КОМПЬЮТЕРНАЯ ИНЖЕНЕРИЯ

УДК 658.512.011: 681.326: 519.713

МОДЕЛІ І МЕТОДИ ЗАХИСТУ КІБЕРПРОСТОРУ

АДАМОВ О.С.

Наводиться аналітичний огляд існуючих моделей, методів і технологій захисту індивідуального сервіс-комп'ютингу. Визначаються переваги і недоліки найбільш затребуваних моделей і методів, опублікованих в спеціальній літературі: матеріалах конференцій і наукових журналах. На основі проведеного аналізу сформульовано мету і задачі дослідження, орієнтовані на усунення проблемних місць і недоліків існуючих моделей і методів у контексті їх реалізації в інфраструктурі захисту індивідуального сервіс-комп'ютингу.

1. Огляд моделей захисту кіберпростору

Розглянемо моделі, що застосовані для захисту кіберпростору:

1. Стримування атак на кіберпростір [1].
2. Захисту персональних даних [2, 3].
3. Управління доступом [4-8].
4. Реагування на інциденти [9-13].
5. Кіберзагроз [14-18].

Модель стримування кібератак на кіберпростір включає контроль над кіберзброєю і кіберрозвідку на національному рівні, захист приватних даних на індивідуальному і корпоративному рівнях [1].

Захист персональних даних представлений регуляторними моделями. Наприклад, General Data Protection Regulation (GDPR) для країн EU [2, 3]. GDPR положення являє собою юридичний інструмент для регулювання обробки персональних даних, що прийнятий в Європейському Союзі. Регламент схвалює стандартизацію. В цьому відношенні можна говорити про моделі захисту даних, з огляду на питання захисту персональних даних, з одного боку, а також етику і відповідальність в контексті кіберпростору – з іншого. Недолік регуляторної моделі в тому, що приписи і стандарти мають на увазі самоконтроль за їх виконанням. Штрафні санкції вводяться тільки в разі, коли витік персональних даних став надбанням громадськості. У більшості випадків подібні інциденти ховаються від користувачів, чиї персональні дані були скомпрометовані.

Існуючі моделі управління доступом Discretionary access control (DAC), Mandatory access control (MAC), Role-based access control (RBAC) описують права доступу для користувача тільки в рамках одного домена або організації [4, 5]. У мультидоменному середовищі можлива ситуація, коли в різних доменах визначені різні ролі і права доступу для різних акаунтів користувача, а також до різних об'єктів інформаційної інфраструктури, наприклад, файлів, процесів, мережевих ресурсів різних обчислювальних кластерів [6 - 8]. Таким чином, дані моделі не можуть бути використані в рамках кіберпростору, де у користувача існують різні уявлення, певні в різних доменах, і немає єдиної провайдераутентифікації для всіх доменів і типів об'єктів.

Традиційні моделі реагування на інциденти, представлені в NIST 800-61 [9], ISO 27035 [10], SANS's Incident Handler's Handbook [11], модель загального реагування на інциденти [12], а також модель Agile [13] містять опис процесів, пов'язаних з виявленням, стримуванням, розслідуванням, аналізом, відновленням і запобіганням інцидентів безпеки в класичній інформаційній інфраструктурі. Однак всі вони не беруть до уваги особливості хмарної інфраструктури, такі як SDN та NFV, масштабованість, відмовостійкість і віртуальний характер середовища.

Моделі кіберзагроз:

1. STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege). Найбільш популярна модель загроз від компанії Microsoft для розробників програмних додатків [14].
2. CAPEC (Common Attack Pattern Enumeration and Classification). Модель від MITRE для побудови периметра захисту організації на основі типових сценаріїв атак [15].
3. Common vulnerability scoring system (CVSS). Модель оцінки загроз через наявність вразливості у системі [16].
4. ATT&CK (Adversary Tactics and Techniques and Common Knowledge) від MITRE. Глобально доступна база знань супротивної тактики та методик, заснована на спостереженнях у реальному світі. База знань ATT&CK використовується як основа для розробки конкретних моделей та методологій загроз у приватному секторі, уряді та у сфері продуктів і послуг кібербезпеки [17].
5. OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) є методологією для раціоналізації та оптимізації процесу оцінки ризиків інформаційної безпеки, щоб організація могла отримати достатні результати з невеликими інвестиціями у час, людей та інші обмежені ресурси. Вона організовує розгляд людей, технологій і засобів у контексті їхнього відношення до інформації та бізнес-процесів і послуг, які вони підтримують [18].
6. Модель розвиненої сталої загрози (Advanced Persistent Threat – APT) або кіберланцюг вбивства (Cyber Kill Chain). Вперше була запропонована компанією Mandiant, яка розкрила цілі та

