервера. Також зловмисник може маніпулювати параметрами URL-адреси з метою одержання доступу до файлів або виконання команд на серверній частині вебдодатка. Наприклад, це такі відомі вразливості, як RFI (Remote file include), LFI (Local file include) та обхід каталогу (Path traversal або Directory traversal) [12].

Клас 2. Некоректні автентифікація й авторизація (логічна помилка)

Автентифікація є частиною моделі безпеки AAA (*authentication, authorization, accounting* – автентифікація, авторизація, облік). Це процес, за допомогою якого система або додаток перевіряє надані користувачем облікові дані та призначає йому відповідні привілеї.

Некоректні механізми автентифікації й авторизації пов’язані з помилками в реалізації функцій автентифікації та політики контролю доступу. Такі недоліки дозволяють зловмисникові одержати доступ до конфіденційних вебсторінок і виконати несанкціоновані дії у вебдодатку.

Клас 3. Некоректне управління сеансом

Вебдодатки використовують механізм управління сеансами, за допомогою якого контролюють та управляють різними користувачами, які взаємодіють із вебдодатками. Управління сеансом охоплює всі процеси, від автентифікації користувача до його дій після виходу з вебдодатка. Будь-який сеанс взаємодії користувача й вебдодатка визначає унікальний ідентифікатор сеансу, за

яким потім кожен вебдодаток однозначно ідентифікує клієнта під час подальших з'єднань. Проте, зловмисники можуть перехоплювати сеансові ідентифікатори або обходити їх для одержання доступу до сеансів з'єднання користувачів. Такі проблеми часто пов'язані з некоректною реалізацією розробниками вебдодатків механізму генерації ідентифікаторів сеансу, їх передаванням мережею й збереженням на сервері.

Управління сеансом. Вебдодаток використовує сеанс для розпізнавання та об'єднання декількох вебзаписів від одного користувача впродовж певного періоду [2, 11]. Уразливості, що належать до цієї категорії: клікджекінг, міжсайтова підробка запитів (Cross-Site Request Forgery, CSRF), фіксація й захоплення сеансу [13]. Під час реалізації CSRF зловмисник відправляє запит зі шкідливим кодом у вебдодаток як законний користувач. Клікджекінг – це тип атаки, що пропонує користувачеві натиснути на об'єкти, розміщені зловмисником на вебсторінках. Водночас можуть відбуватися певні несанкціоновані дії без згоди самого користувача. Фіксація та перехоплення сеансу – це атаки, спрямовані на ідентифікатор сеансу користувача, а також на підроблення інших міжсайтових запитів і ручний клікджекінг [2, 14].

Систематизація методів виявлення вебуразливостей

Багато дослідників та фахівців з інформаційної й кібербезпеки досліджують проблеми пошуку та виявлення вразливостей у вебдодатках, які або є ще розроблюваними, або вже функціонують. Проаналізуємо різні методики й методи пошуку вебуразливостей, такі як статичний аналіз, фазинг і динамічний аналіз, техніки машинного навчання й безпечне програмування.

Безпечне програмування

Основою будь-якого вебдодатка є програмний код. Проте, часто розробники припускаються помилок під час його написання, що, у свою чергу, призводить до появи вразливостей у додатках. Також дехто з розробників не усвідомлює всієї важливості застосування прийомів безпечного програмування.

Додержання принципів безпечного програмування дозволяє враховувати в процесі розроблення програмного забезпечення всі можливі помилки та збої, своєчасно усувати їх і за можливості відновлювати функціонування програми в разі виникнення проблем. Безпечне програмування запобігає випадковому впровадженню вебуразливостей та забезпечує стійкість до впливу шкідливих програм і несанкціонованого доступу за умови, що в процесі розроблення програміст своєчасно перевіряє програмний код, усі вхідні дані, параметри запитів тощо. Щоб захистити вебдодатки від зловмисників, важливо уважно відстежувати реалізацію функцій безпеки на кожному етапі життєвого циклу розроблення вебдодатків.

Розглянемо певні програмні рішення, запропоновані дослідниками для безпечного програмування.

Наукові дослідження [15] присвячені використанню потоку інформації сервлетів (SIF) для створення безпечного вебдодатка. SIF – це нова структура для створення вебдодатків, за якого додержуються політик конфіденційності й цілісності інформації. Вебдодатки SIF розроблені на Jif 3.0. Jif – це мова програмування з безпечним типом, що розширює Java завдяки підтримці управління інформаційними потоками та контролю доступу, застосовуваних як під час компіляції, так і під час виконання. Jif написано на Java й побудовано з використанням розширюваного середовища компілятора

Java Polyglot [16]. Компілятор відстежує відповідність між інформацією та політиками, що обмежують її використання, забезпечуючи наскрізне додержання властивостей безпеки в системі. Після перевірки потоку інформації в програмах Jif компілятор Jif переводить їх у програми Java й використовує звичайний компілятор Java для створення захищених програм.

Автори [17] презентують FABLE, – базовий формалізм для мови програмування, у якому програмісти можуть визначати політики безпеки та причини, з огляду на які ці політики застосовують належним чином. У FABLE політики безпеки можуть бути поданими за допомогою зв'язування міток безпеки з даними або діями, що вони захищають. Програмісти визначають семантику міток в окремій частині програми, що називають політикою застосування. FABLE достатньо гнучкий, щоб реалізовувати широкий спектр політик безпеки, зокрема контроль доступу, інформаційний потік, автоматизацію безпеки. Автори реалізували FABLE як частину мови вебпрограмування LINKS.

Інший метод запропонований у праці [18]. Це інтелектуальна статична перевірка, що координує статичне дослідження в інтегрованому середовищі розроблення (IDE). Дослідники Д. Канг і Д. Парк [19] запропонували інтелектуальну систему, створену в результаті поєднання тестів «чорний» ящик і «білий» ящик, що могла б ефективно виявляти й розпізнавати слабкі місця програмного забезпечення.

Отже, нині розробленню та вдосконаленню методів запобігання й виявлення вразливостей у вебдодатках за допомогою безпечного програмування присвячено чимало досліджень. Розробники та компанії повинні зосередитися на тестуванні безпеки кожного біта програмного забезпечення й кожного програмного продукту. Вчасно

виконуючи зазначені дії на етапі розроблення вебдодатків, можна в подальшому скоротити витрати на убезпечення інформаційних ресурсів компанії.

Тестування методом «білого» ящика

Під час реалізації методу «білого» ящика тестувальник одержує доступ до програмного коду, а отже, повну інформацію щодо внутрішнього процесу, описаного за допомогою нього. Цей метод дозволяє аналізувати всі приховані в коді помилки й виправляти їх.

Перевага методу «білого» ящика – можливість легко передбачувати вхідні значення та створювати сценарій тестування. Проте цей метод потребує значного практичного досвіду та навичок для виявлення недоліків і помилок у програмному коді [19].

Тестування методом «чорного» ящика

Особливість методу «чорного» ящика полягає в тому, що тестувальник не володіє ніякою інформацією про структуру програмного додатка, його код. Він має змогу перевіряти функціональність додатка, відповідність його результатів вхідним значенням. Перевагою методу «чорного» ящика для тестувальника є те, що він не потребує знання початкового коду або певних технічних навичок. Проте під час тестування вхідних значень упродовж короткого часу є певні обмеження, що не дозволяють виявити логічних помилок та ускладнюють їх пошук без знання чітких функціональних специфікацій [19].

Статичний аналіз

Одним із ключових механізмів виявлення вразливостей вебдодатків є тестування їх безпеки (Application Security Testing – *AST*). Його реалізують через статичне (Static AST – *SAST*), динамічне (Dynamic AST – *DAST*) та інтерактивне (Interactive AST – *IAST*) тестування.

На етапі статичного тестування безпеки аналізують програмний код і компоненти, пов'язані з ним. Початковий код перевіряють для виявлення небезпечних конструкцій мови програмування, що можуть призвести до вразливостей. Однією з додаткових компонент початкового коду є зовнішні бібліотеки. Їх необхідно перевіряти на використання версій, у яких раніше виявлено вразливості. Також під час розроблення використовують системи контролю версій, наприклад Git, що може бути небезпечним і створювати загрозу розроблюваному вебдодатку. З огляду на це під час статичного тестування перевіряють репозиторій проєкту на небезпечні елементи програмного коду, що можуть призвести до компрометації конфіденційних даних.

Оскільки SAST-аналіз проводять до компіляції коду й без його виконання, такі інструменти використовують на ранніх етапах життєвого циклу розроблення програмного забезпечення. Більшість інструментів SAST підтримують основні мови вебпрограмування: PHP, Java, .Net, а також певні форми C, C++ або C #.

Перевагами статичного тестування є такі:

1) його інструменти виявляють критичні вразливості на перших етапах розроблення, що можна швидко усунути;

2) оскільки такий тип тестування безпеки встановлює специфіку проблеми, зокрема рядки коду, він спрощує й виправлення помилок;

3) його можна інтегрувати в наявне середовище на різних етапах циклу розроблення вебдодатка;

4) для аналізу програмного коду потрібно менше часу, ніж для ручного тестування вже готового продукту [20].

Серед недоліків SAST виділяють такі:

1) не всі компанії або приватні особи бажають надавати дані для аналізу двійкового або байт- і початкового кодів;

2) масштабне тестування такого типу може виявитися складним завданням, тому що має тенденцію до неточного моделювання поведінки коду. Унаслідок цього розробникам доводиться стикатися з безліччю помилкових спрацьовувань та помилково негативних результатів;

3) мови з динамічною типізацією спричиняють проблеми, тому що інструменти SAST повинні семантично розуміти чимало динамічних частин коду, що можуть бути написаними різними мовами програмування;

4) не має функції тестування додатка в реальному середовищі, тому вразливостей логіки програми або небезпечних конфігурацій не можливо виявити [20].

Для проведення SAST використовують такі види програм: статичні аналізатори коду, програми для перевірки залежностей проєкту та програми для перевірки систем контролю версій.

Нині на ринку репрезентовані різні SAST-рішення, як комерційні, так і безкоштовні. Найпопулярнішими серед них є IBM Security AppScan Source, Synopsys Coverity SAST, Veracode Application Security Platform, Positive Technologies Application Inspector, Checkmarx SAST.

Інструменти SAST є дуже цінною технологією, але не замінюють інших методів. Розробникам доцільно комбінувати різні методи впродовж усього процесу життєвого циклу розроблення для оцінювання й виявлення недоліків перед запуском у масове користування.

Динамічний аналіз і фазинг

Динамічне тестування безпеки (DAST) застосовують для тестування програми або програмного продукту в робочому стані. У рамках цього виду тестування використовують сканери уразливостей вебдодатків, з метою виявлення таких вебуразливостей, як:

- XSS;
- SQL-ін'єкції;
- RCE (Remote code execution);
- XXE (External XML entity);
- Path traversal;
- некоректні конфігурації серверів.

До переваг такого виду тестування належить те, що сканери вебуразливостей можуть перевірити значну кількість точок входу вебдодатка або іншого програмного продукту за допомогою базових атакових векторів.

Недоліки динамічного тестування такі:

1) його інструменти не дозволяють зрозуміти основних причин вебуразливостей;

2) такий тип тестування не орієнтований на початкові стадії розроблення вебдодатка, його доречно проводити лише для додатка, що функціонує;

3) він не ідеально імітує потенційні атаки, тому що експлойти часто виконує сторона, яка має внутрішню базу знань про вебдодаток.

Для динамічного аналізу застосовують фазинг, що є технологією тестування програм в автоматичному режимі для виявлення потенційних уразливостей. Упродовж такого тестування майже повністю відсутні помилкові спрацьовування, характерні для статичних аналізаторів. При цьому забезпечується велика кількість граничних значень шляхом створення некоректних вхідних даних. Вхідними даними є файли та інша

інформація, опрацьована досліджуваним вебдодатком, зокрема визначеними протоколами обміну, прикладними інтерфейсними функціями тощо [21].

Фазинг являє собою автоматичне тестування програмного забезпечення або вебдодатка, за якого на вхід досліджуваної програми подають випадкові, модифіковані або некоректні значення, що спричиняють аномалії в поведінці програми. За допомогою фазингу можна виявляти такі аномалії: розмір відповідей сервера, що істотно відрізняється від типового; різні статус-коди, зміну часу відповіді на запит тощо.

Залежно від методики генерування вхідної інформації технології фазингу умовно поділяють на два класи: 1) мутація наявних даних; 2) генерація нових даних [21].

Спрощений алгоритм, що реалізує методику застосування фазингу для аналізу вразливостей програмного забезпечення наведено на рисунку 10.2.

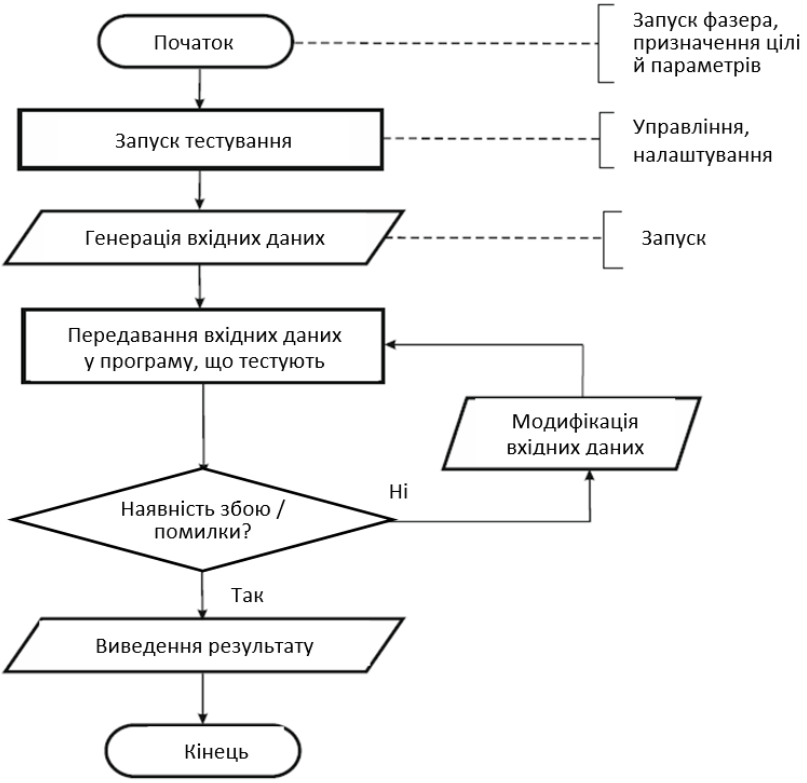


Рисунок 10.2 – Спрощений алгоритм роботи фазера

Інтерактивне тестування безпеки вебдодатків

Інтерактивне тестування безпеки вебдодатків поєднало переваги статичного та динамічного видів.

Зазначене тестування використовує програмні інструменти для оцінювання роботи вебдодатка й виявлення вразливостей. Воно передбачає «агент-подібний» підхід, за якого агенти й датчики запускають для безперервного аналізу функціонування додатка під час автоматизованого, ручного тестування або їх комбінації.

Процес і зворотний зв'язок відбуваються в режимі реального часу в інтегрованому середовищі розроблення, середовищі безперервної інтеграції, системі контролю якості або процесі виробництва. Датчики мають доступ до:

– програмного коду;
– потоку даних і потоку управління;
– даних конфігурації системи;
– вебкомпонентів;
– даних внутрішнього підключення.

Основна відмінність IAST від статичного й динамічного тестування полягає в тому, що таке тестування функціонує всередині додатка (програми). Доступ до ширшого діапазону даних підвищує ефективність IAST порівняно з наявністю лише початкового коду або можливості сканування лише HTTP-протоколу, а також забезпечує більш точний результат тестування безпеки вебдодатка.

Методи машинного навчання

Останнім часом застосування машинного навчання у сфері інформаційного захисту має вагомі результати. Машинне навчання – це процес, упродовж якого за допомогою спеціалізованих технологічних інструментів комп'ютер може вивчати й використовувати нові дані без обов'язкового втручання людини. Продумані алгоритми дозволяють комп'ютеризованій платформі опрацьовувати та «розуміти» дані з величезних сховищ інформації, щоб формулювати певні висновки й виявляти закономірності – патерни. Комп'ютерна система аналізує такі патерни, групує їх за певними ознаками, на основі чого потім робить свої висновки або припущення.

Системи на базі машинного навчання – це програми, функціонально здатні навчатися, використовуючи масиви даних, що дібрані та розмічені людиною. Чим більше така програма повторює цикл

розпізнавання та присвоєння патернам категорій, щоб робити на їх основі висновки, тим краще вона «розуміє», як це можна здійснювати самостійно, без допомоги людини або додаткових, написаних фахівцями сценаріїв.

Машинне навчання вважають абсолютно іншим підходом до аналізу широкого спектра вебдодатків. Водночас його також можна використовувати для виявлення вебуразливостей у початковому коді.

Аналіз програмного коду на помилки й небезпечні конструкції, що в подальшому можуть призвести до порушення безпеки вебдодатків, базується на різноманітному наборі методів машинного навчання. Серед основних завдань, виконуваних за допомогою методів машинного навчання під час аналізу програмного коду, можна виділити такі:

1) класифікацію, що полягає у визначенні належності об’єкта до одного з класів на підставі навчання на інших відомих об’єктах;

2) кластеризацію, що поділяє об’єкти на згруповані за певними ознаками множини.

Методи класифікації й кластеризації застосовують для прогнозування дефектів (помилок) у програмному коді, оцінювання його однорідності (визначення фрагментів, що відрізняються), розпізнавання та виявлення аномалій у початковому коді, патернів уразливого коду тощо [22].

Основні напрями застосування методів машинного навчання для аналізу вразливостей вебдодатків наведені на рисунку 10.3 [22].

Рисунок 10.3 – Аналіз уразливостей
на основі машинного навчання

Висновки

У рамках дослідження проведено всебічний огляд методів, що допомагають своєчасно виявляти вебуразливості й запобігати їх появі. Розглянуто одну з класифікацій вебуразливостей, а також систематизовано методи та техніки їх виявлення, такі як безпечне програмування, тестування безпеки вебдодатків (статичне, динамічне, інтерактивне), фазинг і машинне навчання. Водночас поданий матеріал висвітлює і ті проблемні напрямки дослідження безпеки вебдодатків, які ще потребують вирішення, незважаючи на значну кількість наукових праць іноземних фахівців з інформаційної й кібербезпеки.

СПИСОК ЛІТЕРАТУРИ

1. Pop D. P. Designing an MVC model for rapid web application development / D. P. Pop, A. Altar // Procedia Engineering. – 2014. – Vol. 69. – P. 1172–1179.

2. Deepa G. Securing web applications from injection and logic vulnerabilities: Approaches and challenges /

G. Deepa, P. S. Thilagam // Information and Software Technology. – 2016. – Vol. 74. – P. 160–180.

3. Awoleye O. M. Web application vulnerability assessment and policy direction towards a secure smart government / O. M. Awoleye, B. Ojuloge, M. O. Ilori // Government Information Quarterly. – 2014. – Vol. 31. – P. S118–S125.

4. Уязвимости и угрозы веб-приложений в 2019 году [Электронный ресурс] // Официальный сайт компании «Positive Technologies». − Режим доступа : https://www.ptsecurity.com/ru-ru/research/analytics/web-vulnerabilities-2020/.

5. OWASP Top 10 [Electronic resource] // OWASP Foundation. – Access mode : https://owasp.org/www-project-top-ten/.

6. Bozic J. PURITY: a Planning-based secURITY testing tool / J. Bozic, F. Wotawa // Software Quality, Reliability and Security-Companion (QRS-C). (2015 IEEE International Conference). – P. 46–55.

7. ДСТУ ISO / IEC 29147:2016 Інформаційні технології. Методи захисту. Розкриття вразливостей (ISO / IEC 29147:2014, IDT). – Київ, 2016. – 32 с.

8. Tsipenyuk K. Seven pernicious kingdoms: A taxonomy of software security errors / K. Tsipenyuk, B. Chess, G. McGraw // IEEE Security & Privacy. – 2005. – № 3 (6). – P. 81–84.

9. Meunier P. Classes of vulnerabilities and attacks // Wiley Handbook of Science and Technology for Homeland Security. – John Wiley & Sons, 2008. – P. 947–965.

10. Defending Against Web Application Attacks: Approaches, Challenges and Implications / D. Mitropoulos, P. Louridas, M. Polychronakis, A. D. Keromytis // IEEE Transactions on Dependable and Secure Computing. – 2017. – Vol. 16 (2). – P. 188–203.

11. Xiaowei Li A survey on server-side approaches to securing web applications / Li Xiaowei, Yuan Xue // ACM Computing Surveys. – 2014. – Vol. 46 (4). – P. 1–29.

12. Medeiros I. Automatic detection and correction of web application vulnerabilities using data mining to predict false positives / I. Medeiros, N. F. Neves, M. Correia // Proceedings of the 23rd international conference on World Wide Web. – 2014. – P. 63–74.

13. Wedman S. An analytical study of web application session management mechanisms and HTTP session hijacking attacks / S. Wedman, A. Tetmeyer, H. Saiedian // Information Security Journal: A Global Perspective. – 2013. – Vol. 22 (2). – P. 55–67.

14. Shahriar H. Client-side detection of cross-site request forgery attacks / H. Shahriar, M. Zulkernine // Software Reliability Engineering (2010 IEEE 21st International Symposium). – P. 358–367.

15. Chong S. SIF: Enforcing Confidentiality and Integrity in Web Applications / S. Chong, K. Vikram, A. C. Myers // USENIX Security Symposium. – 2007. – P. 1–16.

16. Pullicino K. Jif: Language-based Information-flow Security in Java [Electronic resource] // ArXiv. – 2014, abs/1412.8639. – Access mode : https://ui.adsabs.harvard.edu/abs/2014arXiv1412.8639P.

17. Swamy N. Fable: A language for enforcing user-defined security policies / N. Swamy, B. J. Corcoran, M. Hicks // Security and Privacy (2008 IEEE Symposium). – P. 369–383.

18. Zhu J. Supporting secure programming in web applications through interactive static analysis / J. Zhu, J. Xie, H. R. Lipford, B. Chu // Journal of advanced research. – 2014. – Vol. 5 (4). – P. 449–462.

19. Kang J. A secure-coding and vulnerability check system based on smart-fuzzing and exploit / J. Kang, J. H. Park // Neurocomputing. – 2017. – Vol. 256. – P. 23–24.

20. SAST, DAST, IAST, and RASP: how to choose? [Electronic resource] // Official website of the company «Positive Technologies». – Access mode : https://www.ptsecurity.com/ww-en/analytics/knowledge-base/sast-dast-iast-and-rasp-how-to-choose/.

21. Разработка методики применения фаззинга для анализа уязвимостей программного обеспечения / И. О. Томилов, И. Н. Карманов, П. А. Звягинцева, Е. В. Грицкевич // Системы управления, связи и безопасности. – 2018. – № 4. – С. 48–62.

22. Noman M. A Survey on Detection and Prevention of Web Vulnerabilities / M. Noman, M. Iqbal, A. Manzoor // International Journal of Advanced Computer Science and Applications. – 2020. – Vol. 11 (6). – P. 521–540.

РОЗДІЛ 11
НАВЧАННЯ АНГЛІЙСЬКОЇ МОВИ СТУДЕНТІВ СПЕЦІАЛЬНОСТІ 125 «КІБЕРБЕЗПЕКА» ІЗ ЗАСТОСУВАННЯМ ЕВРИСТИЧНИХ МЕТОДІВ

Т. М. Плохута

Самостійна пізнавально-творча діяльність студентів ефективно активізується під час евристичного навчання, що базується на критичному мисленні студентів, умінні ставити евристичні запитання, передбачає виконання завдань відкритого типу. Вона активізує знання й уміння, набуті на заняттях, дозволяє студентам застосовувати власні підходи до дослідження навчальних явищ та об'єктів, часто міждисциплінарних, сприяючи створенню самостійних творчих освітніх продуктів.