діяльність глобального актора APT1 [19]. Ці заходи включали в себе крадіжку сотень терабайт конфіденційних даних, у тому числі бізнес-планів, технологічних креслень і результатів тестування щонайменше з 141 організації у різних галузях. Вони оцінили середню тривалість збереження шкідливих програм у цільових організаціях в 1 рік. З того часу зростає список документованих загроз APT із залученням на глобальну сцену потужних суб'єктів, включаючи суб'єктів національних держав. Розуміння мотивів та дій учасників APT відіграє важливу роль у вирішенні цих проблем. Для подальшого розуміння у доповіді Мандіанта також запропоновано модель життєвого циклу APT, відому як кіберланцюг вбивства (cyber kill chain), що дозволяє отримати перспективу про те, які кроки вживає атакуюча сторона задля досягнення своїх цілей. Типова атака APT складається з таких етапів:

1. Кіберрозвідка. Збір даних про цілі.
2. Проникнення. Наприклад, завантаження шкідливого коду за допомогою попутного завантаження (drive-by-download) чи цільової фішингової (spear-phishing) атаки.
3. Фіксація у системі. Встановлення шкідливої програми.
4. Комунікації з центром командування і керування. Для цієї мети можуть використовуватися трояни віддаленого доступу (Remote Administration Tool – RAT).
5. Ескалація привілеїв через експлуатацію вразливостей з метою отримання доступу до ресурсів організації.
6. Внутрішня кіберрозвідка з метою знаходження серверів з цінною конфіденційною інформацією.
7. Скритне переміщення по внутрішній мережі.
8. Ексфільтрація чи вигрузка конфіденційної інформації.
9. Завершення атаки та самоліквідація або повернення до кроку 4 для отримання нових команд.

Таким чином, кіберланцюг вбивства забезпечує посилання для розуміння та відображення мотивів, цілей та дій учасників APT. Наразі існує більше 500 звітів APT [20].

2. Огляд методів захисту кіберпростору

Методи захисту, які використовуються в даний час в інфраструктурі захисту кіберпростору, представленої КСЗІ (Комплексна система захисту інформації) та КСАЗ (Комплексна система антивірусного захисту), можна умовно розділити на групи по використовуваному підходу до детектування кіберзагроз: детерміністичний, імовірнісний.

2.1. Детерміністичний підхід

Детерміністичний підхід можна поділити на методи на основі зловживання, які виявляють поведінку, пов'язану з відомими атаками. Наприклад, такі методи описані в [21].

Методи на основі специфікації [22, 23], які виявляють атаки згідно з політиками, визначеними експертами.

Методи на основі сигнатурних вердиктів використовують сигнатури, послідовності байт, що унікально ідентифікують атаку. Сигнатури зазвичай представлені у вигляді правил, що створюються експертами або спеціальними роботами і використовуються кіберспільнотою. У більшості випадків такі сигнатури можуть ефективно виявляти певну групу відомих атак. Однак ці підписи сприйнятливі до незначної модифікації коду, яка може бути викликана використанням заплутування коду та шифруванням шкідливого коду.

Для швидкого генерування сигнатур на атаки за участю поліформних шкідливих програм використовуються такі методи:

1. Генерація підписів на основі мережі (NSG) [24]. Метод був запропонований як спосіб автоматичного і швидкого створення сигнатур для поліморфних хробаків на основі розробленої моделі NSG-PolyTree. Такі сигнатури і їхні варіанти схожі між собою, а деревоподібна структура може належним чином відображати їхню сімейну подібність.
2. Генерація сигнатури на основі навчання (Learning-based Signature Generation – LSG) включає методи, які шукають єдину велику інваріантну підрядку послідовностей байтів, а також методи, що шукають багато коротких інваріантних підрядків. Методи вилучення зразків є привабливими, тому що сигнатури можуть генеруватися і ефективно узгоджуватися, і більш ранні роботи показали існування інваріантів у експлойтах. Але автори [25] показали фундаментальні обмеження на точність великого класу алгоритмів вилучення зразків у змагальній обстановці.
3. DeepSign. Метод глибокого навчання для автоматичного генерування та класифікації шкідливих програм. Цей метод використовує глибоку мережу переконань (Beep Belief Network – DBN), реалізовану з глибоким стеком автодекодерів, що генерує шуми, генеруючи інваріантне компактне представлення поведінки шкідливого програмного забезпечення.

Незважаючи на те, що для виявлення зловмисних програм звичайні підписи та методи, засновані на маркерах, не виявляють більшість нових варіантів існуючих шкідливих програм, результати, представлені в цій статті, показують, що

сигнатури, створені DBN, дозволяють точно класифікувати нові варіанти шкідливих програм [25]. Проте результати DBN не можуть бути інтерпретовані, порушуючи вимоги GDPR.

Запропонований в [27] підхід націлений на раннє виявлення цільових атак з використанням логічних фільтрів, які в свою чергу приймають на вхід вердикти мережевих сигнатурних сканерів (Firewall L3,4,7 і Network IDS / IPS).

Запропонована методологія забезпечує виявлення індикаторів на ранній стадії таргетованої атаки в мережевому трафіку. Проте даний метод не був протестований на практиці. Автори обмежилися теоретичним моделюванням, що не дозволяє оцінити ефективність даного методу для захисту від реальних кібератак.

Популярним інструментом детектування кіберзагроз є Yara [28], який дозволяє створювати розширені сигнатури на основі регулярних виразів, використовуючи текстові і бінарні рядки в умовних виразах правил.

2.2. Ймовірнісний підхід та застосування машинного навчання

Імовірнісний підхід переважно використовує алгоритми машинного навчання, тому його слід розглядати в контексті методів і моделей, побудованих на основі алгоритмів машинного навчання.

Штучний інтелект (Artificial Intelligence – AI) – це наука, що дозволяє комп'ютеру автоматизувати те, що потрібно людині: інтелект, аналіз і прийняття рішень.

Машинне навчання (Machine Learning) – наука, що дозволяє комп'ютерам вчитися без явно запрограмованого на це. Машинне навчання застосовує статистику і алгоритми масштабування на великих обсягах даних. Одна з цілей машинного навчання – це досягнення штучного інтелекту.

Наука про дані (Data Science) – дисципліна, яка займається витягом інформації з даних. Наука про дані – це широке поле, що включає машинне навчання.

Серед алгоритмів машинного навчання, які застосовуються для вирішення завдань детектування кібератак, автори [29] виділяють такі групи:

1. Supervised: Association Rule Classification (a); Artificial Neural Network (Deep Learning) (b); Support Vector Machines (c); Decision Trees (d); Bayesian Network (e); Hidden Markov Model (f); Kalman Filter (g); Bootstrap, Bagging, and AdaBoost (h); Random Forest (i).
2. Unsupervised: k-Means Clustering (a); Expectation Maximum (b); k-Nearest Neighbor (c); SOM ANN (d); Principal Components Analysis (e); Subspace Clustering (f).
3. Semi-supervised: Generative models (a); Low-density separation (b); Graph-based methods (c).

Алгоритми машинного навчання можуть працювати через навчання з вчителем або без. У навчанні з вчителем (supervised) алгоритм використовує додаткову інформацію та контекст, або дані для навчання надаються окремо, в порядку, щоб машина стала більш розумною. У навчанні без вчителя (unsupervised) алгоритм має всю інформацію та контекст, з якого можна повністю зрозуміти надані навчальні дані до нього, щоб він міг сам вчитися. Існує також комбінований метод, де навчальні дані частково даються алгоритму машинного навчання.

Навчання з вчителем часто необхідно реалізовувати на наборі даних з доброякісними аномаліями, щоб передбачити майбутні аномалії, які не є доброякісними. Але в кібербезпеці високоякісні навчальні дані важко отримати через численні випадки помилкових спрацьовувань. Інженери в центрі операційної безпеки (SOC) повинні переглядати дані навчання і надавати коригування, такі як вказівка певних наборів подій, що представляють вагомі загрози безпеки, а інші – доброякісні. Тому фахівці завжди будуть необхідні для нагляду за навчанням для кібербезпеки.

Далі розглянемо приклади реалізацій методів на основі алгоритмів машинного навчання, що активно застосовуються у кібербезпеці.

Виявлення аномалій (викидів) є виявленням рідкісних подій, які викликають підозри, істотно відрізняючись від більшості даних [30].

Згідно з [31] алгоритми виявлення аномалій існують трьох типів:

1. Виявлення аномалій без вчителя – такі методи виявляють аномалії в наборах даних за умови, що більшість екземплярів у наборі даних є нормальними, шукають приклади, які найменш схожі на більшість даних.
2. Виявлення аномалій із вчителем – такі методи вимагають набір даних, які були б позначені як "нормальні" чи "ненормальні", і передбачають підготовку класифікатора (ключова відмінність від багатьох інших статистичних проблем класифікації полягає в незбалансованому характері виявлення викидів).
3. Виявлення аномалій із напівнаглядом – такі методи конструюють модель, що представляє звичайну поведінку з даного нормального набору даних навчання, а потім перевіряє ймовірність того, що досліджувана модель буде згенерована досліджуваним екземпляром.