Дослідник А. В. Хуторський зазначає, що на відміну від традиційного навчання, у якому об'єктом пізнання є наукові теорії, поняття й закономірності, відповідні світу реальних об'єктів (безпосередня робота з якими дуже незначна), у евристичному навчанні як первісні пропонуються реальні об'єкти пізнання. Викладач ознайомлює студентів зі способами їх пізнання, допомагає визначити особистісні освітні цілі відповідно до досліджуваної теми, організує та спрямовує процес самостійної роботи студентів зі створення власних освітніх продуктів. Якщо студент недостатньо володіє способами пізнавальної діяльності щодо невідомих йому освітніх об'єктів, викладач пропонує застосовувати ті чи інші методи пізнання, здійснюючи, таким чином, супроводжувальну діяльність.

Аналіз сучасних наукових джерел доводить ефективність евристичного навчання для активізації самостійної пізнавально-творчої діяльності студентів, а евристичні методи:

1) підсилюють зміст навчальної програми.

Евристичні методи ефективні для покращення розуміння основних концепцій. Це пов'язано з впливом зацікавленості на мозок. Якщо концепція викликає допитливість, підвищується активність в гіпокампі – ділянці мозку, що відповідає за пам'ять;

2) «розігрівають» мозок для навчання.

Проведення короткого опитування на початку заняття може допомогти студентам краще засвоювати інформацію впродовж заняття. Також допитливість готує мозок до навчання, дозволяючи підвищити ефективність розуміння та запам'ятовування понять, набуття навичок. Евристичні запитання допомагають почати заняття, стимулюючи зацікавленість, інтелектуально «провокуючи»;

3) сприяють глибшому осмисленню інформації.

Завдяки евристичним методам багато студентів розуміють, як розвивається ідея, чому правило чи формула працюють, коли вони можуть правильно застосувати правило, ідею чи формулу.

Це можна пояснити тим, що процес постановки відкритих запитань за допомогою оригінальних стратегій дає можливість студентам брати активну участь у своєму навчанні. Той самий принцип застосовують до експериментального навчання, що ставить здобувачів освіти у центр навчального процесу;

4) формують ініціативу до самовдосконалення.

Студенти можуть покращити певні набуті раніше вміння завдяки евристичним методам, багато з яких стосуються ініціативи та самоуправління. Зокрема,

навчитися ставити запитання, досліджувати, обговорювати, співпрацювати й робити власні висновки. Незважаючи на те що вони можуть вдосконалювати наведені навички за допомогою інших видів діяльності, самостійна перевірка та аналіз пришвидшують цей розвиток.

Дослідження евристичних методів дозволило виокремити фактори, що підвищують ефективність навчального процесу в разі застосування зазначених методів на заняттях з іноземної мови:

– індивідуалізація навчання й створення проблемних ситуацій з метою вдосконалення мовних умінь і навичок;

– активізація творчої та пізнавальної діяльності;

– підвищення мотивації до творчої й навчальної діяльності;

– створення умов для самостійної роботи.

Отже, евристика полегшує прийняття рішень, дозволяючи студентові робити це виважено та продуктивно. Аналітики в будь-якій галузі застосовують евристику для вирішення різних проблем. Евристичні методи роблять процес прийняття рішень простішим і швидшим за допомогою раціональності та обґрунтованих роздумів.

На сьогодні в Сумському державному університеті простежується тенденція до активного залучення студентів, зокрема спеціальності 125 «Кібербезпека», до науково-дослідної діяльності, участі в наукових конференціях. Створення творчих робіт є одним з важливих і актуальних способів формування в здобувачів освіти умінь самостійної пізнавально-творчої діяльності, зокрема грамотно працювати з інформацією, а також аналізувати й презентувати її. Проте згідно з результатами аналізу створених студентами пізнавально-творчих

продуктів (анотацій, рецензій статей відомих науковців, власних студентських наукових статей, тез, усних доповідей за фаховим спрямуванням, письмових професійних проблемних творів, наукових рефератів тощо) довів, що творчі роботи, зокрема з дисципліни «Іноземна мова», не завжди відповідають вимогам якості, а студенти недостатньо володіють уміннями самостійної пізнавально-творчої діяльності.

Компетентно оцінити результати зазначеної діяльності для їх подальшого корегування, доопрацювання, виправлення можна способом публічної апробації (лат. *approbation* – схвалення) у вигляді наукової доповіді на практичному занятті або науково-практичній конференції. Під час евристичного навчання на заняттях із дисципліни «Іноземна мова» проблеми підготовки презентації наукової доповіді допомагають вирішити евристичні прийоми, методи генерування нових ідей, спрямованих на риторичний винахід, тобто винахід предмета публічного виступу (усної доповіді, промови), його змісту, форм і методів мовленнєвої діяльності. У рамках педагогічного експерименту підготовка й захист наукової доповіді іноземною (англійською) мовою були ретельно спланованим викладачем, студентів спеціальності 125 «Кібербезпека» ознайомили з основними етапами (табл. 11.1) [2].

На етапі опанування евристичних методів і прийомів для формування умінь аналізувати, систематизувати, класифікувати інформацію тощо, студенти спеціальності 125 «Кібербезпека» вчилися формулювати евристичні запитання на запропоновану викладачем або іншими студентами тему, застосовуючи метод поставлення креативних запитань і серію запитань, що стимулюють критичне мислення (Questions Provoking Critical Thinking) американського психолога й педагога Е. Кінг [4].

Таблиця 11.1 – Етапи підготовки до публічного виступу

Етап	Короткий опис
1	Ознайомлення студентів із відеозаписами й текстами публічних доповідей провідних науковців, викладачів, аспірантів, магістрантів, студентів з електронної бази даних кафедри іноземних мов Сумського державного університету
2	Опанування комплексу контрольно-діагностичних критеріїв для всебічного та об'єктивного аналізу й оцінювання взірців публічних виступів
3	Опанування евристичних методів і прийомів для формування вмінь аналізувати, систематизувати, класифікувати інформацію тощо
4	Вибір, переформулювання в рамках загальної тематики запропонованої викладачем теми публічного виступу або формулювання власної
5	Самостійне обмірковування теми, складання попереднього плану виступу з означеної проблеми, ознайомлення з науковою літературою та вибір джерел, що достатньо повно розкривають вибрану тему
6	Написання тексту публічного виступу (наприклад, професійного спрямування) на основі евристичних приписів і репрезентування його широкому загалу для корекції
7	Підготовка наочного матеріалу – електронної презентації (PowerPoint) виступу
8	Репетиція публічного виступу вдома (бажано перед дзеркалом та з «аудиторією»), у малих групах (3–5 осіб) на індивідуальних заняттях. Під час репетиції студенти мають змогу переконатися в тому, що вони вкладаються в часові межі виступу (5–7 хвилин), перевірити роботу апаратури (комп'ютера, проєктора), синхронізувати електронну презентацію з усним виступом. Також студенти-слухачі й викладач можуть порадити внести певні корективи в зміст або форму презентації доповіді
9	Власне публічний виступ студента з науковою доповіддю
10	Обговорення основних положень запропонованої проблеми, серія запитань і відповідей
11	Оцінювання публічного виступу (самооцінювання, взаємооцінювання, експертне оцінювання)
12	Запис телепрезентації

Наведемо приклади творчих завдань для активізації самостійної пізнавально-творчої діяльності студентів спеціальності 125 «Кібербезпека» для підготовки до публічного англомовного виступу на навчальній конференції зі спеціальності.

Creative task 1 Use the list of questions provoking critical thinking to research one of the suggested topics or your own topic.

Common Cybersecurity Topics

1 Network Security.
2 E-mail Security.
3 Web Security.
4 NGFW (Next Generation Firewall).
5 Data Loss Prevention (DLP).
6 Cloud Security .
7 Intrusion Detection Systems (IDS) or Intrusion Prevention Systems (IPS).
8 Identity and Access.
9 Cryptography.
10 Antivirus/anti-malware.
11 Ransomware.
12 Phishing Attacks.
13 Social engineering.
14 Advanced Persistent Threat.
15 Botnet.

Questions Provoking Critical Thinking

Thinking Skills	Sample Action Prompts	Example Questions
Remembering	recognize, list, describe, identify, retrieve, name	What do we already know about...? What are the principles of … ? How does ... tie in with what we learned before?
Understanding	describe, generalize explain, estimate, predict	Summarize … or Explain … What will happen if … ? What does ... mean?
Applying	implement, carry out, use, apply, show, solve, hypothesize	What would happen if…? What is a new example of…? How could … be used to...? What is the counterargument for...?
Analyzing	compare, organize, deconstruct	Why is ... important? What is the difference between… and…? What are the implications of...? Explain why / Explain how? What is ... analogous to? How are ... and ... similar?
Evaluating	check, critique, judge, conclude, explain	How does ... affect...? Why is ... happening? What is the best ... and why? Do you agree or disagree with the statement...? What evidence is there to support your answer? What are the strengths and weakness of? What is the nature of…?
Creating	design, construct, plan, produce	What is the solution to the problem of...? What do you think causes...? Why? What is another way to look at...?

From Alison King, "Inquiring Minds Really Do Want to Know: Using Questioning to Teach Critical Thinking".

Creative task 2 Use this technology to generate and transform your questions.

From
A thinking routine from Project Zero,
Harvard Graduate School of Education
URL: http://www.pz.harvard.edu/sites/default/files/Creative%20Questions.pdf

CREATIVE QUESTIONS

1 Pick one topic and brainstorm a list of questions about it.

2 Look over the list and transform some of the questions into questions that challenge the imagination. Do this by transforming questions along the lines of:

- *What would it be like if ...*
- *How would it be different if ...*
- *Suppose that ...*
- *What would change if ...*
- *How would it look differently if …*

3 Choose a question to imaginatively explore. Explore it by imaginatively playing out its possibilities. Do this by: Writing a story or essay, drawing a picture, creating a play or dialogue, inventing a scenario, conducting an imaginary interview, conducting a thought experiment.

4 Reflect: What new ideas do you have about the topic, concept or object that you didn't have before?

Creative task 3 Make a plan of your future presentation using your own list of questions provoking critical thinking and your own list of creative questions.

Creative task 4 Create PowerPoint Presentation to help you speak and make your public speech more attractive to the audience.

Creative task 5 Prepare a written report and an oral presentation on one of the suggested topics or other topic related to CYBERSECURITY to take part in the conference.

Також студентам спеціальності 125 «Кібербезпека» били запропонованими корисні посилання на сучасні сайти для самостійного аналізу та опрацювання вимог до створення PowerPoint-презентацій, ознайомлення з глосарієм найпоширеніших термінів і скорочень зі спеціальності, вебінари сучасних учених, присвячені найпопулярнішим темам у галузі кібербезпеки тощо.

Recommended Links

1. How to Make a Good PowerPoint Presentation (tips)
URL: https://www.youtube.com/watch?v=grJ0FbpfvOw.
2. How To Speak by Patrick Winston (video)
URL: https://www.youtube.com/watch?v=Unzc731iCUY.
3. EnglishCLUB. URL: https://www.englishclub.com/speaking/presentations.htm.
4. Cybersecurity Glossary of Terms
URL: https://www.globalknowledge.com/us-en/topics/cybersecurity/glossary-of-terms/.
5. Perforce URL: https://www.perforce.com/p/kw/top-embedded-software-cybersecurity-vulnerabilities?utm_leadsource=cpc-googleadwords&utm_source=googleadwords&utm_medium=cpc&utm_campaign=KlocworkEMEA&utm_adgroup=Cybersecurity&gclid=CjwKCAiA_9r_BRBZEiwAHZ_v16pr3rrV9Av0ERYV7TmeT5g8P4orheyr5fmfeTW-dHHtTF0XxoUjKBoCC9sQAvD_BwE.
6. Everything Tech! Tweak library URL: https://tweaklibrary.com.
7 Brighttalk. Preventing Attacks on IoT Networks and Devices (webinar). URL: https://www.brighttalk.com/

webcast/16731/460529?utm_campaign=channel-feed&utm_source=brighttalk-portal&utm_medium=web.

8. Brighttalk. Protecting Cloud Assets from DDoS Threats (webinar). URL: https://www.brighttalk.com/webcast/16731/457595?utm_campaign=channel-feed&utmsource=brighttalk-portal&utm_medium=web.

9. Brighttalk. End-to-End Autonomic Closed-Loop Security Management & Control for 5G Networks (webinar). URL: https://www.brighttalk.com/webcast/16731/457497?utm_campaign=channel-feed&utm_source=brighttalk-portal&utm_medium=web.

10. Brighttalk. New Year, New Cloud. URL: https://www.brighttalk.com/webcast/16731/457175?utm_campaign=channel-feed&utm_source=brighttalk-portal&utm_medium=web.

На заняттях із дисципліни «Іноземна мова» для активізації самостійної пізнавально-творчої діяльності студентів ураховували принципи евристичних запитань, сформульовані О. Морозовим: 1) проблемності й оптимальності (способом майстерного ставлення запитань проблемність завдання знижується до оптимального рівня); 2) дроблення інформації (евристичні запитання дозволяють поділити завдання на підзавдання); 3) цілепокладання (кожне нове евристичне запитання формує нову стратегію – мету діяльності) [3, с. 388].

Перш ніж виконувати творчі завдання, студентам необхідно розглянути типологію запитань, що базується на таксономії Б. Блума, так звану «Ромашку Блума»:

1) прості запитання – запитання, відповідаючи на які, потрібно назвати конкретні факти, пригадати та відтворити певну інформацію («Що?», «Де?», «Коли?»);

2) уточнювальні запитання – запитання, спрямовані на одержання невідомої інформації («Яка природа . . . ? » , « Яка різниця між … та … ? »);

3) інтерпретаційні запитання – запитання, спрямовані на встановлення причинно-наслідкових зв'язків («Чому …?»);

4) творчі запитання – запитання з елементами умовності, припущення, прогнозу («Що змінилося б у світі, якщо б …?»);

5) оцінні запитання – запитання, спрямовані на з'ясування критеріїв оцінювання тих або інших подій, явищ, фактів («Чому щось добре, а щось погано?», «Чим …. відрізняється від …?»);

6) практичні запитання – запитання спрямовані на встановлення взаємозв'язку між теорією й практикою («Де ви у звичайному житті можете спостерігати …?», «Як би ви вчинили на місці …?») [3, с. 385].

Також на практичних, консультаційних заняттях і під час самостійної роботи студенти опановували евристичні приписи для підготовки та проведення публічного виступу, розроблені Г. Онуфрієнко, в основу яких покладені евристичні запитання [1].

Евристичні приписи для підготовки
й проведення публічного виступу
Як описати об'єкт

1. Які істотні характеристики об'єкта (розмір, форма, властивості)?
2. Яка його структура (склад елементів, їх зв'язки та відносини)?
3. Чим він відрізняється від подібних, близьких йому об'єктів?
4. Яка історія появи об'єкта?
5. Яке його призначення?
6. Хто найчастіше використовує об'єкт?
7. Для чого можна його використовувати з найбільшою ефективністю?

Опишіть конкретний об’єкт, наприклад комп’ютер, без і з використанням евристичного припису. У чому різниця?

Як описати подію

1. Хто (що), коли, чому навіщо щось зробив?
2. Які умови, обставини події?
3. Як можна кваліфікувати подію?
4. У чому подібність і відмінність від аналогічних подій?
5. Із якого джерела вам відомо про подію? Чи надійне воно?
6. Чи можна було змінити або уникнути події?
7. Які можливі наслідки події?

Спробуйте описати конкретну подію без і з додержанням евристичного припису, і ви переконаєтеся, що він істотно підвищує ефективність вашої мовленнєвої діяльності. Наприклад, опишіть один з останніх конфліктів у вашому житті.

Як висувати твердження в процесі доказу або спростування

1. Поділіть вихідне твердження на складові.
2. У якій послідовності краще використовувати висунуті Вами твердження?
3. Як можна змінити твердження, підсиливши його?
4. Що є основним у твердженні, чим це можна довести?
5. На основі чого ви встановили істинність або хибність висунутого твердження (авторитетної думки, статистики, спостереження, особистого досвіду, іншого)?
6. Що випливає з ваших тверджень?
7. Який ступінь доведення або спростування, чого ви досягаєте в результаті висунення ваших тверджень?
8. Чи закликає ваш доказ або твердження до дії?

Спробуйте спочатку без евристичного припису, а

потім за допомогою нього довести, що знання двох іноземних мов робить випускника вищої школи конкурентоздатнішим і в яких ситуаціях?

Отже, запропоновані серії запитань стимулюють критичне мислення, дають студентам можливість досліджувати об'єкти пізнання під різними кутами, порівнювати їх з іншими, детальніше аналізувати й оцінювати їх недоліки та переваги, знаходити способи використання об'єкта в повсякденному житті та перспективи його подальшого розвитку. Самостійно формулюючи евристичні запитання та шукаючи відповіді на них, студенти окреслюють коло незнаного й поступово заповнюють його значущими для себе даними, знаходять оригінальні способи виконання творчого завдання.

Під час підготовки до усної доповіді студенти спеціальності 125 «Кібербезпека» ознайомлювалися з етапами, цілями, прийомами та способами публічного виступу (таблиця 11.2) і користувалися цими розробками в процесі написання тексту доповіді.

Найважливішим у межах нашого експерименту був етап оцінювання публічного виступу. Він ускладнювався тим, що усна форма подання інформації є швидкоплинною, її важко зафіксувати. На відміну від письмового твору, статті, реферату, які може переглянути рецензент декілька разів для знаходження недоліків і прогалин, усну доповідь неможливо відтворити в режимі он-лайн у тому самому первинному вигляді ще раз. Ураховуючи цей факт, був вибраним такий вид виступу, як *телепрезентація* [2].

Запис усного виступу студента, зокрема англійською мовою, дозволяє детальніше проаналізувати освітній продукт, визначити його недоліки й переваги, порівняти з попередніми роботами студента.

Таблиця 11.2 – Етапи, цілі, прийоми і засоби публічного виступу (В. І. Андрєєв)

Етап	Мета	Прийом і засіб
1. Вступ	Звернути увагу аудиторії, зацікавити її, завоювати довіру	Почати виступ із неочікуваної репліки, факту, гумористичних зауважень
2. Постановка проблеми	Висвітлити актуальність проблеми, проаналізувати основні протиріччя і підпроблеми, сформулювати загальну проблему	Звернення до інтересів слухачів, їх потреб, посилання на факти, документи, авторитетні висловлювання, аналіз усталених, але неправильних точок зору. Демонстрація особистої зацікавленості у вирішенні проблеми
3. Розчленування проблем на підпроблеми, завдання, питання	Чітке виділення переліку проблем, завдань, питань, розкриття їх сутності	Обгрунтування логіки розроблення загальної схеми вирішення проблеми, ідеї, гіпотези, способу, можливих результатів
4. Виклад підходів, способів вирішення проблем	Розкриття в порівняльному аналізі як власних підходів, так і альтернативних точок зору, способів вирішення проблеми	Доказові судження, аргументи, використання засобів критичного аналізу, порівняння, зіставлення
5. Узагальнення, завершення	Сконцентрувати увагу аудиторії на головному, резюмувати викладене	Твердження, що інтегрує основну ідею, думку. Використання найсильнішого аргументу, крилатої фрази, афоризму

У процесі підготовки й запису (здійснюваних

самостійно або з допомогою інших студентів) телепрезентації наукової доповіді студенти спеціальності 125 «Кібербезпека» мали змогу заздалегідь проглянути відеоматеріал, проаналізувати його, знайти помилки та виправити їх. Підготовка телепрезентації – гарна можливість для самооцінювання й самовдосконалення. Запис виступу також зручно використовувати для колективного обговорення переваг і недоліків створеного продукту, проведення взаємооцінювання й під час кінцевого оцінювання результатів самостійної пізнавально-творчої діяльності студентів.

Під час оцінювання усних доповідей студентів англійською мовою додатково враховували їх мовні та мовленнєві особливості (правильність вживання граматичних, лексичних, стилістичних одиниць).

Варто зазначити, що публічні виступи оцінювали на основі поваги, довіри між оратором і рецензентом. В іншому разі це могло не лише не принести користь, а й стати причиною неприязних, ворожих відносин, навіть призвести до «творчого застою».

Логічним завершенням кропіткої роботи над професійно спрямованою доповіддю є її публічне презентування на конференціях. Студенти спеціальності 125 «Кібербезпека» мають змогу виступати зі своїми професійно-творчими доробками на щорічних студентських науково-практичних конференціях англійською мовою.

У результаті експерименту встановлено ефективність евристичних методів навчання для активізації самостійної пізнавально-творчої діяльності студентів спеціальності 125 «Кібербезпека». Зокрема, кількість студентів із творчим рівнем сформованості предметно-методичної компетентності в експериментальній групі (ЕГ) збільшилася на 25 %,

інформаційно-комунікативної – на 17,5 %, діагностико-прогностичної – на 5 %, конструктивно-творчої – на 10 %. У контрольній групі (КГ) прирости цих показників менш істотні, відповідно: +15,6 %, +3,9 %, +1,3 %, +5,2 %.

Отже, застосування на практиці евристичного підходу для самостійного виконання студентами творчих завдань довело, що уміння ставити евристичні запитання допомагає здобувачам освіти детальніше окреслити межі знаного й незнаного, знаходити новизну в досліджуваній проблемі, оперативно та ефективно виконувати зазначене завдання.

Проведене дослідження не вичерпує всіх аспектів проблеми використання евристичних запитань для активізації самостійної пізнавально-творчої діяльності студентів, що на сьогодні посідає вагоме місце в професійній освіті. Перспективним, на нашу думку, залишається, зокрема, детальніше розроблення змісту й структури роботи над підготовкою якісних освітніх продуктів із дисципліни «Іноземна мова» із застосуванням евристичного підходу.

СПИСОК ЛІТЕРАТУРИ

1. Онуфрієнко Г. С. Науковий стиль української мови : навчальний посібник з алгоритмічними приписами / Г. С. Онуфрієнко. – Київ : Центр учбової літератури, 2009. – 392 с.

2. Плохута Т. М. Евристичне тестове завдання як ефективний засіб діагностики й оцінювання самостійної пізнавально-творчої діяльності студентів / Т. М. Плохута // Засоби навчальної та науково-дослідної роботи : Збірник наукових праць. – Харків : ХНПУ імені Г. С. Сковороди. – Харків, 2011. – Вип. 35. – С. 97–107.

3. Плохута Т. М. Евристичні запитання як основа критичного мислення / Т. М. Плохута // Педагогічні науки:

теорія, історія, інноваційні технології : науковий журнал / за ред. А. А. Сбруєва, О. Є. Антонова, Дж. Бішоп та ін. – Суми : СумДПУ ім. А. С. Макаренка, 2015. – № 9 (53). – С. 383–390.

4. King A. Inquiry as a tool in critical thinking / A. King // Changing college classrooms: new teaching and learning strategies in an increasingly complex world. – San Francisco : Jossey-Bass, 1994. – P. 13–38.