Методи, які найчастіше використовуються для виявлення аномалій:

1. Методики на основі щільності: a – k-найближчий сусід [32 – 34]; b – локальний фактор викиду [35]; c – ізоляційні ліси [36].

2. Виявлення викидів для високовимірних даних [37] на основі: a – підпростору [38]; b – кореляції [39, 54]; c – тензора [40].
3. Однокласові машини опорних векторів (Support Vector Machines – SVM) [41].
4. Реплікатор нейронних мереж [42].
5. Байєсові мережі [42].
6. Приховані марковські моделі (HMM) [42].
7. Виявлення аномалій на основі кластерного аналізу [43, 44].
8. Відхилення від правил асоціації та частих наборів.
9. Виявлення аномалій на основі нечіткої логіки.
10. Ансамблі методів: а – беггінг [45, 46]; б – нормалізація балів [47, 48]; в – інші методи визначення різноманітності [49, 50].

Продуктивність різних методів багато в чому залежить від набору даних і параметрів, а методи мають незначні систематичні переваги перед іншими порівняно з багатьма наборами даних і параметрами [51].

Розглянемо на прикладах використання моделей машинного навчання для детектування аномалій.

У статті [52] подано алгоритм з висновком Байєса, який використовує переваги, засновані на сигнатурному методі і детектуванні аномалій. Запропонований підхід дозволяє витягувати патерни SQL-запитів у вигляді регулярного виразу, які можуть бути легко включені в будь-який механізм обробки правил (наприклад, NIDS Snort). Проте даний підхід націлений на створення статичних сигнатур у вигляді правил, які дозволять детектувати лише певний в правилі спектр загроз.

Зіставлення зі зразком (pattern matching) – метод аналізу і обробки структур даних, заснований на виконанні певних умов залежно від збігу досліджуваного значення з тим чи іншим зразком (шаблоном або патерном), яким може бути частина шкідливого коду чи мережевого трафіка.

Дінг, Фанг та Чарленд з Університету Макгілла у своїй роботі [53] вирішують проблеми пошуку схожих частин асемблерного коду у шкідливих програмах, захищених обфускацією, та з використанням оптимізації коду під час компіляції. Пропонується спільно вивчити лексичні семантичні відносини та векторне представлення функцій складання на основі асемблерного коду. Розроблена модель представлення асемблерного коду Asm2Vec. Вона потребує лише коду збірки як вхідних даних і не вимагає попередніх знань, таких як правильне відображення між функціями складання. Метод може знайти і включити багаті семантичні відносини між токенами, що з'являються в коді збірки. Розроблений метод Asm2Vec вивчає векторне представлення функцій складання, відрізняючи його від інших. Asm2Vec не вимагає ніяких попередніх знань, таких як правильне відображення між функціями складання або використаним рівнем оптимізації компілятора. Модель вивчає лексичні семантичні відносини токенів, що з'являються у збірці коду, і представляє функцію складання як внутрішньо зваженої суміші прихованої семантики. Крім функцій складання, вона може застосовуватися на різних гранулярностях послідовностей складання, таких як бінарні файли, фрагменти, основні блоки або функції. Авторами були проведені експерименти з пошуку клонованого коду, у якому були використані різні опції оптимізації компілятора і методи обфускації. Результати показали, що Asm2Vec є точним і надійним проти сильної зміни в асемблерних інструкціях та графіку управління потоком.

Asm2Vec страждає від декількох обмежень. По-перше, він призначений для однієї мови Асемблер. Asm2Vec не застосовується безпосередньо до семантичних клонів через архітектури. Немає спільного лексично-семантичного простору між двома різними мовами Асемблер. По-друге, поточний селективний механізм розширення не може визначити динамічні стрибки, такі як таблиця переходу. По-третє, це обмежена інтерпретативність. Asm2Vec не може пояснити або обґрунтувати свої результати, показуючи клоновані підграфи або доводячи символічну еквівалентність.

Дослідники з університетів Мічиган-Дірборн, Стоуні-Брук та Іллінойс в Чикаго [54] спробували вирішити проблему виявлення поточної кампанії APT (Advanced Persistent Threats) за допомогою *кореляційного аналізу*. APT атака складається з сукупності різнорідних кроків на багатьох хостів протягом тривалого періоду часу, в режимі реального часу та надання аналітику високорівневого пояснення сценарію нападу на основі хост-журналів та сповіщень IPS (Intrusion Prevention System) від підприємства. Існуючі системи IDS / IPS можуть виявляти та виробляти сповіщення про підозрілі події на хості. Проте поєднання цих попереджень низького рівня з метою отримання картини високого рівня поточної кампанії APT залишається серйозною проблемою.