РОЗДІЛ 12
ТЕРМІНОЛОГІЯ У СФЕРІ КІБЕРБЕЗПЕКИ: ЗАГАЛЬНІ ПИТАННЯ ТЕРМІНОТВОРЕННЯ, СИСТЕМАТИЗАЦІЇ ТА УНІФІКАЦІЇ

О. П. Сидоренко, О. А. Шовкопляс

Важливим напрямом у сфері сучасної лінгвістики є дослідження галузевої термінології. Питання становлення та розвитку термінологічної системи в галузі кібербезпеки є *актуальним* з огляду на те, що ця сфера суспільного життя, зумовлена стрімким розвитком інформаційного суспільства та інформаційної інфраструктури України, є достатньо новою (Національна система кібербезпеки в Україні юридично визначена у 2016 році) і стрімко розвивається.

Понятійно-категоріальний апарат професійної діяльності у сфері кібербезпеки перебуває на етапі активного термінотворення й потребує уніфікації, оптимізації та кодифікації. Термінологічна система цієї галузі стала предметом наукових зацікавлень як лінгвістів, так і фахівців вузькоспеціалізованих сфер різних галузей знань. Власне мовних аспектів термінотворення торкалися Є. Карпіловська, І. Мислива-Бунько, Г. Шаповалова, Т. Панченко. Лінгвістичний статус слів із компонентом *кібер-* в системі мовних одиниць дослідила І. Кочан. Процеси кодифікації професійної термінології відображено в лексикографічній праці О. Копана, Є. Скулиша, у методичних напрацюваннях В. Бурячка, Г. Гулака, В. Толубка. Динаміку функціонування окремих термінолексем в науковому дискурсі дослідили Н. Яскал і В. Стьопочкіна. Детальний огляд понятійно-категоріального апарату кібернетичної безпеки з урахуванням технічної, соціально-гуманітарної і

законодавчої складових запропонував В. Мельник. О. Баранов, М. Кондратюк та В. Грохольський здійснили спробу узагальнити й систематизувати зміст і обсяг поняття «кібербезпека» в міжнародному та національному контексті. Огляд праць з питань термінології у сфері кібербезпеки засвідчив, що вітчизняні науковці не мають одностайної думки щодо змісту й обсягу основоположних понять кібербезпеки. Дослідницька активність і широкий проблемно-тематичний спектр наукових підходів визначають потребу систематизації і кодифікації термінології предметної галузі кібернетичної безпеки.

Ставимо за *мету* окреслити загальні тенденції формування термінологічної системи у сфері кібербезпеки. Для цього потрібно: 1) з'ясувати семантичну структуру поняттєвої категорії «кібербезпека»; 2) визначити місце профільної термінології в термінологічній системі української мови; 3) визначити тенденції уніфікації термінолексем. Зауважимо, що слова термін і термінолексема в межах нашого дослідження вживаються як синоніми, семантична відмінність у значенні яких визначається тим, що термін є логіко-понятійною одиницею наукового мислення, частиною терміносистеми, а термінолексема – це слово на позначення терміна, відповідно, є частиною термінолексики. Фактологічним матеріалом дослідження стали терміни в галузі кібербезпеки, що функціонують у ЗУ «Про основні засади забезпечення кібербезпеки України», «Про захист інформації в інформаційно-телекомунікаційних системах», Тлумачному словнику з основ кібербезпеки [11], у підручнику «Інформаційна та кібербезпека: соціотехнічний аспект» [3].

Щоб визначити місце термінології з кібербезпеки в термінологічній системі української мови, потрібно передусім з'ясувати концептуальне ядро й вибудувати

структурно-семантичне поле термінологеми «кібербезпека». Оскільки ця термінолексема є похідним словом, уважаємо за доцільне розкрити семантику кожної з її складових. Традиційно значення префіксоїда ***кібер-*** мовознавці визначають як усічену частину слова «кібернетика», яке в загальному потрактуванні означає «науку про загальні закономірності процесів керування та зв'язку в організованих системах» [14, Т. 4, с. 158], а з огляду на практичну сферу слововживання – це «наука про загальні закони одержання, зберігання, передачі та обробки інформації» [4, с. 539]. Проте, на нашу думку, мотивувати семантику досліджуваної лексеми твірним словом «кібернетика» помилково, зважаючи, зокрема, на визначення кібербезпеки як «захищеності життєво важливих інтересів людини і громадянина, суспільства та держави під час використання кіберпростору, за якої забезпечуються сталий розвиток інформаційного суспільства та цифрового комунікативного середовища, своєчасне виявлення, запобігання і нейтралізація реальних і потенційних загроз національній безпеці України у кіберпросторі» [7]. Зміст наведеної дефініції вказує на безпосередній зв'язок термінолексеми «кібербезпека» з поняттям кіберпростору, а тому, спростивши синтаксичну конструкцію і встановивши тема-ремні зв'язки, можемо зазначити, що кібербезпека – це забезпечення захисту в кіберпросторі. З огляду на це, для встановлення ядерної семи і побудови семантичного поля термінологеми «кібербезпека» розглянемо зміст і семантичний обсяг понять «кібернетика» й «кіберпростір».

Зважаючи на те, що кібернетика, як уже згадувалося, є наукою про управління, зв'язок і переробку інформації, то абстрактна кібернетична система являє собою невизначену кількість взаємопов'язаних об'єктів – елементів системи, здатних сприймати, зберігати й

переробляти інформацію, а також обмінюватися нею. Відтак до предметної галузі кібернетики належать усі сучасні інформаційні й телекомунікаційні технології. Важливо, що в межах кібернетичного підходу елементи системи вважаються такими, що безперервно взаємодіють між собою. Як важливі складники системи до кіберпростору належать люди – активні учасники обміну інформацією й використання інформаційних ресурсів.

Дослідники у сфері кібербезпеки пропонують дуже велику кількість дефініцій, які засвідчують, що розуміння поняття «кібербезпека» настільки різноманітне, багатогранне і не просте, що його важко формалізувати. Про це, зокрема, йдеться у статті «Визначення кібербезпеки» Д. Крейген, Н. Діакун-Тібо, Р. Пурс [2]. Автори наводять низку визначень, які функціонують в інформаційно-нормативному просторі й опираються на технічну складову кібербезпеки: 1) «Кібербезпека – це сукупність інструментів, гарантій безпеки, керівних принципів, підходів до управління ризиками, дій, найкращих практик і технологій, які можуть бути використані для захисту кіберсередовища та користувача» (MCE, 2009); 2) «Здатність захищати використання кіберпростору від кібератак» (CNSS, 2010); 3) «Сукупність технологій, процесів, практик та заходів реагування, призначених для захисту мереж, комп'ютерів, програм та даних від атак, пошкодження або несанкціонованого доступу з метою забезпечення конфіденційності, цілісності та доступності» (Громадська безпека Канади, 2014); 4) «Мистецтво забезпечення існування та безперервності інформаційного суспільства нації, гарантування та захист в кіберпросторі його інформації, активів та критичної інфраструктури» (Canongia & Mandarino, 2014); 5) «Стан захисту від злочинного або несанкціонованого використання електронних даних, або

заходи, вжиті для цього» (Oxford University Press, 2014); 6) «Діяльність або процес, здатність або стан, згідно з яким інформаційно-комунікаційні системи та інформація, що міститься в них, захищені від пошкодження, несанкціонованого використання, модифікації чи експлуатації» (DHS, 2014). Дослідники акцентують на тому, що незважаючи на вагомість технічно-технологічної складової, дослідження в галузі кібербезпеки обов'язково мають ураховувати соціальний контекст й умови, які визначають процес, за допомогою якого ключові суб'єкти приходять до спільного розуміння, як відповісти на загрозу безпеці [2].

Наведені судження ілюструють той факт, що терміну «кібербезпека» властиве широке семантичне охоплення, його визначення неоднозначні, часом малоінформативні, часткові й суб'єктивні, як-от: «Кібербезпека – це інформаційна безпека соціальних мереж, телекомунікацій, мобільних пристроїв, які можуть піддаватися хакерським атакам» [8, с. 278].

Відсутність уніфікованого підходу, на наш погляд, може бути зумовлений і неоднозначним розумінням поняття «кіберпростір». Наприклад, у підручнику «Інформаційна та кібербезпека: соціотехнічний аспект» наведено кілька потрактувань цього поняття, які використовуються в різних міжнародних нормативних документах: кіберпростір – «сфера використання електронних та електромагнітних засобів запам'ятовування, модифікації та обміну даними в мережевих системах та пов'язана з ними фізична інфраструктура»; «світовий віртуальний простір електронних даних персональних комп'ютерів»; «усі форми мережевої активності, які охоплюють контент цифрових мереж»; «вся інформаційна інфраструктура Інтернет, для якої не існує територіальних кордонів» [3,

с. 8–9]. У наведених визначеннях акцент зроблено на технологічній складовій інформаційного поля, на комп'ютерній і телекомунікаційній інфраструктурі і поза увагою залишається питання про активність людини в межах цих структур за допомогою означених технологій. Інший підхід до визначення кіберпростору фіксує таке потрактування: «середовище нефізичного існування, що утворилося внаслідок взаємодії людей, програмного забезпечення і послуг в інтернеті за допомогою технологічних пристроїв і мереж» [3, с. 8–9]. У ньому наявна чітка вказівка на зв'язок кіберпростору з ІКТ інфраструктурою, а також враховано не тільки технологічний чинник, але й діяльність людей як користувачів цифровими інформаційними ресурсами та ІКТ інфраструктурою. У такому випадку кіберпростір розглядається як тріада, що охоплює такі основні складові: 1) інформація в її цифровому поданні; 2) технічна інфраструктура, ІКТ, програмне забезпечення, за допомогою яких реалізуються основні дії з інформацією: збір, обробка, зберігання і передача; 3) інформаційна взаємодія суб'єктів з використанням одержуваної (переданої) й оброблюваної інформації за допомогою технічної інфраструктури. Тут йдеться про всі види діяльності користувачів / учасників кіберпростору, які вони здійснюють з використанням інформаційних ресурсів, потоки і сховища яких розташовуються в технічній інфраструктурі. Усі ці три аспекти враховано у визначенні терміна «кіберпростір», наведеному в ЗУ «Про основні засади забезпечення кібербезпеки України»: кіберпростір – це «середовище (віртуальний простір), яке надає можливості для здійснення комунікацій та/або реалізації суспільних відносин, утворене в результаті функціонування сумісних (з'єднаних) комунікаційних систем та забезпечення електронних комунікацій з

використанням мережі «Інтернет» та/або інших глобальних мереж передачі даних» [7].

На основі наведеної вище інформації робимо висновок про те, що мотиваційними твірними компонентами для терміна «кіберпезпека» стали основа слова *безпека* й префіксоїд *кібер-,* який ще не втратив ознак семантичної самостійності і в нашому випадку засвідчує безпосередній зв'язок з лексемою кіберпростір, що є словом-посередником між кібербезпекою та кібернетикою в загальному її розумінні. Для розуміння механізмів породження аналізованого нами терміна розглянемо його семантичну структуру. Передусім визначимо його ядерну сему, або концептуальне ядро. Ядерна сема, яка являє собою центральний компонент у семантичній структурі слова, закріплює понятійну співвіднесеність слова і визначає його основний зміст. На основі компонентно-змістового аналізу дефініцій термінолексеми «кібербезпека» ядерною семою вважаємо сигніфікативне значення лексеми «безпека», яке в загальновживаному контексті сприймається як «стан, коли кому-, чому-небудь ніщо не загрожує» [14, Т. 1, с. 137], є архісемою в семантичній структурі *кібербезпеки*. Понятійним ядром, що визначає обсяг поняття, вважаємо його денотативне значення, яке уточнюємо й поглиблюємо через атрибуцію безпеки як громадської: «забезпечення захищеності прав і свобод людини та громадянина з метою зростання загального рівня життя населення; підтримка політичної та соціально-економічної стабільності забезпечення законних прав та інтересів всіх суб'єктів держави…» [12, с. 146]. Семема «віртуальний простір», що співвідноситься із денотативним значенням префіксоїда *кібер-,* у складі термінолексеми «кібербезпека» виконує роль кваліфікативно-атрибутивної семи, визначаючи, конкретизуючи й

уточнюючи ознаки й властивості, що виокремлюють логіко-понятійний предмет серед низки подібних. Така ієрархія семантичних компонентів лексеми *кібербезпека* дозволяє побудувати лексико-семантичне поле однойменної термінологеми як ієрархічну «структурно й семантично організовану сукупність термінів, що, з одного боку, відбиває структуру знань відповідної наукової галузі, а з другого, – підпорядковується внутрішньомовним законам організації лексики» [9, с. 8]. Це уможливило групування лексеми за диференційними ознаками в різні семантичні підгрупи.

Оскільки ядро виступає інформаційним центром усього лексико-семантичного поля, до нього віднесли родові поняття, між якими встановлюються парадигматичні відношення. Отже, ядро утворюють терміни, у яких сема співвіднесення із галуззю кібербезпеки є основною: *безпека, інформаційна безпека, кібербезпека.* Усі ці елементи у своїй структурі мають родову аріхсему *«відсутність загрози»*, навколо якої і розгортається польова структура. Варто зауважити, що ці термінолексеми перебувають в гіперо-гіпонімічних відношеннях. Лексема *безпека* є гіперонімом щодо *інформаційної безпеки* і водночас *кібербезпека* – гіпонімом щодо *інформаційної безпеки*. Зважаючи на те, що інформаційна безпека – це «стан захищеності людини, суспільства та держави від реальних і потенційних, внутрішніх і зовнішніх загроз у інформаційному просторі, що спрямовані на порушення безпеки інформації й інформаційно-психологічної безпеки» [10, с. 190], а поняття «‘кібербезпека’ з технологічної точки зору є, безумовно, складовою частиною поняття ‘інформаційна безпека’, оскільки розглядаються ті ж самі загрози, методи, засоби і заходи захисту, реалізація яких обмежується лише технологіями кіберпростору» [10,

с. 194], то інтеграційною семою для них є «стан захищеності», «захищеність». Водночас семантичний обсяг поняття *кібербезпека* звужується за рахунок диференційної семи «віртуальний простір».

Кожен із складників семантичного ядра здатний організовувати мікрополя, які утворюють центр поля, що охоплює одиниці, об'єднані інтегральним значенням. Так, семантика *кібербезпеки* конкретизується в таких термінолексемах, як: *кіберзахист, кібероборона, кіберзагроза, кіберзлочин, кіберзлочинність, кібертероризм.* У семантиці кожного з елементів наявна спільна сема, мотивована значенням *«віртуальний простір».* Аналогічно, загальна семантика *інформації* актуалізується в таких лексемах, як: *інформаційна система, інформаційні потоки, інформаційні ресурси, національні електронні інформаційні ресурси, інформаційна інфраструктура, критична інформаційна інфраструктура, об'єкти інфраструктури.*

Семантичне ядро і мікрополя центральної периферії перебувають у вертикальних ієрархічних зв'язках, а саме в родо-видових відношеннях. Наприклад, зміст терміна *інформаційна безпека* як «стан захищеності інформаційного простору держави, за якого неможливо завдати збитку властивостям об'єкта безпеки, що стосуються інформації та інформаційної інфраструктури, і який гарантує безперешкодне формування, використання й розвиток національної інфосфери в інтересах оборони» [3, с. 12], передається через поняття *інформаційний простір*, *інформаційну структуру*, *інформаційну інфраструктуру, інфосферу,* у яких спостерігається звуження й конкретизація семантики родового слова, яке стало терміноелементом наведених словосполучень. Аналогічно члени мікрополів можуть утворювати ієрархічні семантичні відношення, члени яких утворюють

ближню периферію лексико-семантичного поля «Кібербезпека». Як приклад наявності системних зв'язків в межах ближньої периферії можна навести типологію кіберконфліктів, про яку йдеться в матеріалі Д. Дубова [6, с. 27]. Так, мікрополе «кіберконфлікт» охоплює такі гіпоніми: *кібервандалізм, інтернет-злочин, кібершпигунство, кібертероризм, кібервійна.* Усі компоненти наведеного мікрополя об'єднані інтеграційною ознакою, зафіксованою в архісемі ядерної лексеми – конфлікт, як «зіткнення інтересів» [14, Т. 4, с. 274]. За ступенем реалізації цього значення члени мікрополя утворюють ланцюг інтенсивності ознаки, мотивованої денотатом твірних елементів термінів: *інтернет-злочин* (суспільно небезпечна дія), *кібершпигунство* (злочинна діяльність), *кібервійна* (організована боротьба), *кібервандалізм* (нещадне руйнування), *кібертероризм* (найгостріша форма боротьби). Додавання до основ загальновживаних слів префіксоїда *кібер-* активізувало процеси термінологізації, внаслідок чого відбулися процеси звуження семантики лексичної одиниці: «кібертероризм – терористична діяльність, *що здійснюється у кіберпросторі або з його використанням*»; «кіберзлочин (комп'ютерний злочин) – суспільно небезпечне винне діяння *у кіберпросторі та/або з його використанням*, відповідальність за яке передбачена законом України про кримінальну відповідальність та/або яке визнано злочином міжнародними договорами України» [7].

Якщо ближня периферія – це площина функціонування спеціалізованих термінів у сфері кібербезпеки і термінології предметної галузі знань «Інформаційні технології» (*телекомунікаційна мережа, електронні інформаційні ресурси, інформаційна інфраструктура, комунікаційні системи, інформаційні*

потоки), то дальня периферія аналізованого термінологічного поля представлена міжгалузевими, технічними, загальнонауковими термінами: *політика безпеки, конфіденційність інформації, доступність інформації* тощо. Додавання до термінологічної сполуки видового елемента може перевести його із дальньої периферії в коло ближньої: порівняймо, *доступність інформації* – це «можливість використання інформації, коли в цьому постає потреба», а *доступність інформації в ІС* – це «можливість використання інформації її користувачем або програмою відповідно до встановлених правил» [3, с. 13].

Така лексико-понятійна співвіднесеність термінів, їх семантична ієрархія – взаємодія і взаємозалежність – у межах окресленого лексико-семантичного поля дозволяють вважати термінологію сфери кібербезпеки як цілісну термінологічну систему. Аналіз семантики складників термінологічного поля свідчить про те, що за кожною термінолексемою закріплене своє місце на різних структурних рівнях, їй притаманне індивідуальне значення, яке за змістом споріднене, але не однакове зі значенням інших елементів системи.

Традиційно термінологія сучасної української мови розподілена за сферами відповідно до наявних у суспільстві знань, й представлена такими групами: 1) наукова; 2) суспільно-політична; 3) суспільно-економічна; 4) юридична [15]. Оскільки в окресленому нами лексико-семантичному полі наявні терміни з різних галузей знань, то можемо говорити, що термінологія сфери кібербезпеки має міждисциплінарний характер. На мультидисциплінарному контексті кібербезпеки наголошується і в праці К. Вішика, М. Мацубари, А. Плонка «Ключові поняття в кібербезпеці» [1]. Мультидисциплінарність зумовлюється як понятійною

структурою кібербезпеки, так і понятійною структурою кіберпростору, у якій, як наголошувалося раніше, наявні три компоненти: цифрова інформація; технічна інфраструктура; інформаційна взаємодія суб'єктів. Мультидисциплінарність термінології визначається й тим, що «тезаурус кібербезпеки інтегрований з поняттями безпеки інформації та інформаційно-психологічної безпеки в законодавчому, технологічному та правоохоронному сенсах» [10, с. 192]. Крім того, структура окресленого нами термінологічного поля передає й комплексну сутність кібербезпеки, яка, за В. Бурячком, охоплює соціальний, технічний, інформаційний, комунікаційний аспекти захисту суспільства, що спрямований відповідно на зовнішньополітичну, внутрішньополітичну, воєнну, економічну, соціальну, науково-технічну сфери громадського життя [3, с. 16].

Огляд праць з питань змісту й обсягу ключових понять у сфері кібербезпеки [6, 10–13] засвідчив, що значна увага на сучасному етапі розвитку цієї галузі приділяється впорядкуванню профільної термінології, її уніфікації з подальшою можливістю кодифікації, тобто оформлення її як нормативного термінологічного словника. Уніфікація термінів, а саме зведення кількох варіантів терміна до одного оптимального, необхідна для забезпечення єдиного розуміння сутності об'єктів і процесів з метою досягнення максимального ефекту.

Одним із етапів упорядкування є нормалізація термінів. З огляду на це, термін повинен задовольняти потребу в адекватному відображенні об'єктів, процесів і ознак в спеціальних сферах. Передусім розглянемо, наскільки термінолексема «кібербезпека» відповідає вимогам, що висуваються до терміна як спеціалізованого позначення об'єктів наукового чи професійного знання.

Зміст терміна «кібербезпека», який детально проаналізований вище, відповідає загальному розумінню цього класу лінгвістичних одиниць: «спеціальне слово, словосполучення, яке слугує для вираження поняття певної галузі знань; для розкриття свого значення вимагає дефініції» [15, с. 13]. З'ясуємо, наскільки термінолексема «кіберпезпека» задовольняє вимоги, що висуваються до термінів: 1) точність – термін у плані змісту відповідає понятійній системі професійної галузі кібербезпеки, охоплює інформаційну, технологічну й персональну складові безпеки в кіберпросторі; 2) наявність дефініції – визначення закріплено в нормативно-правовому контенті, дефініція відповідає таким вимогам до неї: ємна, несуперечлива, містить мінімальну кількість ознак, достатніх для ідентифікації позначуваного поняття, не має синонімів в межах термінологічної системи; 3) системність – термінолексема утворює ядро термінологічного поля, вступає в парадигматичні відношення з іншими членами термінологічного поля, системність якого була доведена нами; 4) тенденція до однозначності – проведений структурно-семантичний аналіз засвідчив моносемантичність терміна, відсутність категоріальної багатозначності; 5) семантика терміна є вужчою порівняно із семантикою мотиваційних загальновживаних лексем як елементів його словотворчої структури; 6) деривaційний потенціал – на основі терміна утворюються похідні слова, загальновживані, нетермінологізовані, про що свідчить практика слововживання: *кібербезпечні виміри, кібербезпечні рішення, кібербезпечне функціонування підприємств, кібербезпечні продукти, кібербезпечна тема, кібербезпечна Україна, кібербезпечна країна, кібербезпечний марафон*. Наведені деривати здатні передавати атрибутивні відношення (*кібербезпечна тема*)

та атрибутивно-об'єктні (*кіберпезпечні виміри – виміри кібрбезпеки*); 7) мовна правильність – термін відповідає нормам української мови у частині правопису, словотворення, словозміни і слововживання; 8) нейтральність – емоційно-експресивне забарвлення відсутнє як наслідок виконання термінолексемою лише номінативної функції та відсутності в її семантичній структурі конотативного компонента. Отже, можемо констатувати, що термін «кібербезпека» відповідає вимогам і може бути кодифікованим. Проте є терміни, які задовольняють не всі названі критерії. Наприклад, *кіберзлочинність* тлумачиться як «сукупність кіберзлочинів» [7]. Детермінація поняття через зміст іншого терміна, на нашу думку є неефективною, оскільки порушено вимогу точності, що посилюється абстрактним змістом як самого терміна (утворено від абстрактного іменника злочинність), так і кваліфікативно-атрибутивною лексемою *сукупність*, змістом якої є «неподільна єдність чого-небудь; загальна кількість, сума чогось» [14, Т. 9, с. 832], характеризується семантичною невизначеністю. І хоча сама дефініція є ємною і несуперечливою, проте не містить достатньої кількості ознак для точної ідентифікації позначуваного поняття. Тому кіберзлочинність вважаємо не терміном а дериватом терміна *кіберзлочин (кіберзлочин, кіберзлочинний)*, здатність якого до деривації засвідчує його нормативність. У науковому дискурсі спостерігаємо й іншу дефініцію цього терміноїда: «кіберзлочинність – незаконні дії, які здійснюються людьми, що використовують інформаційні технології для злочинних цілей» [13, с. 33]. Цей факт свідчить про те, що не встановлено єдиної дефініції, а сама термінолексема потребує оптимізації та уніфікації. Актуальним, на нашу думку, питання, що ілюструє динамічні процеси термінотворення, є належність до

термінологічної системи кібербезпеки терміносполуки *система управління технологічними процесами* (*технологічна система)*, що визначається як «автоматизована або автоматична система, яка є сукупністю обладнання, засобів, комплексів та систем обробки, передачі та приймання, призначена для організаційного управління та/або управління технологічними процесами (включаючи промислове, електронне, комунікаційне обладнання, інші технічні та технологічні засоби) незалежно від наявності доступу системи до мережі «Інтернет» та/або інших глобальних мереж передачі даних» [7]. У плані змісту цей термін відповідає понятійній системі предметної галузі знань «Інформаційні технології», а отже, порушено критерій точності. Проте, якщо врахувати гіперо-гіпонімічні відношення між поняттями *безпека, інформаційна безпека, кібербезпека*, що утворюють ядро аналізованого термінологічного поля, а також їхню здатність утворювати власні мікрополя, то термінологічне словосполучення *технологічна система* знаходить своє місце на периферії термінологічного поля, куди можуть потрапляти терміни із суміжних галузей знань. Таким чином, точність цього терміна не викликає сумнівів. Проте запропонована синонімічна назва, у якій відсутні кваліфікативно-атрибутивні ознаки, що звужують семантику загального значення поняття, ускладнює процес функціонування цього терміна в різних комунікативно-дискурсивних практиках. Наведені приклади засвідчують активні процеси у формуванні термінологічної системи, її оптимізації та нормування.