Недоліком розробленої системи HOLMES є аналіз лише логів системних викликів на вузлах мережі. У той час індикатором атаки можуть стати події, інформація про які зберігається в логах файерволів (L3-4, WAF), IDS/IPS чи сервісу аутенфікації, наприклад під час DDoS чи bruteforce атак, які взагалі не є частиною APT моделі ланцюга вбивства.

Метод *асоціативних правил*, використаний у роботі [59], вирішує проблему великої кількості

помилкових позитивних сигналів тривоги, які генеруються у великих інфраструктурах для виявлення вторгнень, ускладнює розділення помилкових оповіщень від реальних атак. Одним із засобів зменшення цієї проблеми є використання метасигналів або правил, які ідентифікують відомі шаблони атак у потоках сигналів тривоги. Очевидний ризик при такому підході полягає в тому, що база правил не може бути повною стосовно кожного профілю справжньої атаки, особливо тих, які є новими. Зараз нові правила відкриваються вручну, процес, який є дорогим і схильним до помилок. Дослідники представляють новий підхід, що використовує видобуток правил асоціації, щоб скоротити час, що минув від появи нового профілю атаки в даних до його визначення, як правило, в інфраструктурі моніторингу організації.

Недоліком методу є обмежений обсяг проведених експериментів та необхідність апріорних знань про атаку задля виявлення асоціацій, побудови ланцюга атаки та генерації правил. Автори також досліджують сигнали тривоги від мережевої системи виявлення вторгнень (Network IDS/IPS) для добування асоціативних правил.

У машинному навчанні та обробці природної мови, тематична модель є типом статистичної моделі для виявлення абстрактних «тем», які відбуваються в наборі документів. Тематичне моделювання є часто використовуваним інструментом видобування тексту для виявлення прихованих семантичних структур у текстовому тілі. Інтуїтивно, враховуючи те, що документ стосується певної теми, можна очікувати, що окремі слова з'являться в документі більш-менш часто [60].

У [61] автори аналізують повідомлення в блогах для різних категорій загроз кібербезпеки, пов'язаних з виявленням кібератак, кіберзлочинів і тероризму. Існуючі дослідження інтелекту зосереджувалися на аналізі новин або форумів для інцидентів кібербезпеки, але лише деякі з них бралися з веб-журналів чи Інтернет блогів. Автори використовують ймовірнісний латентний семантичний аналіз для виявлення ключових слів з веб-журналів кібербезпеки стосовно певних тем. В роботі продемонстровано, як цей метод може представити блогосферу з точки зору тематики з вимірними ключовими словами, таким чином відстежуючи популярні розмови та теми в блогосфері. Застосовуючи імовірнісний підхід, можливо поліпшити пошук інформації в мережі Інтернет і виявлення ключових слів, а також забезпечити аналітичну основу для майбутнього аналізу блогосфери. Недоліком роботи є її спрямованість лише на аналіз блогосфери.

Однак тематичне моделювання можна використати не тільки задля захисту, але ж й для таргетованої атаки. Як це зробити, показали дослідники з SecureData Labs в рамках своєї доповіді на конференції RSA Conference 2019 [62], де вони, проаналізувавши тематику документів жертви, змогли використати цю інформацію для цільової фішингової атаки.

Таким чином, моделі машинного навчання, що побудовані з урахуванням моделей загроз, представляють ефективні інструменти для автоматизації виявлення кібератак на кіберпростір, підвищуючи обороноспроможність організації. При виборі моделі необхідно керуватися не тільки показниками її ефективності, але і типом навчання моделі: з учителем або без; інтерпретованістю результатів і прозорістю моделі – вимоги The EU General Data Protection Regulation (GDPR) [63].

2.3. Атаки на методи машинного навчання

У контексті методів машинного навчання слід згадати, що деякі з них самі по собі вразливі до атак. Так, у роботі [64] наводиться приклад такої атаки. Автори продемонстрували, що спільне машинне навчання та пов'язані з ним методи, такі як федеративне навчання, можуть привести до ненавмисного витоку інформації про навчальні дані учасників через оновлення моделі. Таким чином, це дозволяє розвивати пасивні та активні атаки виводу для використання цього витоку.

Автори [65] стверджують, що моделі глибокого навчання (DL – Deep Learning) є також вразливими до змагальних прикладів, тобто до зловмисно створених вхідних даних. Це призводять до неправильної поведінки цільових моделей DL, що значно ускладнює застосування DL у доменах, чутливих до безпеки. У своїй роботі автори представляють розробку, реалізацію та оцінку DEEPSEC, єдиної платформи, яка має на меті подолати цей недолік. У своїй нинішній реалізації DEEPSEC об'єднує 16 найсучасніших атак з 10 показниками та 13 найсучасніших засобів захисту з 5 показниками оборонної корисності.