Висновки

Зміст і структура термінологічного поля «кібербезпека» визначається категоріально-поняттєвою структурою однойменного терміна. Утворений від двох

основ (усіченої кібер- та безпека), він визначає зміст і обсяг концептуального ядра термінологічного поля, яке утворюється на основі гіперо-гіпонімічних відношень таких якого складників, як безпека, інформаційна безпека, кібербезпека, кожен із яких зданий утворювати на основі інтегральної семи лексико-семантичні мікрополя, елементи яких послідовно на кожному ієрархічному рівні звужують обсяг термінологізованого поняття за рахунок диференційних ознак. Така вертикальна лексико-понятійна ієрархія засвідчує, що термінологія сфери кібербезпеки є цілісною терміносистемою: за кожною термінолексемою закріплене своє місце на різних структурно-семантичних рівнях, їй притаманне індивідуальне значення, яке за змістом споріднене, але не однакове зі значенням інших елементів системи. Концептуальне ядро, центр термінологічного поля утворюють терміни, денотативне значення яких безпосередньо відповідає понятійній системі кібербезпеки. Ближня периферія – це площина спеціалізованих термінів у сфері кібербезпеки і термінології предметної галузі знань «Інформаційні технології». Дальню периферію утворюють міжгалузеві, технічні й загальнонаукові терміни. Така структура поля та логіко-поняттєва ієрархія термінів вказує на мільтидисциплінарний характер кібербезпеки.

Термінологія сфери кіббербезпеки є відкритою, змінюваною системою. У ній активно відбуваються процеси оптимізації, уніфікації та нормалізації. Тому

перспективним напрямком вважаємо розроблення єдиної термінології у сфері кібербезпеки, гармонізованої із уже існуючою, унормованою термінологічною системою предметної галузі «Інформаційні технології».

СПИСОК ЛІТЕРАТУРИ

1. Claire Vishik, Mihoko Matsubara, Audrey Plonk. Key Concepts in Cyber Security: Towards a Common Policy and Technology Context for Cyber Security Norms. International Cyber Norms: Legal, Policy & Industry Perspectives, NATO CCD COE Publications. – Tallinn, 2016. – P. 221–242.

2. Craigen Dan, Diakun-Thibault Nadia, Purse Randy. Defining Cybersecurity. Technology Innovation Management Review, 2014. [Електронний ресурс]. − Режим доступу : https://www.researchgate.net/publication/267631801_Defining_Cybersecurity

3. Інформаційна та кібербезпека: соціотехнічний аспект : підручник / В. Л. Бурячок, В. Б. Толубко, В. О. Хорошко, С. В. Толюпа. – Київ : ДУТ, 2015. – 288 с.

4. Великий тлумачний словник сучасної української мови (з дод. і допов.) / уклад. і голов. ред. В. Т. Бусел. – Київ, Ірпінь : ВТФ «Перун», 2005. – 1728 с.

5. Гладківська О. В. Вимоги до нормативно-правової термінології / О. В. Гладківська // Інформація і право. – 2015. – № 1. – С. 55–62.

6. Дубов Д. Підходи до формування тезаурусу у сфері кібербезпеки / Д. Дубов // Політичний менеджмент. – 2010. – № 5. – С. 19–30.

7. Про основні засади забезпечення кібербезпеки України : Закон України від 05.10.2017 № 2163-VIII [Електронний ресурс]. − Режим доступу : https://zakon.rada.gov.ua/laws/show/2163-19#Text.

8. Качан І. Слова з компонентом кібер- у сучасній українській мові / І. Качан // Вісник Львівського університету. Серія філологічна. – 2016. – Вип. 63. – С. 277–285.

9. Книшенко Н. П. Поняття «лексико-семантичне поле» й «термінологічне поле» в сучасному мовознавстві /

Н. П. Книшенко // Лінгвістичні дослідження : зб. наук. праць ХНПУ ім. Г. С. Сковороди. – 2020. – Вип. 52. – С. 1–9.

10. Мельник С. В. Понятійно-категоріальний апарат у системі професійної підготовки майбутніх фахівців з кібербезпеки / С. В. Мельник // Інформаційні технології і засоби навчання. – 2016. – Т. 55, вип. 5. – С. 187–197.

11. Основи інформаційної безпеки. Тлумачний словник. Методичні вказівки для самостійної роботи студентів. – Харків : ХНТУСГ, 2017. – 42 с.

12. Піх Н. С. Забезпечення громадської (публічної) безпеки у країнах Європейського Союзу / Н. С. Піх // Інвестиції: практика та досвід. – 2020. – № 3. – С. 143–146.

13. Пфо О. М. Основні поняття і класифікація кіберзлочинності / О. М. Пфо // Актуальні задачі та досягнення у галузі кібербезпеки : матеріали Всеукраїнської науково-практичної конференції. – Кропивницький : КНТУ, 2016. – С. 33–34.

14. Словник української мови : в 11 т. / АН УРСР. Інститут мовознавства ; за ред. І. К. Білодіда. – Київ : Наукова думка, 1970–1980.

15. Томіленко Л. М. Термінологічна лексика в сучасній тлумачній лексикографії української літературної мови : монографія / Л. М. Томіленко. – Івано-Франківськ : Фоліант, 2015. – 160 с.

РОЗДІЛ 13
РЕКОМЕНДАЦІЇ ЩОДО РОЗРОБЛЕННЯ НАЦІОНАЛЬНОЇ МЕТОДОЛОГІЇ ОЦІНЮВАННЯ КІБЕРЗАХИЩЕНОСТІ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМ

В. В. Кальченко

Вступ

Конституція України проголошує, що забезпечення інформаційної безпеки є найважливішою функцією держави [1]. З точки зору комп'ютерних мереж це досягається за рахунок властивості інформаційної системи впродовж установленого часу протистояти несанкціонованому одержанню, модифікації та унеможливленню блокування інформації, тобто підтримання системою трьох основних властивостей інформації: конфіденційності, цілісності й доступності. На цей час в Україні гостро стоїть питання створення систем кібербезпеки, що будуть забезпечувати протидію кіберзагрозам та кібератакам, автоматичний вибір параметрів функціонування інформаційних систем і мереж зв'язку в умовах деструктивних впливів.

Протистояння держав вимагає проведення розвідувальних, диверсійних операцій у кіберпросторі, що також передбачає цифрове проникнення в мережі й системи управління потенційного супротивника з використанням тактичних і технічних прийомів різного ступеня складності. Для мінімізації і попередження деструктивного впливу кіберзагроз проводять роботи зі створення систем захисту. Але водночас з'являється проблема оцінювання того, наскільки якісно побудована система захисту та наскільки якісно вона може протидіяти сучасним загрозам. Сфера кібербезпеки є такою, що

постійно розвивається і, як наслідок, потребує від установ усіх форм власності та технічних спеціалістів аналізу актуальних кіберзагроз, вжиття заходів з упровадження нових засобів захисту, проведення постійних перевірок захищеності, проведення навчання зі звичайними користувачами.

Огляд законодавства України у сфері проведення оцінювання захищеності інформації комп'ютерних систем

В Україні основним методом оцінювання захищеності комп'ютерних систем державних органів, органів місцевого самоврядування, підприємств, установ, організацій є проведення перевірок стану технічного захисту інформації. Цей вид перевірок було створено в рамках системи технічного захисту інформації. У зв'язку з розвитком галузі кібербезпеки відбувається відповідний розвиток нормативної бази. У новітніх нормативних актах із кібербезпеки встановлені вимоги з побудови комплексної системи захисту інформації, що є наріжним каменем системи технічного захисту інформації. На сьогодні перевірка кіберзахищеності є перевіркою технічного захисту інформації.

Основними документами, що регулюють питання, пов'язані із захистом інформації в інформаційно-комунікаційних системах, є закон України [2] і Постанова Кабінету Міністрів України [3]. Відповідно до зазначених нормативних документів основною вимогою захищеності комп'ютерних систем є наявність побудованої комплексної системи захисту інформації (КСЗІ) з підтвердженою відповідністю. Необхідність побудови КСЗІ визначається видом інформації, що циркулює в системі. Згідно із законодавством України такою інформацією є секретна, службова, конфіденційна,

персональні дані та відкрита інформація, що належить до державних інформаційних ресурсів. Порядок побудови та вимоги до КСЗІ визначені нормативними документами системи технічного захисту інформації (ТЗІ). Основним регулятором у сфері ТЗІ й кібербезпеки є Державна служба спеціального зв'язку та захисту інформації України (Держспецзв'язку) [8], яка встановлює вимоги в зазначених сферах і проводить перевірки стану ТЗІ.

Перевірки стану ТЗІ здійснюють посадові особи регіональних управлінь Держспецзв'язку. Основним документом, яким керуються ці співробітники, є Положення про державний контроль за станом ТЗІ [4]. Цей документ визначає національну процедуру оцінювання захищеності інформації в державних органах, що складається з шести етапів:

1. Вивчення загальних питань, а саме: з'ясування юридичної назви установи, адреси, організаційно-штатної структури, основних видів діяльності, умов розміщення, пропускного режиму, наявності систем захисту тощо.

2. Аналіз заходів із захисту мовної інформації з обмеженим доступом (ІзОД), необхідність захисту якої встановлено законодавством та розголошення якої заборонене.

3. Аналіз заходів із захисту інформації, що циркулює в комп'ютерних системах та інших пристроях оброблення інформації.

4. Аналіз заходів із захисту інформації під час створення продукції або технологій для державних потреб, при виконанні науково-дослідних, дослідно-конструкторських робіт, що фінансуються з державного бюджету.

5. Аналіз заходів із захисту інформації під час проєктування, будівництва, реконструкції або капітального ремонту об'єктів інформаційної діяльності.

6. Аналіз заходів із технічного захисту ІзОД під час приймання іноземних делегацій.

З точки зору логіки процесу, питання оцінювання захищеності інформації розглядають у процесі проведення третього етапу, а саме під час аналізу заходів із ТЗІ. На цьому етапі:

− визначаються комп'ютерні системи, в яких циркулює інформація, що підлягає захисту відповідно до вимог законодавства України;

− визначають комп'ютерні системи з побудованими КСЗІ;

− визначають осіб, які відповідають за захист інформації в комп'ютерних системах;

− перевіряють технічну та експлуатаційну документацію на КСЗІ і її відповідність нормативним документам;

− перевіряють відповідність даних, зазначених у технічний та експлуатаційній документації на КСЗІ з реальними умовами функціонування комп'ютерних систем і системи безпеки установи в цілому;

− визначають програмні засоби, які використовують для оброблення інформації, що потребує захисту;

− перевіряють працездатність комплексу засобів захисту, що використовуються в цільовій системі та його конфігурацію;

− перевіряють наявність антивірусного програмного захисту і періодичність оновлення антивірусних баз.

З практичної точки зору зазвичай як засоби захисту для комп'ютерної системи використовують сервіси безпеки операційної системи Windows. На серверах це операційна система Windows Server, а на робочих станціях це Windows Professional/Enterprise. Під час проведення

перевірок здійснюють порівняння налаштувань серверів та робочих станцій із вимогами, висунутими в інструкціях зі складу документації на КСЗІ. Далі перевіряють виконання вимог експертного висновку на КСЗІ та реальних умов експлуатації комп'ютерної системи. За результатами проведених робіт складають звіт, де зазначають повноту та достатність заходів із ТЗІ і відповідність КСЗІ вимогам нормативно-правових актів. Перевірку параметрів безпеки здійснюють відповідно до інструкцій (настанов), розроблених організацією, що будувала КСЗІ. Водночас у разі відсутності систем захисту параметри безпеки не перевіряють через відсутність еталонного документа, з яким необхідно порівнювати поточні налаштування системи.

За результатами перевірки посадові особи Держспецзв'язку складають акт із рекомендаціями стосовно приведення стану ТЗІ у відповідність до вимог законодавства України. Здебільшого такими рекомендаціями є створення КСЗІ та розроблення документів, необхідних для побудови КСЗІ:

− наказів про створення комісій із проведення категоріювання, обстеження умов функціонування системи;

− моделей загроз та порушника безпеки інформації;

− політики безпеки;

− плану захисту інформації;

− технічного завдання на створення КСЗІ.

Проте рекомендації не містять вказівок щодо налаштувань параметрів безпеки операційних систем, конфігурування параметрів мережевого обладнання, що не дозволяє підвищити реальний ступінь захищеності системи, яку перевіряли. Можливим варіантом вирішення проблеми реального оцінювання захищеності інформації (з технічної точки зору) у відповідних інформаційно-

комунікаційних системах є проведення тестування на проникнення.

Тестування на проникнення – це контрольовані атаки як на систему в цілому, так і на її складові частини або окремі технічні елементи (сервери, комп'ютери, мережеве обладнання). Під час тестування фахівець (група фахівців) намагається одержати доступ до інформації, що зберігається (обробляється) в цільовій системі, отримати контроль над функціонуванням системи або призвести до неможливості виконання заданих функцій. Водночас тестувальник використовує набір програмних, апаратних засобів, а також ті самі методи роботи, що й зловмисники. Під час своєї роботи тестувальник знаходить найбільш уразливі місця в системі, з'ясовує можливі варіанти їх використання, фіксує їх у звіті і дає технічні рекомендації для підвищення рівня захищеності. Процес тестування максимально повторює процес злому, що проводиться зловмисниками.

В Україні відповідно до Наказу Адміністрації Держспецзв'язку від 02.12.2014 № 660 [5] проводять перевірки фактичного стану безпеки інформаційно-комунікаційних систем, які фактично є тестуваннями на проникнення. Цей вид перевірки в інтересах державних органів здійснює один підрозділ Держспецзв'язку – CERT-UA (Computer Emergency Response Team of Ukraine – Команда реагування на комп'ютерні надзвичайні події України). Аналізуючи наявні у відкритому доступі нормативні акти, розпорядження, можна виділити дві основні проблеми існуючої методики проведення перевірок стану захищеності інформаційних систем в Україні та системи ТЗІ в цілому [6]:

− відсутність методології проведення тестування на проникнення. Виходячи зі змісту і вимог «Порядку оцінки

стану захищеності державних інформаційних ресурсів в інформаційно-телекомунікаційних системах» [5], оцінювання проводять посадові особи Держспецзв'язку на підставі їх знань, умінь та суб'єктивних суджень;

− побудовою КСЗІ в державних органах здебільшого займаються приватні суб'єкти господарювання, які мають обмежену кількість співробітників. Унаслідок цього, а також недосконалості НД ТЗІ, один або декілька співробітників фірм-ліцензіатів ухвалюють одноосібне рішення щодо налаштування параметрів безпеки інформаційно-комунікаційних систем.

Перевірка стану ТЗІ передбачає лише порівняння фактичних налаштувань з інструкціями на КСЗІ. Тому з правової точки зору інспекторам Держспецзв'язку достатньо складно опротестувати конфігурацію параметрів системи захисту.

Цілі кібербезпеки та огляд міжнародних методологій тестувань на проникнення

Для вирішення вищезазначених проблем було розглянуто цілі кібербезпеки. Для розуміння цих цілей беруть до уваги дві моделі: CIA (confidentiality, integrity, availability) та DAD (disclosure, alteration, denial) [7] .

Фахівці, які займаються забезпеченням кібербезпеки, розглядають модель CIA, що в українських нормативних документах має назву тріада КЦД: конфіденційність, цілісність та доступність. Зловмисники і фахівці, які проводять оцінювання захищеності, розглядають другу модель для досягнення своїх цілей, а саме DAD: розголошення, зміна, відмова. Таким чином, з одного боку, є підхід до забезпечення трьох основних властивостей безпеки інформації, з іншого боку, – три основні методи їх порушення.

Цілі, яких намагаються досягти відповідні суб'єкти, наведені в таблиці 13.1.

Таблиця 13.1 – Цілі кібербезпеки з точки зору моделей CIA та DAD

Модель CIA	Модель DAD
Заходи спрямовані на запобігання несанкціонованому доступу до інформації або систем	Заходи спрямовані на отримання несанкціонованого доступу до інформації або системи
Заходи спрямовані на запобігання несанкціонованій зміні інформації чи налаштуванню систем	Заходи спрямовані на проведення несанкціонованих змін інформації чи налаштувань систем
Заходи спрямовані на забезпечення санкціонованому доступу до інформації чи системи	Заходи спрямовані на запобігання санкціонованому використанню інформації чи системи

Розглядаючи тестування на проникнення, всі дії необхідно розглядати з точки зору моделі DAD.

Метою створення кожної методології є структурування процесу тестування з метою його найбільш якісного проведення. Аналізуючи міжнародний досвід, варто виділити такі методології: OSSTMM, NIST Special Publication 800-115, PTES, ISSAF, CompTIA. Крім того, для проведення тестування на проникнення можна використовувати модель Cyber-Kill Chain від корпорації Lockheed Martin. Кожна з цих методологій має різну

кількість фаз та відрізняється між собою способом структурування діяльності тестувальника.

Розглянемо зазначені методології з точки зору виконуваних дій. Методологія NIST Special Publication 800-115 виділяє чотири фази: планування, дослідження (збирання інформації та аналіз уразливостей), атака, звітність.

PTES виділяє сім фаз: першочергове спілкування, збирання інформації, моделювання загроз, аналіз уразливостей, експлуатація, пост експлуатація, звітність.

ISSAF виділяє вісім фаз: планування та підготовку, оцінювання системи (збирання інформації про систему, ідентифікація уразливостей), тестування на проникнення, одержання доступу або розширення привілеїв, проведення додаткових тестів, компрометація віддалених користувачів та інформаційних ресурсів, підтримання несанкціонованого доступу до мережі, приховування слідів роботи.

CompTIA виділяє чотири фази: планування і визначення масштабу, збирання інформації та визначення уразливостей, проведення атак й експлуатація уразливостей, звітування та повідомлення результатів.

Cyber-Kill Chain виділяє сім фаз: розвідку, озброєння, доставку, експлуатацію, встановлення, створення каналу керування, дію на об’єкті.

Методологія OSSTM в явному вигляді не виділяє фази тестування, проте містить такі розділи, які відображають логіку процесу тестування, а саме:

− визначення рамок тестування, ролей та процесів;

− аналіз безпеки об’єкта;

− аналіз соціальних процесів у персоналі об’єкта тестування;

− тестування стійкості персоналу до соціальної інженерії;

− тестування безпеки фізичної інфраструктури; тестування безпеки безпровідних технологій;

− тестування безпеки телекомунікаційних технологій;

− тестування безпеки даних;

− підготовка звітності про тестування.

В узагальненому випадку процедура тестування зводиться до виконання трьох основних фаз: розвідки, виконання проникнення в цільову мережу, складання звітного матеріалу. Кожна із зазначених методологій має ряд своїх переваг та недоліків [12].

На цей час в Україні на державному рівні нормативно не закріплена методологія проведення тестування на проникнення (крім банківської сфери, де проведення тестування на проникнення є необхідною умовою для здійснення операцій із міжнародними платіжними системами). Тому постає питання розроблення національної методології, що буде враховувати українське законодавство та сучасні реалії.

Восьмирівнева методологія проведення тестування на проникнення

Як зазначалося раніше, завданнями методології є описання та структурування процесу тестування на проникнення. З одного боку, вона повинна чітко визначати етапи проведення, з іншого − в ній у загальному вигляді повинні зазначатися підходи до проведення тестування. Усе це дозволяє скоротити час, необхідний для проведення тестування, максимально повно провести пошук слабких місць у системі, скласти найбільш інформативний звіт для усунення виявлених недоліків.

Проаналізувавши існуючі методології та стандарти, пропонується така методологія проведення тестування на проникнення, що складається з восьми фаз (етапів):

1. Одержання первісної інформації від замовника про систему, бажані методи тестування, цілі та верифікація даних.

2. Збирання інформації про цільову систему з відкритих джерел.

3. Сканування та перелічення сервісів, визначення уразливостей.

4. Одержання доступу до системи та підвищення привілеїв у системі.

5. Досягнення цілей тестування.

6. Забезпечення безперебійного доступу до системи.

7. Складання звіту з рекомендаціями.

8. Демонстрація виявлених уразливостей та видалення слідів наявності в системі.

Кожна фаза повинна складатися з відповідних модулів. Це дозволить раціонально планувати та виконувати тестування виходячи з цілей та завдань, які висуває замовник. Ураховуючи вимоги замовника, можна виключати деякі модулі з процесу проведення тестування. Також у майбутньому це дозволить доповнювати методологію.

Під час тестування на проникнення ключовим моментом є те, що під час цього тестування відбувається не лише пошук усіх відомих уразливостей у системі, а й процес визначення найбільш негативних наслідків, до яких може призвести використання зловмисниками цих уразливостей.

На першому етапі тестування підписується договір, у якому визначають юридичні аспекти дій, які будуть проводитись із системою, що тестується. Під час складання складанні договору на проведення тестування повинна бути визначена мета проведення такої роботи. Метою тестування може бути перевірка таких питань:

− що трапиться із системою, якщо внутрішній користувач буде скомпрометований;

− що відбудеться, якщо організація зазнала цілеспрямованої атаки з боку зловмисної сторони;

− яких можливостей набувають зловмисники після злому бездротових точок доступу;

− які негативні наслідки можуть бути в разі злому того чи іншого сервера;

− які наслідки будуть після проведення DDoS-атаки на основні технічні засоби комп'ютерної системи.

З метою формалізації процесу тестування необхідно в договорі визначити та документально зафіксувати його основні правила проведення:

− з якими системами дозволено працювати тестувальникам, щоб не вивести з ладу критичні системи;

− дата початку та дата кінця проведення тестування;

− в які часові проміжки дозволено проводити роботи;

− які техніки дозволено використовувати під час проведенні робіт (DDoS-атаки, ARP-spoofing, fishing тощо)

− порядок використання виявлених уразливостей. Якщо використання не дозволене, повинно бути визначено, яким чином буде здійснюватись інформування посадових осіб, пошук наступних уразливостей тощо;

− який порядок дій тестувальника в разі виявлення істотної проблеми з безпеки, яким чином її вирішувати;

− який порядок дій у разі невідкладних проблем та порядок оповіщення про це;

− яким чином запам'ятовувальні пристрої зі складу обладнання тестувальників будуть передані установі-замовнику (для тих випадків, коли секретна або

конфіденційна інформація замовника буде скопійована на носії виконавця робіт);

− яким чином буде відбуватися передавання звіту за результатами проведених робіт.

Крім того, необхідно підписати договір про нерозголошення відомостей, що стануть відомі тестувальникам під час проведення робіт.

З метою виконання першого етапу методології у договорі на надання послуг визначити параметри проведення тестування. Як зазначалося в праці [6], методи проведення тестування на проникнення можна класифікувати за шістьма ознаками (параметрами):

1. За розміщенням тестувальника щодо периметра організації замовника.
2. За обізнаністю тестувальника.
3. За обізнаністю працівників установи-замовника.
4. За характером дій, що будуть вживатися.
5. За повнотою тестування.
6. За технікою проведення.

З метою унеможливлення притягнення тестувальників до відповідальності за злам чи руйнування системи, необхідно максимально повно прописати сценарії їх можливих дій.

Після одержання початкової технічної інформації варто її перевірити на предмет достовірності. Перед початком проведення робіт необхідно точно встановити, що система, яка визначена в договорі, дійсно належить установі-замовнику. Наприклад, може з’ясуватися, що система знаходиться у хмарному середовищі, або що деякі з наданих IP-адрес були передані іншим установам.

Крім того, на цьому етапі повинно бути прописано, що буде вважатись критерієм ефективності проведеного тестування, після якого можна позачергово зупинити тест.

На другому етапі тестування відбувається збирання максимально можливого переліку відомостей про цільову систему. Цей етап можна поділити на дві частини: пасивний та активний збір інформації.

За пасивного збору відбувається пошук інформації про цільову систему з відкритих джерел, таких як корпоративний вебсайт, сайти газет і журналів, де публікуються дані про компанію замовника, інтерв'ю з її працівниками, вивчення роздруківок, які були викинуті на смітник, здійснення дзвінків від імені служби підтримки для з'ясування технічних даних у працівників, вивчення вимог, які висуваються до кандидатів на зайняття вакантних посад, резюме працівників компанії замовника. Практичними прикладами таких сайтів можуть бути: пошукові системи Google, Bing, соціальні мережі Facebook, LinkedIn. Залежно від цілей тестування перелік відомостей, який необхідно одержати, може бути різним. У загальному випадку пропонується збирати таку інформацію: перелік IP-адрес цільової системи, корпоративні адреси електронної пошти, номери телефонів співробітників компанії замовника, інформацію про операційні системи та програмне забезпечення, яке використовується в комп'ютерній мережі тощо.

На третьому етапі тестування потрібно просканувати систему та визначити перелік наявних у системі сервісів, додатків, версій програмного забезпечення та наявних у них уразливостей.

Уразливості можна поділити на такі категорії:

− неправильні конфігурації параметрів безпеки, що можуть бути використані зловмисниками;

− переповнення буфера, що дозволяє виконати зловмисний код з адміністративними правами або призвести до краху системи;

− недоліки ядра операційної системи, що може призвести до компроментації всієї системи захисту;

− символічні посилання, які можуть призвести до запуску зловмисних програм із правами системи або правами адміністратора;

− неправильне налаштування атрибутів доступу, внаслідок чого можливе несанкціоноване ознайомлення з чутливою інформацією та її подальшою модифікацією: файлами з конфіденційною інформацією, паролями, переліком довірених пристроїв тощо;

− відсутність перевірки вхідних даних, що може призвести до неправильної роботи відповідних програм, виконання зловмисного коду, одержання чутливої інформації.

Процес сканування в загальному випадку можна поділити на чотири етапи:

− сканування мережі: проводиться пошук усіх активних (увімкнутих) пристроїв та з'ясовується топологія мережі;

− сканування портів та ідентифікація сервісів, а саме проводиться пошук відкритих мережевих портів та пов'язаних із ними сервісів;

− пошук уразливостей, а саме відповідно до баз даних уразливостей, відбувається пошук служб, які мають відомі вразливості;

− сканування бездротового сегмента цільової системи, а саме пошук несанкціонованих бездротових пристроїв, пошук пристроїв, що порушують політику безпеки установи або створюють передумови до несанкціонованого проникнення зловмисників у систему.

Знайдені програмні та апаратні засоби за допомогою сканерів уразливостей перевіряють на наявність потенційних проблем та уразливостей, тобто йде пошук за відомими сигнатурами. Ці сигнатури

складаються з комбінацій точок початку відліку, які призначені для демонстрації відомих проблем. З метою найбільш якісної ідентифікації проблем та уразливостей повинна використовуватись якомога більша кількість точок початку відліку. Такими точками відліку можуть бути:

− версія операційної системи, оскільки програмне забезпечення може бути вразливим на одній версії операційної системи, але водночас не бути вразливим на іншій;

− рівень патчів, які встановлені в операційній системі, оскільки чим вищий рівень, тим менша ймовірність наявності уразливостей (недоліків);

− версія програмного забезпечення, оскільки саме від цього залежить успішність проведення тестування.

З метою одержання кращих результатів необхідно використовувати автентифіковане сканування. У цьому разі сканер уразливості буде використовувати надані йому облікові дані. Це дозволить одержати більш глибокий рівень видимості цілі та інформацію, яка недоступна за звичайного сканування.

На четвертому етапі тестування відбувається одержання доступу до системи. Якщо відбувається тестування зовнішнього периметра мережі, то такий доступ здійснюється за трьома найбільш імовірними напрямками:

− злам бездротових точок доступу з подальшим проникненням у систему;

− злам додатків і сервісів, які доступні з глобальної мережі;

− шляхом вдалого використання засобів соціальної інженерії, а саме: надсилання вірусів, троянів, програм віддаленого доступу на корпоративні електронні адреси, виманювання логінів і паролів до сервісів у працівників

установи-замовника, проникнення на контрольовану територію та підключення обладнання тестувальників до внутрішньої мережі.

Для досягнення цілей тестування необхідно вибирати елемент (пристрій), злам якого дасть найбільше переваг. Злам сервера є найбільш доцільним, оскільки це дозволить проводити більш ефективне просування мережею та виконати завдання тестування. Популярними атаками є злам пароля, перехоплення даних, перехоплення сесії, переповнення буфера. Атаки можуть бути комбінованими та включати не лише технічну, а й соціальну сторону (використання методів соціальної інженерії).

Після одержання мінімальних прав доступу в системі відбувається підвищення привілеїв та подальше просування до системи. Завданням тестувальника на цьому етапі стає одержання максимальних прав. Для початку необхідно визначити рівень прав у системі та набір доступних дій, які були досягнуті під час проведення атаки. Виконання деяких операцій буде недоступним, у зв’язку з цим необхідно провести підвищення своїх привілеїв. Можна встановити і запустити клавіатурного шпигуна, який дозволить здійснити перехоплення паролю адміністратора або встановити програму для спостереження за монітором зламаного пристрою. Методики підвищення привілеїв залежать від операційних систем, засобів захисту, програмних налаштувань тощо. Після вдалого підвищення привілеїв проводиться спроба досягнення цілей тестування. Якщо це не вдається, проводиться повторний пошук інформації про систему та сканування уразливостей із використанням підвищених прав.

Після одержання максимально доступних прав повинне бути виконане основне завдання, визначене в

договорі на надання відповідних послуг. Найбільш поширеними цілями тестування є перевірка:

− можливості одержання несанкціонованого доступу до конфіденційної інформації, комерційної таємниці або іншої чуттєвої інформації;

− можливості порушення штатного режиму роботи системи, сервісів, додатків;

− можливостей внутрішнього (зовнішнього) порушника під час його роботи в цільовій системі;

− виконання політики безпеки;

− дієвості засобів захисту;

− злагодженості роботи співробітників, відповідальних за забезпечення кібербезпеки установи-замовника;

− відповідності системи захисту вимогам міжнародних стандартів тощо.

Після досягнення цілей тестування або для їх досягнення в майбутньому необхідно забезпечити постійний доступ до цільової системи. Цей етап також потрібний для демонстрації та підтвердження самого факту компроментації системи. Здебільшого ця фаза тестування реалізується шляхом установлення в системі бекдору – спеціалізованої програми, яка дозволяє одержати доступ до цільової машини без використання процедури автентифікації і залишатися непомітним для користувачів.

За результатами тестування на проникнення повинен складатися звіт. Для вжиття найбільш ефективних заходів із підвищення рівня захищеності та усунення виявлених недоліків пропонується такий зміст кожної з двох вищезазначених частин:

− перша частина для керівництва установ замовника. Інформація в цьому розділі повинна

викладатися для людей, які не мають відповідної фахової освіти або технічних знань;

− друга частина – для керівників підрозділів, відповідальних за кібербезпеку та налаштування відповідного обладнання, системних адміністраторів та адміністраторів безпеки цільової системи.

Перша частина повинна містити:

− загальні відомості про проведені роботи, а саме відомості про цільову систему, що були відомі на початок проведення робіт, часові проміжки, типи атак, що використовувалися, відомості щодо програмного забезпечення, прізвища та контактні дані фахівців, що проводили тестування;

− мету проведення тестування та інформацію щодо її досягнення;

− результати тестування та їх пояснення;

− коротку інформацію про наявні недоліки в системі кібербезпеки, можливості, які відкриваються зловмисникам у разі використання цих недоліків;

− інформацію про контрзаходи, які не були вжиті під час налаштування системи захисту, а особливо ті, які призвели до наявності вразливості;

− орієнтовні сценарії дій зловмисників у разі експлуатації виявлених уразливостей;

− рекомендації (дії), які повинно вжити керівництво установи-замовника для усунення виявлених недоліків у системі кібербезпеки та підвищення рівня захищеності системи.

Друга частина звіту повинна найбільш детально розкривати технічні недоліки в системі безпеки цільової інформаційно-комунікаційної системи. У ній повинні надаватися конкретні рекомендації щодо усунення (мінімізації) виявлених недоліків у системі безпеки. У цій частині доцільно вставляти скриншоти, на яких буде

показане програмне забезпечення, що використовувалося для проникнення, відповідні команди (налаштування) та скриншоти з результатами експлуатації уразливостей.

Друга частина повинна містити:

− інформацію про сильні та слабкі сторони в системі захисту цільової системи;

− можливі недоліки в політиці безпеки, процедурах;

− відомості про виявлені вразливості;

− оцінку ризиків;

− рекомендації щодо впровадження нових систем кіберзахисту (брандмауерів, апаратних засобів шифрування інформації, системи виявлення та попередження вторгнень, антивірусних засобів) та проведення періодичних тестувань, оцінювань захищеності, навчання персоналу тощо.

Також звіт про виявлені вразливості повинен містити:

− інформацію щодо кількості виявлених недоліків із деталізацією за рівнем їх небезпечності (критичні, небезпечні, середні, слабкі);

− перелік серверів, комп'ютерів, маршрутизаторів тощо, в яких були виявлені вразливості.

Звіт з оцінкою ризиків про кожну вразливість, завдяки якій було здійснено втручання в систему, повинен містити:

− оцінку кожної вразливості (критична, небезпечна, середня, слабка);

− короткий семантичний (смисловий) опис причин, що призвели до вразливості;

− детальний опис вразливості та можливостей, які відкриваються зловмисникам у разі її експлуатації;

− рекомендації з усунення кожної (конкретної) вразливості.

У разі необхідності наявність недоліків у системі повинна бути продемонстрована наочно як керівництву компанії-замовника, адміністраторам безпеки, так і іншим технічним працівникам, відповідальним за функціонування комп'ютерної системи, яку тестували.

Після демонстрації наявних недоліків у системі кібербезпеки повинні видалятись усі «точки входу» у систему, які використовували тестувальники. Такі «точки входу» повинні видалятись за присутності адміністраторів безпеки та керівників підрозділів, відповідальних за кібербезпеку та технічне обслуговування системи з оформленням і підписом відповідного акта.

Деякі проблеми, що можуть негативно вплинути на результати тестування на проникнення

Під час організації та проведення тестування на проникнення можуть виникнути проблеми, що не дозволять одержати об'єктивну оцінку захищеності системи, а саме:

1. Під час проведення тестування на проникнення можливий супротив із боку технічних працівників (адміністраторів безпеки, системних адміністраторів) та звичайних користувачів. Зазвичай це пов'язано з людським фактором. Як тільки стане відомо про проведення тестування, вищезазначені категорії користувачів можуть ужити дій із підвищення рівня безпеки серверів, окремих комп'ютерів, мережевого обладнання, провести періодичне вимкнення мережевого обладнання. Підвищення рівня безпеки може відбуватися лише на період проведення перевірки, після цього конфігурація системи безпеки буде повернута до попереднього стану. Причиною такої поведінки можуть бути страх втрати роботи, страх отримання догани, втрати керованості обладнанням. Причиною також може бути

бажання отримати подяку або підвищену премію від керівництва у разі неможливості досягнення тестувальниками цілей тестування.

Можливими варіантами вирішення цієї проблеми є такі:

− роз'яснення з боку керівництва установи-замовника про відсутність стягнень для технічних працівників у разі вдалого проведення перевірки;

− проведення перевірок без інформування технічних працівників та звичайних користувачів;

− проведення раптових перевірок.

2. Часові обмеження варто розглядати з двох боків. По-перше, будь-який тест повинен займати певний проміжок часу, наприклад один місяць. У цьому разі тестувальники повинні провести відповідні дії максимально повно та якісно, а також скласти відповідні документи за результатом тестування. За наявності великої кількості серверів, комп'ютерів, мережевого обладнання це стає досить складним завданням. З іншого боку, під час тестування критичних систем із метою внеможливлення втрати їх доступності власник може вводити обмеження стосовно часових проміжків, упродовж яких дозволено виконувати тестування (наприклад, із 21:00 до 8:00).

3. Обмеження, пов'язані з недостатністю ресурсів. Під час проведення тестувань на проникнення необхідно враховувати наявність відповідного апаратного та програмного забезпечення. Так, за великої кількості тестувальників їх необхідно забезпечити відповідним обладнанням: ноутбуками, WI-FI-обладнанням, зовнішніми накопичувачами тощо. Перед проведенням тестування необхідно переконатися, що наявне спеціалізоване програмне забезпечення допоможе якісно виконати поставлене завдання та досягти цілей

тестування. Крім того, варто враховувати терміни дії ліцензії на відповідні програми.

4. Стрімкий розвиток технологій, методів та інструментів для проникнення в мережі потребує постійних фінансових витрат не лише на придбання відповідного апаратного і програмного забезпечення, а й періодичного навчання співробітників та/або підвищення їх кваліфікації.

5. Негативний вплив на цільову систему внаслідок проведення тестування. Проведення тестування на проникнення може спричинити непередбачені наслідки для цільової системи: відмову в обслуговуванні, втрату інформації, короткочасне переривання виконання функцій, втрату інформації тощо. Тому необхідно перебачити встановлення програмного забезпечення, яке буде повністю записувати дії тестувальників та/або зовнішніх засобів відеоспостереження. Відповідні записи можуть допомогти в протидії звинуваченням у тому, що тестування негативно вплинуло на роботу цільової системи. Також такі записи можуть допомогти відновити роботу системи, якщо збій відбувся внаслідок дій тестувальників.

За можливості варто здійнювати відеофіксацію всіх дій тестувальників. Особливо це важливо для об’єктів критичної інфраструктури. Наявність такого відеозапису дозволить технічним працівникам установи-замовника максимально швидко відновити роботу комп’ютерної системи в разі, якщо дії тестувальників призведуть до негативних наслідків.

Висновок

Постійний розвиток інформаційно-комунікаційних систем та цифрова трансформація органів державної влади України потребує впровадження нових засобів

забезпечення кібербезпеки та злагоджених дій команд, відповідальних за захист електронних державних інформаційних ресурсів. Ураховуючи той факт, що державні інформаційно-комунікаційні системи постійно під'єднані до глобальної мережі «Інтернет», це не дозволяє використовувати застарілі підходи до захисту інформації. Кожну окрему систему кібербезпеки необхідно постійно перевіряти, конфігурувати та модернізовувати для протидії сучасним загрозам, новим інструментам та методам хакерських атак. Тому оцінювання ступеня захищеності інформаційно-комунікаційних систем повинне передбачати не лише перевірку організації роботи служб захисту інформації, а й перевірку практичної захищеності з використання інструментів та методів зламу, доступних широкому загалу. Ураховуючи величезну кількість операційних систем, програмних застосунків, засобів захисту необхідно розробити національну методологію оцінювання кіберзахищеності, що дозволить максимально повно та якісно оцінювати захищеність без урахування різноманітності програмного забезпечення інформаційно-комунікаційних систем.

СПИСОК ЛІТЕРАТУРИ

1. Конституція України // Відомості Верховної Ради України (ВВР). – 1996. – № 30. – С. 141.

2. Про захист інформації в інформаційно-телекомунікаційних системах : Закон України від 05.07.1994 № 1170-VII (1170−18) від 27.03.2014 // Відомості Верховної Ради. – 2014. – № 22. – С. 816.

3. Про затвердження Правил забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах» : Постанова

Кабінету Міністрів України від 29.03.2006 р. № 373 // Офіційний вісник України. – 2006. – № 13. – С. 164.

4. Про затвердження Положення про державний контроль за станом ТЗІ : Наказ Адміністрації Держспецзв'язку від 16.05.2007 № 87.

5. Про затвердження Порядку оцінки стану захищеності державних інформаційних ресурсів в інформаційно-телекомунікаційних системах : Наказ Адміністрації Держспецзв'язку від 02.12.2014 № 660.

6. Кальченко В. В. Огляд методів проведення тестування на проникнення для оцінки захищеності комп'ютерних систем / В. В. Кальченко // Системи управління, навігації та зв'язку. – 2018. – № 4. – С. 109−114.

7. Samonas S. The CIA Strikes Back: Redefining Confidentiality, Integrity and Availability in Security / S. Samonas, D. Coss // Journal of Information System Security. – 2014. – № 10 (3). – P. 21–45.

8. Technical Guide to Information Security Testing and Assessment. Recommendations of the National Institute of Standards and Technology. Special Publication 800−115. [Electronic resource]. – Access mode : http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication 800− 115.pdf.

9. Information Systems Security Assessment Framework (ISSAF) [Electronic resource]. – Access mode : http://www.oissg.org/files/issaf0.2.1.pdf.

10. Chapple M., Seidl D. CompTIA® PenTest+ Study Guide: Exam PT0−001. – Indianapolis : John Wiley & Sons, Inc., 2019. – 519 p.

11. Hutchins E. M. Intelligence − Driven Computer Network DefenseInformed by Analysis of Adversary Campaigns andIntrusion Kill Chains [Electronic resource] / E. M. Hutchins, M. J. Cloppert, R. M. Amin. –

Access mode : https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf.

12. The Open Source Security Testing Methodology Manual (OSSTMM) [Electronic resource]. – Access mode : http://www.isecom.org/mirror/OSSTMM.3.pdf.

РОЗДІЛ 14
МОДЕЛЬ І МЕТОД МАШИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ПРИСТРОЯХ ІНТЕРНЕТУ РЕЧЕЙ

А. С. Москаленко, В. В. Москаленко

Вступ

В умовах гібридної війни кіберпростір України є ціллю для кібератак, наслідком яких можуть бути значні економічні втрати, а в певних ситуаціях і екологічні катастрофи. Зокрема, у 2017 році, коли комп'ютерний вірус Petya дестабілізував економіку України. З огляду на це дослідження й розроблення технологій кіберзахисту є актуальними для вирішення безпекових питань держави.

Кіберфізичні системи та Інтернет речей як їх складова збільшують свій економічний, комерційний і соціальний вплив на наше життя. Крім того, кіберфізичні системи стають звичайним елементом критичної інфраструктури. Водночас розподілені пристрої кіберфізичних систем, здебільшого, обмежені в ресурсах, що робить їх привабливими цілями для кібератак. Згідно з даними лабораторії AV-TEST (https://www.av-test.org/en/news/) у 2021 році очікують на появу понад 1 мільярда шкідливих програм. Незважаючи на те, що традиційно ціллю шкідливого програмного забезпечення були персональні комп'ютери, останнім часом зростає кількість атак на пристрої Інтернету речей, такі як смартфони, холодильники, пральні машини, розподілені сенсори, медичне обладнання та інше. Водночас більшість кібератак спрямовані на зараження пристроїв для їх контролю, щоб одержати несанкціонований доступ до конфіденційної інформації, реалізувати ботнет-мережі чи

вчинити інші неправомірні дії.

Загалом пристрої Інтернету речей є незахищеними внаслідок складності створення уніфікованих стандартів для гетерогенного кіберфізичного середовища. Пристрої Інтернету речей мають різноманітну архітектуру апаратного й програмного забезпечення. На відміну від персональних комп'ютерів у них домінують різноманітні операційні системи сім'ї Linux. На ринку немає ефективних розробок для розпізнавання шкідливого програмного забезпечення на таких платформах. Водночас традиційні методи синтезу систем розпізнавання шкідливого програмного забезпечення є обчислювально трудомісткими й чутливими до атак «нульового дня», тому задача розроблення обчислювально ефективних засобів розпізнавання шкідливого програмного забезпечення, що забезпечуватимуть робастність до цільової платформи й високу узагальнювальну здатність, є актуальним завданням.

Кіберзагрози від шкідливих програм (malware) зростають із кожним роком. Традиційні сигнатурні методи розпізнавання шкідливого програмного забезпечення є малоефективними внаслідок швидкого зростання кількості і різноманітності шкідливого програмного забезпечення. Для протидії таким атакам потрібно регулярно оновлювати антивірусне програмне забезпечення, але навіть це не вбезпечить від атак «нульового дня». Методи машинного навчання завдяки узагальненню даних дозволяють прогнозувати й розпізнавати нові шкідливі програми, проте застосування традиційних підходів призводить до частих хибних спрацювань (false positive). Багато досліджень проводяться для їх зменшення.

Сучасні системи кіберзахисту пристроїв кіберфізичних систем не забезпечують високої

достовірності рішень, тому що кількість і різноманітність нових джерел кіберзагроз постійно зростає, а актуальні розмічені дані є досить обмеженими [1]. Гетерогенність кіберфізичних систем зумовлює високу варіативність спостережень і знижує репрезентативність будь-яких вибірок даних.

Уручну спроєктовані ознаки, використовувані в класичних методах машинного навчання, не є гнучкими й адаптованими до появи нових реалізацій шкідливого програмного забезпечення. З огляду на це варто розглядати застосування функціонально здатного до навчання екстрактора ознак, що може автоматично формувати ефективні ознаки з вхідних навчальних даних. Екстрактори ознак здебільшого базуються на методах глибокого машинного навчання. Найбільш потужною частиною глибокого навчання є навчання ієрархічного ознакового подання (рис. 14.1).