В роботі [66] було показано, що машинне навчання та глибокі нейронні мережі можуть бути обдурені атаками ухилення (також називаються прикладами змагальності), тобто малими змінами вхідних даних, які викликають помилкову класифікацію під час тестування. Ця робота висвітлює уразливість методів виявлення шкідливих програм, які використовують глибокі мережі для вивчення з сирих байтів на прикладі атаки на основі градієнта, здатної уникнути нещодавно запропонованої глибокої мережі, пристосованої для цієї мети, лише змінюючи кілька конкретних байтів у кінці кожного зразка

шкідливого програмного забезпечення та зберігаючи при цьому його шкідливу функцію. Таким чином, змагальні зловмисні програмні файли ухиляються від детектування у цільовій мережі з великою ймовірністю, навіть якщо менше 1% їхніх байтів змінено.

Автори [67] представили першу надійну та узагальнюючу систему виявлення та пом'якшення атак на DNN на основі методів, які ідентифікують бекдори, описані в [68–72], і реконструюють можливі тригери атак. Вони ідентифікують кілька методів пом'якшення за допомогою вхідних фільтрів, обрізання нейронів і відривання від навчання. У роботі демонструється їх ефективність за допомогою великих експериментів на різних DNN проти двох типів ін'єкцій, визначених попередньою роботою: атаку з повним доступом до навчальної моделі і троянська атака, керована нейронами, без доступу до моделі навчання. Запропоновані методи також виявляють надійність у відношенні ряду варіантів бекдор атаки.

Недолік методу виявлення атак на DNN є проблема узагальнення за межами поточного домену. Методи виявлення / пом'якшення можуть бути узагальнюючими: інтуїція для виявлення полягає в тому, що інфікована мітка є більш вразливою, ніж неінфіковані мітки, і це не повинно залежити від домену. Проблема адаптації моделі до домену потребує сформулювати процес атаки бекдора і розробити метрику, що вимірює, наскільки вразлива конкретна мітка. Інша проблема запропонованого методу це великий простір потенційних зустрічних заходів зловмисника, які неможливо охопити в рамках одного дослідження.

Таким чином, моделі на основі нейронних мереж (ANN, DNN), володіючи низкою інтерпретованих результатів і стійкістю до атак на моделі машинного навчання, не можуть бути використані як надійний засіб виявлення кібератак реального світу.

2.4. Оцінка методів виявлення кібератак

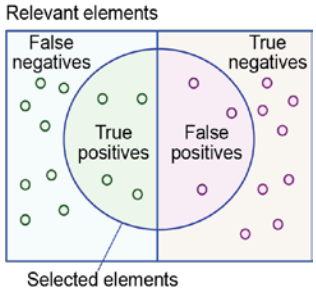
Точність (Precision) та повнота (Recall, Sensetivity, True Positive Rate), які використовуються в бінарній класифікації, є найбільш адекватними критеріями оцінювання для проблем виявлення кібератак.

Для таких проблем *точність* – це міра того, наскільки точним є класифікатор, що виявив атаку. Більша точність відповідає меншій кількості помилкових тривог (FP – False Positives), у той час як *повнота* (Recall) показує, скільки атак класифікатор фактично виявив. Більш високе значення повноти відповідає меншій кількості пропущених атак (FN – False Negative). В ідеалі ми хочемо мати класифікатор з високою точністю і повнотою, оскільки це відповідає низьким значенням FP і FN [73].

Точність та повнота обчислюються за формулами:

$$\text{Precision}=TP/(TP+FP),\ \text{Recall} = TN/(TN+FN), \quad (1)$$

де TP – істинно-позитивне рішення, тобто кількість атак, коректно виявлених класифікатором; TN – істинно-негативне рішення, тобто кількість доброякісних подій, коректно виявлених класифікатором; FP – хибно-позитивне рішення, тобто кількість доброякісних подій, помилково виявлених класифікатором, як кібератака; FN – хибно-негативне рішення, тобто кількість невиявлених класифікатором атак (рисунок).



Діаграма Ейлера, що показує відношення множин рішень класифікатора

У багатокласовому сценарії точність і повнота розраховуються окремо для кожного класу. Щоб розрахувати ці метрики для певного класу, інші класи розглядаються як один клас. Це також називається “одним проти всіх”. Нарешті, точність і повнота для всіх класів об'єднуються разом, використовуючи середньозважену величину [74]. Існують також інші показники, наприклад, F-міра чи міра Ван Ризбергена, але вони рідко застосовуються на практиці для оцінки методів виявлення кібератак, тому ми не будемо їх використовувати у роботі.

3. Аналіз великих даних

Генерація великих об’ємів даних (від 10 ТВ до 1 РВ на день) у комп'ютерних мережах та на вузлах великої організації створює проблему для виявлення кіберзагроз класичними системами протидії кіберзагрозам та перетворює ці традиційні рішення на застарілі.

Хоча аналіз журналів, мережевих потоків і системних подій для криміналістики та виявлення вторгнень був проблемою в спільноті інформаційної безпеки протягом десятиліть, традиційні технології не завжди можуть зберігати аналітичні дані протягом тривалого часу та робити пошук у них.

По-перше, збереження великої кількості даних раніше не було економічно доцільним. Як наслідок, у традиційних інфраструктурах більшість журналів подій та інших записаних

комп'ютерних операцій видалялась після фіксованого періоду зберігання (наприклад, 2-3 місяці). По-друге, виконання аналітичних і складних запитів на великих, неструктурованих наборах даних з неповними або зашумленими атрибутами було неефективним.

Для вирішення цієї проблеми застосовують методи Big Data Analytics (BDA). BDA може допомогти в реальному часі виявляти шкідливі та підозріли дії. Таким чином, ця технологія дозволяє посилити традиційні методи кібербезпеки [75].

Наприклад, BDA дозволяє виявляти банківські шахрайства та застосовувати системи запобігання вторгнень на основі виявлення аномалій (Intrusion Prevention System – IDS). Нові інструменти керування інформацією та подіями безпеки (SIEM) були розроблені для аналізу та управління неструктурованими даними, оскільки вони можуть ефективно чистити, готувати та запитувати дані в різнорідних, неповних та зашумлених форматах даних.

Виявлення шахрайства є одним з найбільш помітних способів використання аналітики великих даних: кредитні картки та телефонні компанії проводили широкомасштабне виявлення шахрайства протягом десятиліть. Однак спеціально побудована інфраструктура, необхідна для розкриття великих даних для виявлення шахрайства, не була достатньо економічною для широкомасштабного прийняття.

Одним з основних наслідків технологій великих даних є те, що вони сприяють широкому колу галузей промисловості для створення доступних інфраструктур для моніторингу безпеки. Великі засоби передачі даних також особливо підходять для того, щоб стати фундаментальними для вдосконаленого виявлення просунутої сталої загрози (Advanced Persistent Threat – APT) та цифрової криміналістики [76].

APTs працюють в повільному режимі, тобто з малою кількістю подій та довгостроковим виконанням. Такі атаки можуть відбуватися протягом тривалого періоду часу, тоді як вторгнення залишається поза увагою жертви. Щоб виявити ці атаки, необхідно зібрати та зіставити велику кількість різноманітних даних, включаючи внутрішні джерела даних та зовнішні спільні дані розвідки, і виконати довгострокову історичну кореляцію для включення апостеріорної інформації про атаку в мережевій історії [77].

Недоліком всіх цих методів є відсутність способів отримання апріорної інформації про кібератаки або критеріїв виявлення аномалій, які можуть відрізнятися залежно від джерела даних.

4. Інфраструктура захисту кіберпростору

Інфраструктура захисту кіберпростору може включати такі компоненти:

1. Системи розмежування доступу до інформації.
2. Криптографічні системи.
3. Системи ідентифікації та автентифікації.
4. Системи аудиту та моніторингу.
5. Системи виявлення та попередження вторгнень.

Розглянемо системи активного захисту від кібератак, до яких можна віднести системи виявлення та попередження вторгнень. Система виявлення та попередження вторгнень може бути встановлена:

1. На кінцевій точці, наприклад, Host Intrusion Prevention System (HIPS) або антивірус.
2. У мережі, наприклад, Network Intrusion Prevention System (NIPS).
3. На контроллері безпеки (у хмарі), який збирає інформацію з хостів та мереж, наприклад Next-Gen SIEM або SOAR.

На сьогоднішній день системи активного захисту від кібератак поєднують з системами аудиту та моніторингу стану кібербезпеки кіберпростору організації з метою виявлення аномалій в агрегованих даних та під час їх обробки, які можуть свідчити про наявність активної кіберзагрози. Найчастіше використовують системи безпеки та управління подіями (Security Information and Event Management – SIEM), такі як: Splunk [55], LogRhythm [56], AlienVault OSSIM [57] та IBM QRadar [58], що допомагають робити кореляцію сигналів тривоги. Ці системи збирають журнал подій і сповіщень з різних джерел і корелюють їх. Таке співвідношення часто використовує доступні індикатори, наприклад, часові мітки.

Нова генерація SIEM рішень включає також оркестрацію інфраструктури безпеки та автоматизоване реагування на інциденти (Security Orchestration, Automation and Response – SOAR).