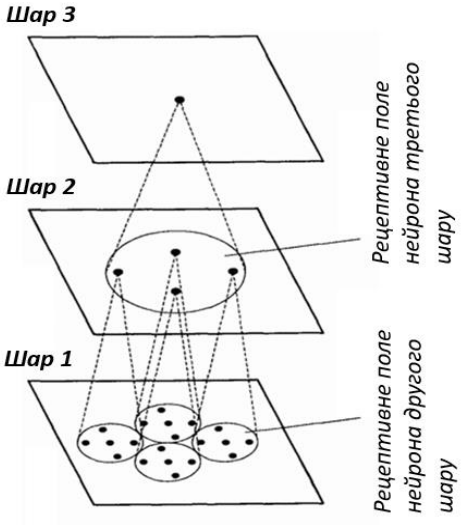
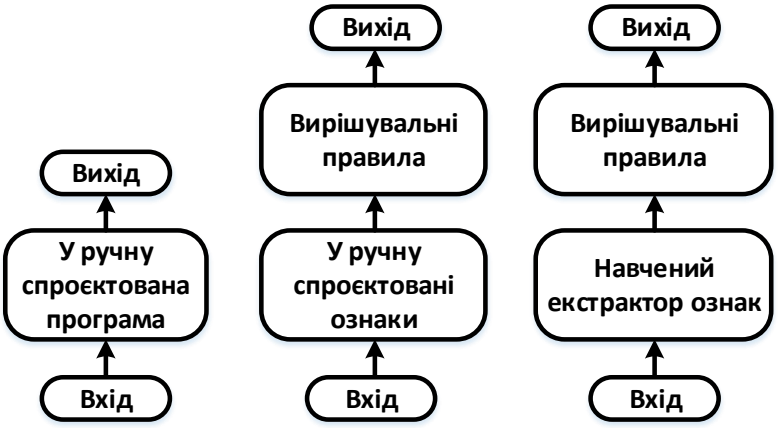


Рисунок 14.1 – Схема ієрархічного ознакового подання даних

Високорівневі ознаки формуються способом компонування низькорівневих. Вище за ієрархією перебувають менше ознак, проте вони є складнішими, інформативнішими та менш чутливими до варіації спостережень одного й того самого класу розпізнавання.

Узагальнену структуру моделі класифікаційного аналізу шкідливого програмного забезпечення в рамках кожного окремого підходу наведено на рисунку 14.2.



а) *б)* *в)*

Рисунок 14.2 – Узагальнена структура моделі аналізу програмного забезпечення: а) моделі, що базуються на правилах (наприклад, сигнатурний аналіз); б) моделі, що базуються на застосуванні класичного машинного навчання; в) моделі, що базуються на навчанні ієрархічного екстрактора ознак

Дослідники з Intel й Microsoft довели ефективність аналізу шкідливого програмного забезпечення комп'ютерів із використанням згорткових глибоких

нейронних мереж [2, 3]. Водночас бінарні виконувані файли напряму байт за байтом подавали як одноканальне 2D-зображення в сірих відтінках. Такий підхід дозволяє застосовувати добре напрацьовані моделі й методи навчання згорткових мереж. Проте в зазначеній праці розглянутий як цільова платформа персональні комп'ютери під управлінням операційної системи Windows. Однак пристрої Інтернету речей мають більшу варіативність операційних систем, бібліотек та апаратного забезпечення, що потребує підвищення інформативності ознакового подання й удосконалення самої моделі аналізу даних з метою запобігання ефектам упередженості чи насиченості.

Крім екстрактора інформативного високорівневого ознакового опису спостережень, у системі детектування шкідливого програмного забезпечення важливим компонентом є вирішальні правила, що здебільшого є класифікатором. Водночас ефективність навчання класифікатора часто розглядають як показник ефективності навчання екстрактора ознак.

У завданнях класифікації подібні об'єкти переважно перебувають в одному класі, тому часто користуються гіпотезою компактності. Згідно з нею класи утворюють компактно локалізовані підмножини в просторі об'єктів.

Водночас для формалізації поняття «подібності» вводять функцію відстані або метрику $d(x, y)$ в N-вимірному просторі об'єктів.

Алгоритм найближчого сусіда (nearest neighbor, NN) є найбільш простим метричним алгоритмом, що класифікує об'єкт як реалізацію того класу, до якого належить найближчий об'єкт навчальної вибірки. Навчання такого класифікатора зводиться до елементарного запам'ятовування навчальної вибірки

$\{y_{m,i}^{(j)} \mid m = \overline{1, M};\ j = \overline{1, n};\ i = \overline{1, N}\}$, де M – потужність алфавіту класів розпізнавання; n – обсяг спостережень класу X_m^o; N – кількість ознак розпізнавання. Єдиною перевагою цього алгоритму є простота реалізації, а недоліків значно більше. По-перше, викиди в навчальній вибірці призводять до нестійкості й похибок. По-друге немає параметрів, що можна налаштовувати за навчальною вибіркою. Алгоритм повністю залежить від успішності вибору дистанційної міри *d (x, y).* Міра близькості обирається відповідно до властивостей об'єктів. Найбільш поширені такі метрики, як Евклідова, Манхеттенська, Степенева та Журавльова [4; 5].

Метричні алгоритми локально апроксимують вибірку, під час чого обчислення відкладаються доти, доки не стане відомим вхідний об'єкт. З огляну на це метричні алгоритми належать до методів лінивого навчання (lazy learning).

У рамках універсальнішого підходу до аналізу й синтезу здатних навчатися систем прийняття рішень, яким є геометричний підхід, став популярним метод опорних векторів SVM (support vector machine) [6]. В основі SVM лежить ідея розділення згущення векторів гіперплощинами, що знаходяться на максимальній відстані від згущень за мінімізації зміщення реалізацій класу від опорного вектора.

У праці [6] доведено, що дискретне двійкове подання даних забезпечує регуляризувальні та метарегуляризувальні ефекти. Воно є ефективним для боротьби з перенавчанням і зменшенням трудомісткості побудови вирішувальних правил. Подібний підхід розроблено для розв'язування задач багатокласової класифікації способом її заміни еквівалентною множиною двокласових задач. Одним із найефективніших методів

зведення багатокласової класифікації до серії двокласових є двійкове кодування міток класів кодами, що виправляють помилки (Error Correcting Output Codes, ECOC) [7].

Коди, що виправляють помилки, використовуються для кодування класів. Для цілей навчання мітку чи номер класу записують як двійкове число з N-розрядів. Кожен розряд може відповідати виходу окремої моделі бінарного класифікатора чи одному з виходів однієї моделі. Навчена модель (моделі) формуватимуть на виході двійковий код, що можна порівнювати з двійковим кодом кожного з класів. Вхідний зразок належить до того класу, до якого мінімальна відстань Хемінга. Найпоширенішим способом вибору кодів для кодування класів є побудова кодової матриці Адамара. Її використовують для порівняння прогнозованого коду з кодовою матрицею. Вона забезпечує максимальну кодову відстань як між рядками, так і між стовпцями. Проте ECOC у такому виконанні не враховують структури і варіативність класів розпізнавання, що знижує його ефективність у практичних завданнях класифікаційного аналізу.

Отже, питання вибору оптимальних в інформаційному сенсі параметрів моделей розпізнавання вторгнень є актуальним. Його вирішення ускладнене неповною визначеністю даних, зумовленою нестаціонарністю процесів формування шкідливого трафіку та обмеженим обсягом актуальних розмічених навчальних даних. Водночас одним із найперспективніших підходів до класифікаційного аналізу шкідливого трафіку, описаного високорівневими ознаками, є застосування ідей і методів інформаційно-екстремальної інтелектуальної технології. Інформаційно-екстремальний класифікатор є обчислювально

ефективний та характеризується ефективністю машинного навчання за умов обмеженого обсягу розмічених навчальних вибірок.

Модель і метод навчання класифікатора шкідливого програмного забезпечення

Двійкові файли програмного забезпечення доцільно аналізувати способом їх перетворення на кольорові
RGB-зображення за наведеною нижче схемою (рис. 14.3).

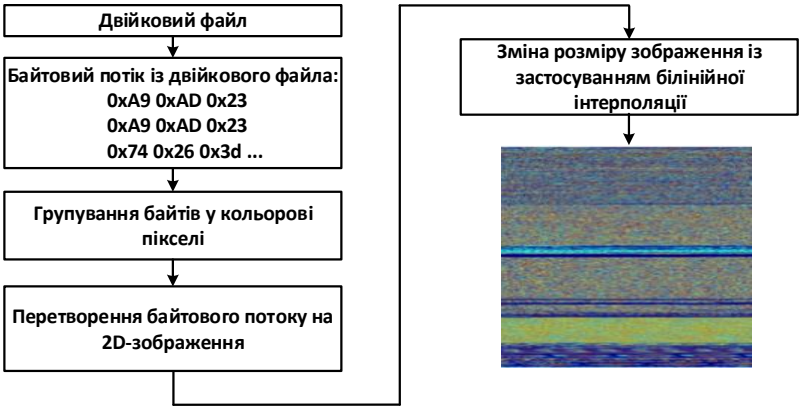


Рисунок 14.3 – Етапи запропонованого методу навчання

Для вибору ширини кольорового зображення використовують таблицю 14.1 [3].

Таблиця 14.1 – Залежність рекомендованої ширини зображення від кількості сформованих пікселів

Кількість пікселів	Ширина зображення
Від 1 до 10	32
Від 10 до 30	64
Від 30 до 60	128
Від 60 до 100	256
Від 100 до 200	384
Від 200 до 1 000	512
Від 1 000 до 1 500	1 024
Більше ніж 1 500	2 048

Для ефективності ініціалізації вагових коефіцієнтів великих згорткових нейронних мереж згідно принципом трансферу знань (transfer learning) потрібно змінити розмір зображення. Він повинен дорівнювати одному з розмірів, використовуваних під час попереднього навчання нейромережі на наборі ImageNet.

Для експериментів і перевірки концепції як основу для екстрактора ознак запропоновано використати згорткову мережу загального призначення MobileNet без вихідних шарів із коефіцієнтом ємності 0,25 [4].

Зазначений метод охоплює чотири основні етапи (фази) (рис. 14.4). Перші три з них – це безпосередньо навчання екстрактора ознакового опису. На останньому етапі розробляють вирішальні правила та коригують їх ураховуючи дисперсію спостережень усередині класів у двійковому просторі Хеммінга [8].

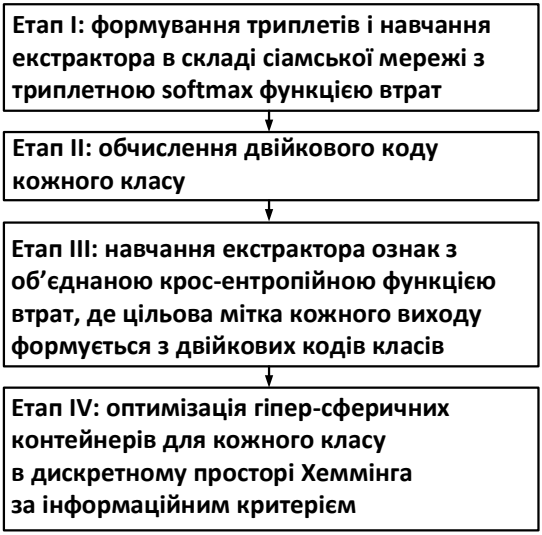


Рисунок 14.4 – Етапи запропонованого методу навчання

Перший етап навчання моделі аналізу зображення проводять із застосуванням адаптивних алгоритмів зворотного поширення помилки, найбільш популярним із яких є Adam [8]. Триплетну функцію втрат розраховують за формулою

$$L = -\log \frac{\exp(\left\| f(x_a) - f(x_{ep}) \right\|)}{\exp(\left\| f(x_a) - f(x_{ep}) \right\|) + \exp(\left\| f(x_a) - f(x_{shn}) \right\|)} \quad (14.1)$$

де f(x) – функція, що описує екстрактор і встановлює залежність між вхідним зображенням та вектором виходу сигмоїдного шару; x_a – зображення, випадково вибране з мініпакета; x_{ep} – найближчий сусід із мінібатчу, що належить до того самого класу, тобто

$$x_{ep} = \arg \min_{x:C(x)=C(x_a)} \left\| f(x_a) - f(x) \right\|,$$

(14.2)

де C(x) – функція що повертає клас зображення; x_{shn} – зразок зображення з мініпакета, що є найближчим серед зразків протилежних класів, проте знаходиться далі, ніж зразок, тобто

$$x_{shn} = \arg \min_{\substack{C(x) \neq C(x_a) \\ x: \left\| f(x_a) - f(x) \right\| > \left\| f(x_a) - f(x_p) \right\|}} \left\| f(x_a) - f(x) \right\| \cdot$$

(14.3)

Друга фаза потрібна для перетворення на бінарну форму вихідного вектора відповідно до принципів самокоректувальних кодів (error-correcting output codes), але з урахуванням внутрішньої структури класів і відношень між зразками різних класів. Це відбувається способом бінаризації вихідного ознакового опису й порівняння фонової частоти кожного ознаки з частотою цієї ознаки в конкретному класі.

Третя фаза навчання полягає в точному налаштуванні за кросентропійною функцією втрат (joint binary cross-entropy loss) для збільшення компактності розподілу класів у просторі Хеммінга

$$L = -\sum_{i=1}^{N} \left(b_i \log f_i(x) + (1 - b_i) \log(1 - f_i(x)) \right),$$

(14.4)

де $f_i(x)$ – значення і-го виходу сігмоїдного шару для

вхідного зображення x; b_i – значення і-го розряду еталонного вектора класу, до якого належить зображення *x*.

Остання фаза машинного навчання пов'язана з оптимізацією радіуса контейнерів за інформаційним критерієм [8] для врахування меж відхилення двійкового подання спостережень кожного класу від відповідних еталонних векторів. Як інформаційний критерій доцільно використовувати модифікацію інформаційного показника Шеннона, що вираженого як функціонал від точнісних характеристик вирішувальних правил. Також для валідації одержаної моделі можна застосовувати традиційну метрику – правильність або акуратність моделі (accuracy).

Отже, для створення інформативного ознакового опису й завадозахищених вирішувальних правил варто будувати модель способом поєднання ідей і методів сіамських мереж та інформаційно-екстремального навчання.

Результати експерименту й висновки

Запропоновані модель і метод навчання перевірялися як на загальнодоступних, так і на самостійно зібраних наборах даних. Оскільки розглядали робастний алгоритм до платформи пристроїв інтернету речей, то вибіркові дані містили двійкові файли для популярних архітектур з операційними системами Windows та Linux. Вибірка даних для платформи Windows (тобто PE-файли) була сформованою з набору даних KISA Data Challenge 2019 [3] і містила 18 000 зразків шкідливого й 12 000 зразків нешкідливого програмного забезпечення, з яких по 5 000 зразків відібрали для тестування моделі. Вибірка даних для платформи Linux (тобто ELF-файли) складалася з 10 000 зразків шкідливого програмного забезпечення, одержаних із набору даних Virus-Share

Dataset [3], та 10 000 нешкідливих зразків зібраних з директорій /bin і /usr/bin на одноплатних комп'ютерах Raspberry pi та Jetson Nano з різними Linux-операційними системами. Водночас у тестову вибірку шкідливих і нешкідливих програм було відібрано по 2 000 зразків. Роздільна здатність зображень під час кодування виконуваних файлів у зображення становила $160 \bullet 160$ пікселів.

Розглянемо результати машинного навчання класифікатора шкідливого програмного забезпечення на сформованих навчальних даних. На рисунку 14.5 зображені зміни точності моделі на тестовій (test_acc) і навчальній (train_acc) вибірках. Кожен мінібатч містив 128 зображень, а коефіцієнт швидкості навчання становив 0,000 01.

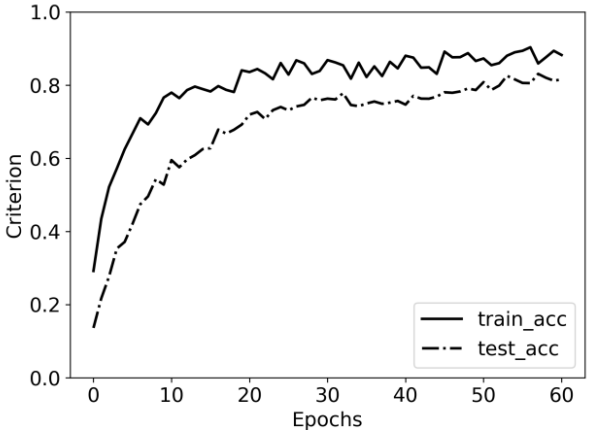


Рисунок 14.5 – Графік залежності точності під час навчання класифікатора дефектів від кількості епох навчання в рамках традиційного підходу

Згідно з рисунком 14.5 підвищення точності припинилося після 43 епох навчання з максимальним

значенням точності остаточної моделі, що дорівнювала 81,5 % на тестовій вибірці. Водночас відстань між кривою на тестовій і навчальній вибірках свідчить про невеликий ефект перенавчання.

На рисунку 14.6 наведені результати навчання запропонованого екстрактора ознак із сигмоїдним шаром та функцією втрат (14.1). Кожні п’ять епох будували інформаційно-екстремальні вирішувальні правила та обчислювали середній за алфавітом класів інформаційний критерій на тестовій (test_info_cri) і навчальній (train_info_cri) вибірках [9–12]. Після 30 епох навчання модель було збережено для наступного експерименту.

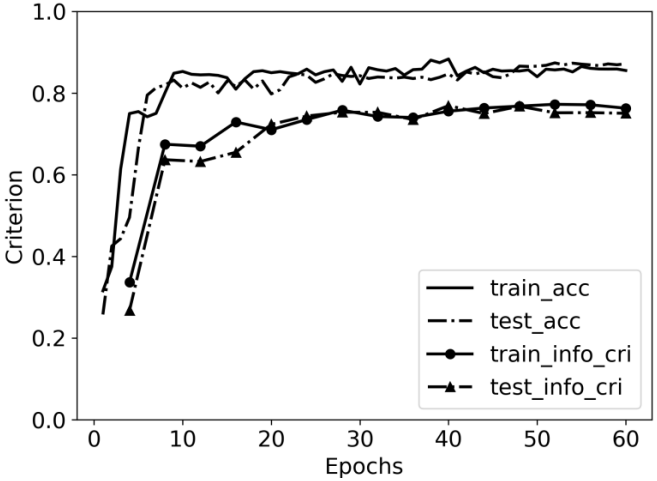


Рисунок 14.6 – Графік залежності точності й інформаційного критерію від кількості епох навчання в рамках запропонованого підходу з функцією втрат (14.1)

Відповідно до рисунка 14.6 з 10-ї епохи зростання точності значно сповільнилося, а після 40 епох було досягнуто максимальної точності 87,3 %, що на 7,11 % більше, ніж у базовій (baseline) моделі. Проте після 40 епох

навчання подальше покращання майже припинилося. Водночас тестові й навчальні криві приблизно збігаються, що свідчить про високу узагальнювальну здатність одержаних вирішувальних правил.

Середнє за алфавітом класів значення інформаційного критерію становило 0,751, а середня відстань між центрами радіально-базисних вирішувальних правил – 17 кодових одиниць, а середній радіус контейнерів класів – 9 кодових одиниць.

На рисунку 14.7 можемо бачити подальші результати навчання моделі, збереженої на попередньому етапі після 30 епох навчання. Далі навчали також 30 епох, але із застосуванням функції втрат (14.4) для зменшення внутрішньокласової дисперсії в просторі Хемінга [8].

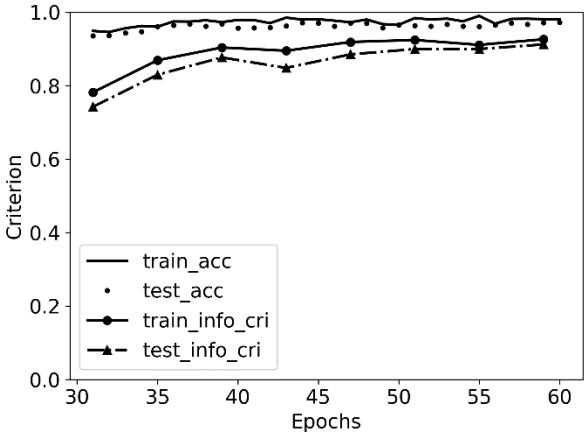


Рисунок 14.7 – Графік залежності точності та інформаційного критерію від кількості епох навчання в рамках запропонованого підходу з функцією втрат (14.4)

Згідно з рисунком 14.7 застосування функції joint binary cross-entropy loss (10.4) забезпечило зростання точності на 11 % (вона становила 97,2 %) порівняно з

попереднім етапом. Водночас середнє за алфавітом класів значення інформаційного критерію дорівнювало 0,91. Середня відстань між центрами гіперсферичних контейнерів дорівнює 18 кодових одиниць, а середній радіус контейнерів класів – 8 кодових одиниць. Тобто регуляризувальний ефект від поєднання принципів геометричного й інформаційного підходів дозволяє побудувати завадозахищені вирішувальні правила.

Отже, одержана модель класифікатора виконуваних програмних кодів забезпечує прийнятну для практичного використання точність класифікації на тестовій вибірці 97,2 % і перевищує результат традиційної схеми навчання із softmax вихідним шаром на 19 %. Крім того, результати є кращими за результати інших дослідників на 3,2 % [3, 7].

СПИСОК ЛІТЕРАТУРИ

1. Lightweight Classification of IoT Malware based on Image Recognition / J. Su, D. V. Vasconcellos, S. Prasad et al. // 42nd IEEE Annual Computer Software and Applications Conference (Tokyo, 22 June 2018). – 7 p. – DOI: 10.1109/COMPSAC.2018.10315.

2. Ke He. Malware Detection with Malware Images using Deep Learning Techniques / Ke He, Dong Seong Kim // 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (Rotorua, 31 October 2019). – P. 95–102. – DOI: 10.1109/TrustCom/BigDataSE.2019.00022

3. Platform-Independent Malware Analysis Applicable to Windows and Linux Environments / Chanwoong Hwang, Junho Hwang, Jin Kwak, Taejin Lee // https://www.mdpi.com/journal/electronics. – Vol. 9, 2020. – 18 p. – DOI: 10.3390/electronics9050793.

4. Autoencoder-based feature learning for cyber

security applications / M. Yousefi-Azar, V. Varadharajan, L. Hamey, U. Tupakula // Proceedings of the 2017 International Joint Conference on Neural Networks (Anchorage, 14–19 May 2017). – P. 3854–3861.

5. Going deeper with convolutions / C. Szegedy, W.Liu, Y. Jia, P. Sermanet et al. // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (Boston, MA, 7–12 June, 2015). – P. 1–9.

6. Москаленко В. В. Моделі і методи інтелектуального аналізу багатовимірних даних да умов апріорної невизначеності : монографія / В. В. Москаленко. – Суми : СумДУ, 2019. – 184 с.

7. Deep learning of support vector machines with class probability output networks / S. Kim, Z. Yu, R. Man Kil, M. Lee // Neural Networks. – 2015. – Vol. 64. – P. 19–28.

8. Sewer Pipe Defects Classification Based on Deep Convolutional Network with Information-extreme Error-correction Decision Rules / A. S. Moskalenko, V. V. Moskalenko, M. O. Zaretskyi, V. Lysyuk // Communications in Computer and Information Science (CCIS-2020). – Springer : Cham, 2020. – 20 p. – DOI: 10.1007/978-3-030-61656-4_16.

9. Dovbysh A. S. Information-extreme learning algorithm for a system of recognition of morphological images in diagnosing oncological pathologies / A. S. Dovbysh, M. S. Rudenko // Cybernetics and Systems Analysis. – 2014. – Vol. 50, Issue 1. – P. 157–162.

10. Moskalenko V. V. Optimizing the parameters of functioning of the system of management of data center IT infrastructure / V. Moskalenko, S. Pimonenko // Eastern-European Journal of Enterprise Technologies. – 2016. – Vol. 5, Issue 2 (83). – P. 21–29.

11. Multi-Layer Model And Training Method For Malware Traffic Detection Based On Decision Tree

Ensemble / V. V. Moskalenko, A. S. Moskalenko, M. O. Zaretsky et al. // Radioelectronic and Computer Systems – 2020. – № 2. – P. 92–101.

12. Кудрявцев А. М. Метод самонавчання згорткового екстрактора ознак для розпізнавання багатоканальних діагностичних зображень / В. В. Москаленко, А. С. Москаленко, А. М. Кудрявцев // XIV International scientific conference «Intellectual systems of decision-making and problems of computational intelligence (ISDMCI'2018)». – 2018. – С. 258–260.

РОЗДІЛ 15
ІНФОРМАЦІЙНА БЕЗПЕКА ВЕБДОДАТКІВ

О. Б. Проценко

Захист вебдодатків належить до аспекту інформаційної безпеки, що стосується безпеки вебпрограм, веббезпеки та вебслужб. Безпека вебдодатків виходить за рамки просто веббезпеки, спираючись на принципи безпеки додатків, щоб забезпечити безпеку Інтернету та вебсистем [1].

За деякими даними, більше ніж 80 % сайтів містять критичні вразливості, тому безпека вебдодатків представляє захисні заходи, за яких зловмисник не може одержати доступ до конфіденційних даних як ззовні під час спроби злому, так і всередині вебсистеми через нелегітимний доступ.

Вебдодатки стають фінансово-привабливими не лише для їх розробників, а й для бажаючих нелегально скористатися даними, що зберігаються в них. У цілому метою зловмисників, які здійснюють несанкціонований доступ до вебдодатків, є SEO, поширення спаму, шкідливого коду, крадіжка і спотворення даних, видалення вмісту сайта (DoS-атака), використання сайта як плацдарму для проведення атак на інші ресурси.

Атаки можна умовно поділити на три категорії загроз: порушення *конфіденційності інформації, доступності й цілісності інформації*.

Інформація, системи та ресурси повинні бути *доступні* користувачам у потрібний їм час для виконання ними своїх обов’язків. Відсутність доступу до інформації може дуже негативно впливати на продуктивність роботи користувачів. Необхідно застосовувати механізми забезпечення стійкості до відмов і відновлення для

забезпечення безперервної доступності ресурсів. *Цілісність* інформації характеризується достовірністю і захищеністю від несанкціонованих змін. Механізми безпеки, що забезпечують цілісність інформації, повинні повідомляти користувачів або адміністраторів про факти незаконних змін. Конфіденційність інформації забезпечується захистом від несанкціонованого доступу до неї невповноваженим особам, стороннім програмам або процесам. Розробникам необхідно застосовувати механізми управління даними з ранжуванням рівнів доступу до них [2], оскільки ризики витікання конфіденційної інформації з вебдодатків можуть бути і фінансові, і репутаційні.

Найбільш поширена загроза безпеки вебдодатків – це визначення та використання їх уразливостей. Для злому і виведення додатків із ладу можуть використовуватися різні інструменти, як аматорські, так і професійні, здійснюючи кібератаки з використанням автоматичних систем сканування для експлуатації вразливостей.

Перший крок зловмисника у разі спроби атак – сканування за допомогою різних утиліт та спеціалізованих програм. Це можна виявити за великою кількістю запитів з однієї IP-адреси до різних сторінок і великою кількістю числу помилок 404. Тому безпека вебдодатків повинна передавати механізми безперервного моніторингу.

Захист вебдодатків актуальний за будь-яких умов. Останнім часом кількість працівників, які працюють віддалено, які нерідко використовують вебсистеми компаній з особистих комп'ютерів в обхід VPN, що може спричинити витікання цінної інформації. Тому необхідно забезпечити безперервний моніторинг запитів подібного характеру.

Такі фактори, як зростання хмарних платформ, використання інструментів і технологій і відкритим

кодом, ускладнення вебдодатків, збільшення потреб щодо оброблення даних та збільшення рівня вишуканості кіберзловмисників, приводять до надзвичайно складного та складного середовища для інформаційної безпеки будь-якого вебдодатка. Оскільки хакери більше цікавляться конфіденційними даними користувачів, а випадки кібератак зростають, тому життєво важливо убезпечити вебдодатки.

Найпоширеніші ризики безпеки вебдодатків

Існують різні атаки на вебсистеми, починаючи від безпосереднього маніпулювання базами даних і закінчуючи масштабними розладами мережі. Необхідно враховувати всі можливі види ризиків безпеки вебдодатків, серед яких [5]:

1) ін'єкції (Injection) – ризик безпеки вебдодатків, що трапляється, якщо ненадійні дані надсилаються інтерпретаторові за допомогою команди або запиту. Це досягається, коли зловмисник вводить шкідливий код, який здається звичайним кодом. Недоброзичливі дані зловмисника призводять до виконання шкідливих команд або одержання доступу до даних без відповідних на це повноважень.

Ін'єкційні атаки на вебпрограми можуть призвести до втрати дозволу на доступ або до повної втрати контролю над системою та втрати цінних даних. До вад ін'єкції належать LDAP, OS, NoSQL та ін'єкція SQL;

2) помилки автентифікації (Broken Authentication) характеризуються тим, що порушені дозволи становлять серйозний ризик для організацій. Функції програми, пов'язані з автентифікацією та управлінням сеансами, можуть виконуватися неточно. Це дозволяє зловмисникам компроментувати маркери сеансу, ключі, паролі або використовувати інші недоліки реалізації, щоб прийняти

ідентичність інших користувачів на короткий або невизначений час;

3) іншим ризиком для вебдодатків є розкриття конфіденційних даних (Sensitive Data Exposure), таких як фінансова інформація (дані облікового запису, ПІН-коди, особисті дані), інформація про охорону здоров'я та особиста інформація, що ідентифікує особу. Після доступу зловмисники можуть викрасти або змінити ці погано захищені дані для здійснення атак, шахрайства з кредитними картками, фішингом, викрадення особистих даних та інших пов'язаних із цим атак;

4) eXternal XML Entities (XXE) – це атака, спрямована на додаток, яке обробляє сутність XML (Entity). Існують різні типи сутностей, але конкретно зовнішні сутності можуть бути використані, щоб одержати доступ до локальних або віддалених даних за допомогою певного системного ідентифікатора. Часто ним є URI, до якого парсер може одержати доступ у процесі оброблення документа.

Тобто можливість цієї атаки виникає, якщо XML-код містить посилання на зовнішні сутності, які обробляються погано налаштованим парсером віддаленого хоста. Шляхом передавання шкідливого XML-коду атакуючий може використовувати парсер проти даної системи, або розкрити важливу інформацію, і навіть виконати довільний код. Зловмисники можуть використовувати зовнішні сутності для виявлення внутрішніх файлів за допомогою внутрішнього сканування портів, внутрішнього спільного використання файлів, віддаленого виконання коду тощо;

5) зламаний контроль доступу (Broken Access control) обумовлений випадками, якщо кількість користувачів, уповноважених виконувати певні завдання, не зазначена і не обмежена. Ця вада використовується

кіберзлочинцями для доступу до несанкціонованих даних, для перегляду конфіденційних файлів, доступу до облікових записів інших користувачів, зміни прав доступу або навіть модифікації даних інших користувачів;

6) неправильне налаштування безпеки (Security Misconfigurations) – один із найпоширеніших ризиків для вебдодатків. Неправильна конфігурація безпеки – це недолік, що виникає на основі недопрацьованої системи, відкритого хмарного сховища, небезпечних конфігурацій за замовчуванням, неправильно налаштованих заголовків HTTP, неповних або спеціальних конфігурацій та/або довгих повідомлень про помилки, які можуть містити конфіденційну інформацію.

Експерт із захисту вебдодатків повинен не лише убезпечувати конфігурації всіх програм, фреймворків, операційних систем та бібліотек, а також забезпечити своєчасне оновлення та виправлення помилок;

7) міжсайтовий сценарій (Cross-Site Scripting XSS) – це вразливість, яка дає хакеру можливість виконувати сценарії на боці клієнта, щоб викрасти доступ до сеансів користувача, одержати безпосередній доступ до конфіденційної інформації, зіпсувати вебсайти, видавати себе за користувача з широкими правами або переспрямовувати й на шкідливі вебсайти;

8) вразлива десеріалізація (Insecure Deserialization) належить до процедур, що беруть участь у відтворенні об’єкта даних із байтового потоку. Небезпечна десеріалізація відбувається, якщо ненадійний код застосовується для створення вразливості або віддаленого виконання коду. Вважається, що недоліки десеріалізації не ведуть до віддаленого виконання коду, але їх все одно можна використати для виконання таких атак, як атаки ескалації привілеїв, ін’єкції та відтворення;

9) загроза застосування вразливих компонентів (Using Components with known vulnerabilities) виникає, якщо використовуються такі компоненти, як фреймворки, бібліотеки та інші програмні модулі з відомими вразливими місцями. Це може призвести до серйозної втрати даних або викрадення сервера;

10) некоректне логування та моніторинг (Insufficient Logging and Monitoring) пов'язані з тим, що зазвичай для виявлення систематичних порушень витрачається багато часу, і це зазвичай ідентифікується дотичними засобами, а не внутрішніми процесами чи моніторингом. Отже, якщо відбувається некоректне ведення журналів та моніторинг, а також відсутність або невдала інтеграція з реагуванням на інциденти, зловмисники можуть надалі атакувати системи, модифікувати або знищувати дані, підтримувати стійкість і переходити до більшої кількості систем.

Необхідно пам'ятати також і про найбільш небезпечні вектори інформаційних атак зловмисників [1], які спричиняють витікання конфіденційних даних.

Пошукова оптимізація. Спосіб поширення злоякісних програм. Якщо відбуваються певні події, інформація про які з'являється в усіх новинах, зловмисники використовують технології SEO, щоб підвищити позиції своїх попередньо заражених web-сторінок у результатах видачі пошукових систем. Якщо користувач зробить у пошуковій системі запит до новини, яка його цікавить, він, одержавши результат, почне рух згідно з одним із верхніх посилань і потрапить на сторінку зловмисника, яка намагатиметься завантажити й запустити на ПЕОМ користувача злоякісну програму завдяки вразливості ПЗ користувача.

Цільовий фішинг (Spear phising). Зловмисники надсилають певним особам (передусім керівникам) у

компанії цільові й реалістичні сценарії (повідомлення), щоб спонукати жертву відкрити злоякісне вкладення або перейти за посиланням на сайт, який містить експлойти для зламування програм на боці користувача.

Перехоплення браузера (browser hooking). Експлуатуючи вразливості, що дозволяють здійснювати міжсайтове виконання сценаріїв на довірених web-сайтах, зловмисники розміщують контент, який містить злоякісні скрипти (сценарії). Коли користувач відкриває такий сайт, то запускає ці крипти та надає зловмиснику контроль над самим браузером користувача. Перехоплений таким чином контроль над браузером користувача дозволяє зловмиснику використати його як точку відліку для подальших атак на інші системи, зокрема на внутрішні ресурси мережі та сервери компанії.

Сайти соцмереж як засіб крадіжки інформації та поширення злоякісних програм. Зловмисники використовують популярні соцмережі типу FaseBook, LinkedIn, Twitter для збирання критичної інформації про діяльність компанії. Більш того, вони поширюють експлойти та скрипти для перехоплення браузера через сайти соцмереж.

Масові SQL-ін'єкції. Упродовж багатьох років атаки, що використовували SQL-ін'єкції, зосереджувалися на крадіжці конфіденційних даних в окремих web-додатках та базах даних. Останнім часом зловмисники розширили спектр використання SQL-ін'єкцій за допомогою автоматизованого програмного забезпечення, яке дозволяє одночасно атакувати тисячі web-додатків. Замість викрадення даних сучасні атаки на основі SQL-ін'єкцій найчастіше намагаються змінити вміст баз даних, які будуть відображатися на web-сайтах, спотворити web-контент, розмістити на сайті злоякісні скрипти для атаки на

браузери користувачів, а також інші експлойти, які використовують уразливості на боці користувача.

Атаки на адміністративні інтерфейси. Більшість великих корпоративних систем, таких як комплекси забезпечення безпеки кінцевих точок, засобів мережевого адміністрування, ERP-системи тощо налаштовуються через адміністративні web-інтерфейси. Виконуючи атаки перехоплення браузера або експлуатуючи уразливості програмного забезпечення на боці користувача, зловмисники все частіше полюють на адміністративні інтерфейси, які можуть забезпечити контроль відповідної системи або інфраструктури.

Атаки «передача хешу» (pass the hash) у Windows інтегровано в пакети для проведення атак. Зловмисники замість паролів використовують техніку «передачі хешу» стосовно Windows-систем, щоб одержати доступ до корпоративного домену. Нині ці можливості є широко використовуваними інструментами комп'ютерних атак, такі як Metasploit та Nmap, що значно спрощує проведення широкомасштабних атак із використанням відповідної техніки.

Зламування обладнання. Оскільки захист ПЗ останнім часом значно поліпшився й водночас набули значного поширення пристрої, такі як смартфони, безпроводові маршрутизатори тощо, то кіберзлочинці частіше почали зламувати обладнання. Через перехоплення інформації, що передається шинами даних (bus sniffing), зламування прошивок, змінювання системного часу (clock glitching) та інші витончені атаки на обладнання зловмисники обминають захисні механізми й одержують ключі шифрування, що допомагає їм під час подальших атак на інфраструктуру компанії-жертви.

Способи виявлення вразливостей у вебдодатках:

– тестування функцій (методи чорного, білого, сірого ящиків);

– фазинг (fuzzing) – автоматична або напівавтоматична техніка тестування програмного забезпечення, що полягає в передаванні додатку на вхід неправильних, несподіваних або випадкових даних;

– аналіз вихідного коду (статичний / динамічний / ручний)

– бінарний аналіз додатка (binary analysis).

Алгоритми захисту вебдодатків

Захист вебдодатків – це процедура захисту онлайн-сервісів та вебсайтів від різноманітних кібернетичних та безпекових загроз, що полегшує загрози в коді програми. Найбільш поширеними цілями для атак вебдодатків є інструменти адміністрування баз даних (наприклад, phpMyAdmin), системи управління вмістом (наприклад, WordPress) та програми SaaS.

Існують різні способи гарантувати безпеку вебдодатків. Нижче наведено найкращі практики, які потрібно впроваджувати:

1) **зробити безпеку частиною процесу розроблення**, а не задумом. Надати кожному веброзробникові можливість підвищення кваліфікації з безпеки додатків.

Першим і найголовнішим кроком для гарантування безпеки вебдодатків є систематичні корпоративні тренінги, вивчення навчальних курсів із безпеки програмного забезпечення на всіх рівнях. Навчання не повинне бути обмежене для розробників вебдодатків, однак залучати весь відповідний персонал, який бере участь у процедурі, наприклад, спеціаліст із контролю

якості, оперативний персонал та управління проєктами. Цей вид навчання для всього персоналу, пов'язаного з життєвим циклом розроблення, допомагає формувати культуру безпеки в компанії;

2) **здійснювати тестування** штучними атаками розробленого вебдодатка для аналізу ступеня безпеки.

3) **розробити тактику безпеки.** Використовувати брандмауер вебдодатків – фільтр для HTTP-трафіку між клієнтом та сервером. Він не допускає жодних зловмисних запитів та не проникає у бази даних. Брандмауер – це найважливіший спосіб захисту програмного забезпечення на вхідних точках у мережу, оскільки він здійснює аналіз вхідного трафіку і зупиняє будь-які сумнівні дії. Брандмауер не вимагає від розробників трансформування у вихідний код, що також робить його придатним для використання. Але традиційні брандмауери мають свої недоліки: вони не можуть визначити деякі види атак;

4) **оновлювати застарілі програми**, підсистеми і модулі. Пріоритетність вебпрограм – логічний наступний крок. Після закінчення інвентаризації існуючих вебпрограм сортування їх за пріоритетом є звуковим кроком. Важливим є сортування програм за категоріями: нормальна, критична, серйозна;

5) регулярно **створювати резервні копії даних.** Краще створити резервну копію всієї інформації вебсайт, або системи. Якщо відбувається будь-яке порушення безпеки або зараження шкідливим програмним забезпеченням, після цього потрібно відновити вебсистему, тому корисно зберігати останню або оновлену версію сайта. Варто зазначити, що більшість постачальників послуг вебхостингу створюватимуть резервні копії зі своїх серверів, якщо це трапиться;

6) **сканувати вебсайт** на наявність уразливостей. Регулярно потрібно перевіряти вебсайт на наявність

загроз та вразливостей, проводити перевірки та сканування безпеки, щоб ураховувати всі можливі вразливості та захист вебпрограми. Крім того, потрібно сканувати систему після кожної зміни, яка була впроваджена у вебдодатку. Важливо, що кілька сканерів безпеки, навіть досвідчені люди, не зможуть виявити всі труднощі безпеки. Сканери мають евристичну чи шаблонну основу, а шкідливе програмне забезпечення постійно розробляється таким чином, щоб його не можна було виявити. Багато сканерів безпеки виявляють зловмисне програмне забезпечення краще за інших;

7) **безпечне використання файлів cookie**. Ці файли cookie надзвичайно зручні як для користувачів, так і для розробників. Вони дозволяють зберігати дані про користувачів вебсайтами, які вони переглядають, щоб майбутні відвідування були швидкими та в деяких випадках надзвичайно персоналізованими. Але файли cookie також можуть використовувати хакери для одержання доступу до захищених зон;

8) **залучати експертів** із безпеки щодо аналізу вразливості, перевірок безпеки та інших потреб безпеки вебдодатків.

Механізми захисту від атак [2]

1. **Багатошаровий захист** – це стратегія безпеки, в якій кілька захисних шарів розміщені через всю інформаційну систему. Це допомагає уникнути прямих атак проти інформаційної системи і даних, оскільки злом одного шару призводить зловмисника лише до наступного рівня.

2. **Управління інцидентами** – це набір певних процесів для ідентифікації, аналізу, присвоювання пріоритетів і рішення інцидентів безпеки для відновлення нормальних сервісних операцій так швидко, як це

можливо, та уникнення майбутнього повторення інциденту.

3. Політика безпеки – це документ або набір документів, який описує управління безпекою, яке буде реалізовано в організації. Процес дослідження вразливостей і помилок проєктування, який відкриває операційну систему та її застосування для атаки або зловживання.

4. Тестування на проникнення – це метод оцінювання інформаційної безпеки системи або мережі симуляцією атаки для пошуку вразливостей, які може використовувати зловмисник. Тестування передбачає активний аналіз конфігурації системи, пошук недоліків проєктування, архітектури мережі, технічних недоліків і вразливостей.

Забезпечення безпеки вебдодатків є постійним і динамічним процесом. Навіть додержуючись усіх найкращих практик безпеки вебдодатків, згаданих вище, ви не можете дозволити собі бути повністю задоволеними. Вам потрібно продовжувати моніторинг, все одно потрібно бути пильним і досліджувати свій вебдодаток на наявність ризиків безпеки та просувати свої заходи безпеки.

СПИСОК ЛІТЕРАТУРИ

1. Інформаційна та кібербезпека: соціотехнічний аспект : підручник / В. Л. Бурячок, В. Б. Толубко, В. О. Хорошко, С. В. Толюпа ; за заг. ред. д-ра техн. наук, професора В. Б. Толубка. – Київ : ДУТ, 2015. – 288 с.

2. Носов В. В. Конспект лекцій з навчальної дисципліни «Кібербезпека» / В. В. Носов. – Харків : Видавництво ХНУВС, 2019. – 25 с.

3. Security architecture for systems providing end-to-end communications / ITU-T Recommendation X.805,

10/2003 [Electronic resource]. – Access mode : https://www.itu.int/rec/T-REC-X.805-200310-I/en.

4. Кібербезпека для спеціальних агентів кіберполіції (лекції). OSCE. 2016. Додаткова: 2. ITU-T Rec. X.805.

5. OWASP / Top 10 Web Application Security Risks [Electronic resource]. – Access mode : https://owasp.org/www-project-top-ten/.

РОЗДІЛ 16
АВТОМАТИЗОВАНІ СИСТЕМИ УПРАВЛІННЯ ІНЦИДЕНТАМИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Т. В. Лаврик

Для сучасних компаній тема інформаційної безпеки є однією з найбільш актуальних та обговорюваних. Це пов'язано зі значною кількістю кібератак та інцидентів на інформаційні ресурси державних і приватних компаній, здійснюваних зловмисниками майже кожного тижня. Така ситуація змушує керівників компаній діяти негайно і розвивати систему управління інформаційною безпекою, одним із важливих процесів якої є управління інцидентами інформаційної безпеки (ІБ). Саме процес управління інцидентами ІБ дозволяє виявляти сліди атак і вторгнень в інформаційне середовище компанії, що, також надає інформацію про реальні уразливі та слабкі місця системи ІБ компанії.

Міжнародний стандарт ISO/IEC 27001 визначає подію та інцидент інформаційної безпеки таким чином [1]:

1) подія інформаційної безпеки – ідентифікований випадок стану системи або мережі, що свідчить про можливе порушення політики інформаційної безпеки або відмову засобів захисту, чи раніше невідома ситуація, яка може бути істотною для безпеки;

2) інцидент інформаційної безпеки – одинична подія або ряд небажаних і непередбачуваних подій інформаційної безпеки, які з великою ймовірністю призводять до компрометації бізнес-інформації та загроз інформаційній безпеці.

Подію розглядають як частину інциденту ІБ. Якщо подія виникає знову і може завдати будь-якої шкоди компанії, то таку подію вважають інцидентом ІБ.

Кожного року список уже відомих інцидентів ІБ доповнюється новими інцидентами, наслідки від яких іноді вражають. Незважаючи на це, всі інциденти ІБ, що вже відомі та лише з'явилися, класифікують із метою подальшого ефективного управління ними.

Найпоширенішими класифікаціями інцидентів ІБ є класифікації за такими ознаками:

– ступенем критичності впливу інциденту на інформаційну систему та ресурси компанії;

– пріоритетами реагування на інцидент;

– чинниками, що спричинили появу інциденту;

– ступенем завданих інцидентом збитків компанії [2].

Попри той факт, що інциденти ІБ розрізняють за ступенем їх критичності впливу на інформаційні ресурси компанії, управління такими інцидентами необхідно здійснювати однаково. Зрозуміло, що процес управління інцидентами, які є ще недостатньо відомими та дослідженими, ускладнюється порівняно з реагуванням на вже відомі інциденти і потребує значно більшого часу на збирання інформації та її аналіз. Ефективне управління інцидентами ІБ зменшує ймовірність їх повторення і, як наслідок, мінімізує завдані компанії збитки.

Управління інцидентами ІБ реалізується послідовністю процесів від реєстрації інциденту до його аналізу та реагування на інцидент. Організація процесу управління інцидентами ІБ ускладнюється за наявності більш потужнішої інформаційної інфраструктури компанії, що вимагає відповідно й більше ресурсів для своєчасного виявлення інцидентів та їх запобігання. Через великий обсяг ручної роботи і відомий людський фактор опрацювання інцидентів може виявитися неповним або призведе до того, що інциденти, критичні для функціонування системи, виявляться поза увагою

відповідальних осіб і до них не буде вжито відповідних заходів. За такої ситуації відстежувати загальну картину активності та подій, що відбуваються в мережі компанії, оперативно виявляти інциденти ІБ та своєчасно реагувати на них неможливо без спеціалізованих інструментів. Крім того, необхідно постійно збирати й аналізувати в реальному часі інформацію про спроби несанкціонованого проникнення з багатьох джерел. Отже, для якісного та комплексного управління інцидентами ІБ необхідне використання автоматизованих програмних рішень.

Сучасний ринок автоматизованих систем управління інцидентами ІБ представлено різними категоріями продуктів:

– системами управління інформаційною безпекою та подіями безпеки;

– системами аналізу мережевого трафіку;

– системами виявлення загроз та реагування на них на кінцевих точках;

– системами автоматизованого реагування на інциденти;

– системами оркестрування, автоматизації та реагування на інциденти.

Кожна із зазначених вище систем має своє призначення і певну роль у процесі управління інцидентами ІБ. Охарактеризуємо ці системи і визначимо їх особливості.

Системи управління інформаційною безпекою та подіями безпеки (Security Information and Event Management, SIEM) об’єднують найважливіші функції засобів управління подіями безпеки і управління інформаційною безпекою. Рішення SIEM реалізують функції збирання і зберігання, оброблення й аналізування зареєстрованих подій безпеки з метою виявлення та фіксації інцидентів, а також перевірки відповідності

системи управління ІБ існуючим вимогам і нормам. Нині SIEM-системи набули найбільшого поширення в середніх та великих компаніях як автоматизоване рішення для управління інцидентами ІБ.

Типова модель архітектури SIEM-системи наведена на рисунку 16.1 [3].

Рисунок 16.1 – Типова архітектура SIEM-системи

Нижній рівень моделі, наведеної на рисунку 16.2, представлено різними платформами, з яких збирається інформація про події (журнали подій серверів та робочих станцій, систем виявлення і запобігання вторгненням, маршрутизаторів та комутаторів, сканерів уразливостей тощо). Колектор являє собою такий компонент, який фактично збирає інформацію про події. Збирання інформації здійснюється за допомогою агентів, які знаходяться на досліджуваній платформі, або без них за допомогою централізованого управління.

Агент – це програмне забезпечення, яке збирає, перетворює та передає записи журналів цільових платформ до колекторів SIEM. Ключовими функціями

агента є фільтрація записів журналів на основі їх типу, їх нормалізація для подальшого порівняння та кореляція з іншими подіями. У безагентних рішеннях сама цільова платформа відправляє записи журналів колектору SIEM. Syslog, який може відправляти дані журналу безпосередньо колектору, є таким прикладом безагентного рішення [4].

Однак існують різноманітні платформи з різними стандартами повідомлень, різними типами даних та семантикою, наприклад, Syslog, NetFlow, SNMP, jFlow, NCSA, ELF тощо. Програмні агенти використовують у тих випадках, коли платформа не передає даних про події до колектора SIEM.

Менеджер загроз безпеки опрацьовує та порівнює події, зібрані колектором у режимі, близькому до реального часу, та повідомляє про виявлені загрози й атаки до центральної консолі управління безпекою. Менеджер загроз повинен зберігати інформацію про події, що сталися за короткий проміжок часу, зазвичай упродовж 30 днів, щоб швидко виявляти загрози і тим самим запобігати інцидентам [3].

Менеджер журналів безпеки зберігає великі обсяги даних. Він може приймати або неопрацьовані журнали, або вже відфільтровані події, що є цікавими для розслідування інцидентів. Також у менеджері журналів передбачається можливість стиснення та довгострокового зберігання даних для проведення подальших розслідувань інцидентів і складання звітів. Зазвичай термін зберігання даних може тривати до 12 місяців [3].

Центральна консоль управління безпекою подає інформацію про події фахівцям відділів, служб або центрів ІБ на користувацькому рівні. Це основний інтерфейс системи для відстежування інформації про події, їх пріоритети, кореляцію подій, загрози, інциденти, історію

записів журналів тощо. Фіксація й аналіз експертами такої інформації дозволяють підвищувати рівень безпеки та захисту компанії [3].

Традиційно SIEM-системи застосовують для накопичення та оперативного опрацювання інформації про події ІБ. Крім того, за допомогою SIEM-систем вирішують такі важливі завдання, як виявлення та дослідження інцидентів ІБ, інвентаризація активів, контроль за захистом інформаційних ресурсів компанії.

Нижче наведено перелік основних подій ІБ, які виявляють за допомогою SIEM-систем:

– шкідливий контент, що доставляється легальним шляхом, зазвичай електронною поштою (спам, фішингові атаки);

– збирання даних про інфраструктуру компанії, визначення доступних сервісів, пошук уразливих вузлів;

– порушення доступності окремих сервісів і систем у цілому (наприклад, у разі DDoS-атаки);

– експлуатація уразливостей у компонентах системи;

– виконання шкідливого коду;

– використання хакерських утиліт, що застосовуються для злому системи або інших протиправних дій;

– витік конфіденційних даних будь-якими комунікаційними каналами;

– порушення політик ІБ компанії [5].

Аналізуючи дослідження [3; 6; 7], присвячені SIEM-системам, необхідно акцентувати на важливих проблемних аспектах їх використання. Будь-яка інформаційна система не є статичною і вона постійно зазнає різних змін, удосконалюється, модернізується. Це призводить до потреби внесення змін у налаштуваннях

SIEM-систем, до зміни існуючих та додавання нових правил кореляції, нормалізації й агрегації.

Однак, незважаючи на такі істотні проблеми, нині SIEM-системи застосовують досить активно різні компанії для оперативного виявлення і реагування на інциденти ІБ. Серед рейтингових систем є такі, як IBM QRadar SIEM, McAfee Enterprise Security Manager, Elastic SIEM, FortiSIEM тощо.

У більшості компаній інформаційна безпека вибудовується на базі SIEM-систем, випускаючи з уваги події, що відбуваються на рівні мережі або на рівні кінцевих пристроїв. Без урахування цих компонентів частина подій ІБ може залишатися непоміченою і невиявленою. У зв'язку з цим тенденція щодо використання лише SIEM-систем для забезпечення інформаційної безпеки компанії змінюється і доповнюється системами аналізу мережевого трафіку, виявлення загроз та реагування на них на кінцевих точках.

Традиційно для відстеження мережевого трафіку використовують системи виявлення вторгнень. Однак вони функціонують лише в режимі реального часу, що унеможливлює перегляд історії інцидентів та атак. Крім того, системи виявлення вторгнень зазвичай не застосовують для виявлення небезпечної та підозрілої активності в інформаційних потоках внутрішньої мережі. Для вирішення таких проблем доцільно застосовувати системи аналізу мережевого трафіку.

Системи аналізу мережевого трафіку (Network Traffic Analysis, NTA) – це нова категорія систем мережевої безпеки, які є не лише джерелами даних для центрів моніторингу ІБ, а й забезпечують можливості ретроспективного пошуку, розслідування інцидентів, централізованої протидії шкідливій активності.

До основних функцій NTA-систем належать: аналіз трафіку як на периметрі мережі, так і всередині інфраструктури, виявлення атак за допомогою поєднання технологій виявлення, допомога в розслідуванні інцидентів.

Типова архітектура NTA-систем вміщує мережевий сенсор, що збирає трафік, сервери централізованого управління та консоль моніторингу. Деякі рішення NTA поєднують можливості пошуку проблем безпеки з автоматизованими завданнями реагування на ризики та пом'якшення наслідків інцидентів. Інструменти такого типу постійно шукають у мережі підозрілі або шкідливі дані, діагностують їх та допомагають нейтралізувати проблему [8].

Отже, виділимо основні особливості NTA-систем.

NTA-системи аналізують трафік як на периметрі мережі, так і всередині інфраструктури. Зазвичай інші системи, що працюють із трафіком (системи виявлення вторгнень, міжмережеві екрани), застосовують на периметрі. Тому, коли зловмисники проникають у мережу, їх дії залишаються непоміченими.

NTA-системи виявляють атаки за допомогою поєднання різних технологій: машинного навчання, поведінкового аналізу, правил детектування, індикаторів компрометації, ретроспективного аналізу, що дозволяють виявляти атаки як на початкових стадіях, так і якщо зловмисник уже проник в інфраструктуру компанії.

Застосування NTA-систем допомагає в розслідуванні інцидентів та проактивному пошуку загроз, що не виявляються традиційними засобами безпеки. NTA-системи зберігають інформацію про мережеві взаємодії, а деякі з них – ще й запис «сирого» трафіку. Такі дані можуть бути корисними джерелами знань під час

розслідування інцидентів та їх локалізації, а також під час перевірки гіпотез у рамках проактивного пошуку [8].

На міжнародному ринку нині популярним є такі рішення, як Cisco Stealthwatch, NetVizura NetFlow Analyzer, ExtraHop Reveal(x), Progress WhatsUp Gold тощо.

Системи виявлення загроз та реагування на них на кінцевих точках (Endpoint Detection and Response, EDR) – це нова категорія систем безпеки, що здатна виявляти шкідливу активність на кінцевих точках: під'єднані до мережі робочі станції, сервери та будь-які пристрої Інтернету речей тощо. На відміну від антивірусних рішень, завданням яких є захист від типових та масових загроз, EDR-системи орієнтовані на виявлення цільових атак та нетипових загроз.

Архітектура EDR-систем представлена серверною частиною та агентами, що встановлюються на кінцеві точки.

Агент відповідає за моніторинг дій користувача, процесів, що запущені на кінцевій точці, та мережевих комунікацій, а також забезпечує передавання інформації серверній частині.

Серверна частина аналізує одержану інформацію за допомогою технологій машинного навчання, порівнює її з базами індикаторів компрометації та іншою доступною інформацією про складні загрози. Якщо EDR-система виявляє підозрілу подію, то вона сповіщає про це фахівців відповідного центру чи служби безпеки.

Акцентуємо на особливостях сучасних EDR-систем, які:

– дозволяють здійснювати збирання інформації з кінцевих точок у режимі реального часу;

– дозволяють записувати та зберігати інформацію про дії користувачів, їх мережеву активність та запущені

на кінцевих пристроях програми з метою подальшого вивчення і дослідження;

– виявляють та класифікують підозрілу активність, про яку оперативно повідомляють центрам чи службам безпеки;

– здійснюють заходи щодо блокування атаки – ізолюють підозрілі файли, зупиняють шкідливі процеси, розривають мережеве з'єднання [9].

Крім того, для підвищення ефективності процесу управління інцидентами EDR-системи можуть інтегруватися з іншими автоматизованими рішеннями, зокрема SIEM-системами.

EDR-системи дозволяють фахівцям з інформаційної та кібербезпеки виконувати проактивний пошук загроз, аналізуючи нетипову поведінку та підозрілу активність на кінцевих пристроях.

Провідними рішеннями серед сучасних EDR-систем нині визначають CrowdStrike Falcon, SentinelOne Endpoint Protection Platform, ESET Enterprise Inspector, Palo Alto Cortex XDR тощо.

За результатами опитування, проведеними компанією Positive Technologies, 58 % фахівців з ІБ визначили найбільш складним та трудомістким завданням під час роботи зі SIEM-системами – налаштування правил кореляції для зменшення помилкових спрацьовувань, а 52 % додали до цього ще й аналіз інцидентів. Крім того, 30 % опитаних зазначили, що налаштування джерел даних та відстежування їх працездатності займає значну частину часу. Ці виявлені проблеми є стимулом для подальшого розвитку SIEM-систем у продукти іншого класу. Нині такими рішеннями є системи автоматизованого реагування на інциденти та системи оркестрування, автоматизації й реагування на інциденти.

Спершу на ринку продуктів з'явилися системи автоматизованого реагування на інциденти (Incident Response Platform, IRP), і лише після деякого розвитку їм на зміну прийшли системи оркестрування, автоматизації та реагування на інциденти (Security Orchestration, Automation and Response, SOAR).

IRP-системи призначені для автоматизації і підвищення ефективності процесів керування, реагування та розслідування інцидентів ІБ.

Системи IRP пропонують набір ключових можливостей з автоматизації процесу керування інцидентами:

1) сценарії реагування;
2) скрипти автоматизації;
3) карту робочого процесу аналітика щодо інциденту [10].

Сценарії реагування (їх також називають playbooks або runbooks) дозволяють в автоматичному режимі реалізувати алгоритм дій, який задано для конкретного типу інциденту в разі спрацьовування певного правила. Сценарій реагування може містити такі дії, як:

– поставлення завдань, відправлення повідомлень, ухвалення рішень;

– дії, спрямовані на блокування атаки та мінімізацію можливих наслідків;

– збирання ключової інформації для розслідування інциденту, відправлення запитів, запит подій щодо інциденту в систему SIEM;

– запуск необхідних сценаріїв реагування.

Зазвичай в IRP-системах для швидкого старту передбачено набір типових сценаріїв реагування, які легко можна адаптувати під специфіку компанії та конкретну структуру команди реагування за допомогою вбудованих засобів IRP.

Скрипти автоматизації дозволяють віддалено здійснювати збирання даних і виконувати певні дії на обладнанні. IRP-системи містять готові скрипти, а також можна створювати власні скрипти будь-якою скриптовою мовою.

Карта робочого процесу аналітика щодо інциденту дозволяє швидко оцінити статус оброблення інциденту, відстежити кількість виконаних уже кроків, дії, що виконуються в даний момент, та оперативно внести необхідні зміни.

Використання перелічених інструментів дозволяє значно підвищити швидкість оброблення інцидентів, збирання необхідних даних і розгортання захисних заходів, порівняно з діями вручну.

SOAR-системи призначені для оркестрування систем безпеки, тобто їх інтеграції, координації та керування ними. Зокрема, SOAR-системи дозволяють збирати дані про події ІБ з різних джерел, обробляти їх та автоматизувати типові сценарії реагування на них. Такі системи, як і IRP-системи, дозволяють автоматизувати і сам процес опрацювання інциденту.

SOAR-системи збирають та опрацьовують дані про події з різних джерел. За допомогою автоматичних сценаріїв або в ручному режимі аналізуються події ІБ та об’єкти, безпека яких порушена. На основі комплексного аналізу одержаних даних SOAR-системи виділяють потенційно небезпечні події, ранжують їх за ступенем критичності, інформують про них фахівців та передають команди іншим ІБ-продуктам. За необхідності SOAR блокують шкідливі процеси або вживають інші заходи, що визначаються політикою безпеки компанії. Звітна інформація про інциденти за різними параметрами, про поточний стан рівня безпеки компанії формується у

вигляді діаграм або інформерів, що оновлюються в режимі реального часу [10].

Відомими на світовому ринку є такі IRP- та SOAR-рішення: Fortinet FortiSOAR, IBM Resilient SOAR, IBM Resilient Incident Response Platform, Cisco SecureX, Cyberbit SOC 3D, Palo Alto Networks Cortex XSOAR, R-Vision Incident Response Platform.

Підсумовуючи вищенаведене, зазначимо, що управління інцидентами ІБ відіграє істотну роль у постійному успіху будь-якої компанії. Цей процес дозволяє компаніям швидко виявляти, аналізувати і вирішувати проблеми безпеки. Однак нині будь-який аналітик ІБ не може конкурувати за швидкістю з автоматизованою системою управління інцидентами ІБ, а іноді й за якістю ухвалених рішень.

Кожна компанія обирає свій комплекс автоматизованих рішень, але потрібно чітко розуміти те коло завдань, які вирішує певний клас систем. Нині фахівці з інформаційної та кібербезпеки виділяють основну трійку автоматизованих засобів, які складають комплекс управління інцидентами ІБ: 1) системи управління інформаційною безпекою та подіями безпеки (SIEM); 2) системи аналізу мережевого трафіку (NTA); 3) системи виявлення загроз та реагування на них на кінцевих точках (EDR). Однак розвиток технологій і поява нових загроз сприяє вдосконаленню існуючих рішень та виходу на ринок нових продуктів, таких як IRP- та SOAR-систем. Водночас, усі розглянуті системи не є взаємозамінними, а разом вони дозволяють підвищити ефективність роботи фахівців центрів, служб та відділів безпеки.

СПИСОК ЛІТЕРАТУРИ

1. ДСТУ ISO/IEC 27001:2015. Інформаційні технології. Методи захисту системи управління інформаційною безпекою. Вимоги (ISO/IEC 27001:2013; Cor 1:2014, IDT). – Київ : УкрНДНЦ, 2016.

2. Карташевский В. Г. Анализ методов и средств выявления инцидентов информационной безопасности / В. Г. Карташевский, А. В. Крыжановский // Вестник УрФО. Безопасность в информационной среде. – 2018. – № 3 (29). – С. 50–54.

3. Swift D. A Practical Application of SIM/SEM/SIEM Automating Threat Identification / D. Swift // SANS Institute. – 2007. – 80 p.

4. Žgela M. Security Information and Event Management – Capabilities, Challenges and Event Analysis in the Complex IT System [Electronic resource] / M. Žgela, I. Penga // Proceedings of the Central European Conference on Information and Intelligent Systems. – Varaždin, Croatia, 2019. – P. 259–266. – Access mode : http://archive.ceciis.foi.hr/app/public/conferences/2019/Proceedings/QSS/QSS4.pdf.

5. Выявление инцидентов ИБ с помощью SIEM: типичные и нестандартные задачи [Електронний ресурс] // Офіційний сайт компанії Positive Technologies. − Режим доступу : https://www.ptsecurity.com/ru-ru/research/analytics/incidents-siem-2020/.

6. Ушатов В. Проблеми оперативного виявлення і реагування на інциденти інформаційної безпеки / В. Ушатов, О. В. Сєвєрінов // Global Cyber Security Forum : матеріали Першого міжнародного науково-практичного форуму, 14–16 листопада 2019 р. – Харків : ХНУРЕ, 2019. – С. 104–105.

7. Pantola A. Normalization of Logs for Networked Devices in a Security Information Event Management System

[Electronic resource] / A. Pantola, R. Yatco, J. D. Pineda // CT Research Symposium. – De La Salle University-Manila, Philippines, 2010. – Access mode : https://www.researchgate.net/publication/286937242_Normalization_of_Logs_for_Networked_Devices_in_a_Security_Information_Event_Management_System.

8. Системы анализа сетевого трафика (NTA) – обзор мирового и российского рынка [Електронний ресурс] // Офіційний сайт проєкту Atni-Malware. – 2020. – Режим доступу : https://www.anti-malware.ru/analytics/Market_Analysis/Global-and-Russian-market-Network-Traffic-Analysis-systems-review.

9. Технология EDR как элемент ядерной триады SOC [Електронний ресурс] // Офіційний сайт проєкту Habr. – Режим доступу : https://habr.com/ru/post/457838/.

10. Incident Response Platform: The Road to Automating IR [Електронний ресурс] // Офіційний сайт компанії Cynet. – Режим доступу : https://www.cynet.com/incident-response-services/incident-response-platform-the-road-to-automating-ir/.

Довідка про авторів

1. Бабій Михайло Семенович, кандидат технічних наук, доцент кафедри комп'ютерних наук, Сумський державний університет, m.babiy@cs.sumdu.edu.ua.
2. Барченко Наталія Леонідівна, кандидат технічних наук, доцент кафедри комп'ютерних наук, Сумський державний університет, n.barchenko@cs.sumdu.edu.ua.
3. Біленький Роман Ігорович, магістрант, факультет комп'ютерних наук, Технологічний університет Вроцлава, bilenkyi.r@gmail.com.
4. Довбиш Анатолій Степанович, доктор технічних наук, професор, завідувач кафедри комп'ютерних наук, Сумський державний університет, a.dovbysh@cs.sumdu.edu.ua.
5. Зарудна Катерина Олександрівна, студентка, Сумський державний університет, zarudna.k@ms.sumdu.edu.ua.
6. Кальченко Вадим Володимирович, асистент кафедри комп'ютерних наук, Сумський державний університет, kavavLa@ukr.net.
7. Коваль Віталій Вікторович, кандидат фізико-математичних наук, старший викладач кафедри комп'ютерних наук, Сумський державний університет, v.koval@oeph.sumdu.edu.ua.
8. Колесніков Валерій Анатолійович, доктор комп'ютерних наук, професор кафедри комп'ютерних наук, Сумський державний університет, v.kolesnikov@cs.sumdu.edu.ua.
9. Котух Євген Володимирович, кандидат технічних наук, доцент кафедри комп'ютерних

наук, Сумський державний університет, yevgenkotukh@gmail.com.

10. Кузіков Борис Олегович, кандидат технічних наук, старший викладач кафедри комп'ютерних наук, Сумський державний університет, b.kuzikov@cs.sumdu.edu.ua.

11. Лаврик Тетяна Володимирівна, кандидат педагогічних наук, старший викладач кафедри комп'ютерних наук, Сумський державний університет, t.lavryk@cs.sumdu.edu.ua.

12. Лавров Євгеній Анатолійович, доктор технічних наук, професор, Сумський державний університет, e.lavrov@cs.sumdu.edu.ua.

13. Маслова Зоя Іванівна кандидат технічних наук, доцент кафедри комп'ютерних наук, Сумський державний університет, z.maslova@cs.sumdu.edu.ua.

14. Москаленко Альона Сергіївна, кандидат технічних наук, старший викладач кафедри комп'ютерних наук, Сумський державний університет, a.moskalenko@cs.sumdu.edu.ua.

15. Москаленко В'ячеслав Васильович, кандидат технічних наук, доцент кафедри комп'ютерних наук, Сумський державний університет, v.moskalenko@cs.sumdu.edu.ua.

16. Ободяк Віктор Корнелійович, кандидат технічних наук, доцент кафедри комп'ютерних наук, Сумський державний університет, v.obodyak@cs.sumdu.edu.ua.

17. Плохута Тетяна Миколаївна, кандидат педагогічних наук, старший викладач кафедри іноземних мов, Сумський державний університет, t.plohuta@el.sumdu.edu.ua.

18. Проценко Олена Борисівна, кандидат фізико-математичних наук, доцент кафедри

комп’ютерних наук, Сумський державний університет, o.protsenko@cs.sumdu.edu.ua.

19. Сидоренко Ольга Павлівна, кандидат педагогічних наук, доцент, Сумський державний університет, o.sadovnikova@journ.sumdu.edu.ua.

20. Теницька Альона Олексіївна, студентка, Сумський державний університет, kb71-15@ssu.edu.ua.

21. Халімов Геннадій Зайдулович, доктор технічних наук, професор, Харківський національний університет радіоелектроніки, hennadii.khalimov@nure.ua.

22. Чибіряк Яна Іванівна, кандидат технічних наук, доцент кафедри комп’ютерних наук, Сумський державний університет, y.chibiryak@cs.sumdu.edu.ua.

23. Шелехов Ігор Володимирович, кандидат технічних наук, доцент кафедри комп’ютерних наук, Сумський державний університет, i.shelehov@cs.sumdu.edu.ua.

24. Шовкопляс Оксана Анатоліївна, кандидат фізико-математичних наук, доцент кафедри комп’ютерних наук, Сумський державний університет, o.shovkoplyas@mss.sumdu.edu.ua.

Наукове видання

Довбиш Анатолій Степанович,
Ободяк Віктор Корнелійович,
Шелехов Ігор Володимирович та ін.

Сучасні інформаційні технології в кібербезпеці

Монографія

За редакцією В. К. Ободяка, І. В. Шелехова

Художнє оформлення обкладинки І. В. Шелехова
Редактори: Н. З. Клочко, С. М. Симоненко, О. В. Федяй
Комп'ютерне верстання В. К. Ободяка

Формат 60×84/16. Ум. друк. арк. 20,23. Обл.-вид арк. 19,88. Тираж 300 пр. Зам. №

Видавець і виготовлювач
Сумський державний університет,
вул. Римського-Корсакова, 2, м. Суми, 40007
Свідоцтво суб'єкта видавничої діяльності ДК № 3062 від 17.12.2007.