Ці методи кореляції є корисними, але часто не вистачає розуміння складних відносин, які існують між попередженнями і фактичними випадками вторгнення, і точності, необхідної для узгодження кроків атаки, які відбуваються на різних хостах протягом тривалих періодів часу (тижні, або в деяких випадках місяці).

Тому у SIEM та SOAR системах впроваджуються методи обробки великих даних на основі використання штучного інтелекту (AI) та машинного навчання (ML). Через потребу зберігати великі об’єми даних впродовж 6-12 місяців, а також шукати в них інформацію про потенційну кібератаку, ці сервіси розгортають у хмарному середовищі.

5. Висновки

Проведено аналітичний огляд існуючих моделей, методів і технологій захисту індивідуального сервіс-комп'ютингу. Визначено переваги і недоліки найбільш затребуваних моделей і методів. На основі проведеного аналізу сформульвано мету і задачі дослідження, що орієнтовані на усунення проблемних місць і недоліків існуючих моделей і методів у контексті їх реалізації в інфраструктурі захисту індивідуального сервіс-комп'ютингу.

Мета дослідження – істотне зменшення часу виявлення і блокування кібератак, спрямованих на кіберпростір суб'єкта, шляхом використання розроблених матричних моделей і логічних методів тестування, перевірки та діагностування за рахунок введення обчислювальної надмірності в інфраструктуру кіберпростору. Задачі: 1) Удосконалити структурно-логічні моделі і методи перевірки кіберпростору для тестування і діагностування шкідливих компонентів на основі використання дедуктивного аналізу обчислювальних систем. 2) Розробити сигнатурно-кубітні методи синтезу еталонних логічних схем malware-функціональностей і паралельного моделювання malware-driven великих даних для визначення належності поточного коду до існуючих деструктивних компонентів у malware бібліотеці. 3) Розробити сигнатурно-кубітну модель активного online cyber se-curity комп'ютингу для моніторингу вхідних потоків malware-даних і управління процесом видалення деструктивних компонентів. 4) Удосконалити засоби захисту кіберпростору шляхом логічного тестування і діагностування атак і шкідливих компонентів на основі використання алгоритмів машинного навчання. 5) Розробити метод атрибутно-орієнтованого розпізнавання URL-адрес з використанням частотних паттернів і метод перевірки поліморфних шкідливих програм на основі врахування контрольних сум Portable Executable секцій у виконувані файли і застосування апарату інтелектуального аналізу даних. 6) Виконати тестування і верифікацію розроблених програмних засобів тестування, перевірки та діагностування шкідливих програм шляхом емуляції атак на основі існуючих malware бібліотек.

Функція мети (Z) – мінімізація проміжку часу між моментом запуску атаки (A) на кіберпростір і моментом її діагностування (D), протягом якого обчислювальний сервіс залишається скомпрометованим (C), що одночасно дозволяє поліпшити якість сервісу шляхом забезпечення доступності, цілісності і конфіденційності оброблюваної інформації на період атаки; мінімізація витрат на відновлення працездатності сервісу і фінансових втрат від його простою (TDT) за рахунок введення мінімально необхідної надмірності в інфраструктуру діагностування (I):

$$Z = F(TDT, TC, I) = \min[⅓(TDT + TC + I)],$$

де TC – час, на протязі якого обчислювальний сервіс залишається скомпрометованим з моменту запуску атаки зловмисником;

$$TC = t(D) – t(A),$$

тут t(D) – момент детектування атаки; t(A) – момент запуску атаки.

Література:

1. *Cybersecurity* Dilemmas: Technology, Policy, and Incentives: Summary of Discussions at the 2014 Raymond and Beverly Sackler U.S.-U.K. Scientific Forum, National Academy of Science, 2014.

2. *P De Hert, V Papakonstantinou*, The proposed data protection Regulation replacing Directive 95/46/EC: A sound system for the protection of individuals, Computer Law & Security Review, Elsevier, 2012.

3. *E. Lachaud*, The General Data Protection Regulation and the rise of certification as a regulatory instrument, Computer Law & Security Review, Vol. 34, Issue 2. April 2018. P. 244-256.

4. *Bokefode J.D, Ubale S. A, Apte Sulabha S*, Modani D. G, Analysis of DAC MAC RBAC Access Control based Models for Security, International Journal of Computer Applications, Vol. 104–No. 5, October 2014.

5. *Luo L., He H., Zhu J.* Defect Analysis and Risk Assessment of Mainstream File Access Control Policies. In: Wang G., Ray I., Alcaraz Calero J., Thampi S. (eds) Security, Privacy, and Anonymity in Computation, Communication, and Storage. SpaCCS. Springer. 2016. Lecture Notes in Computer Science. Vol. 10066.

6. *Elsayed W., Gaber T., Zhang N., Ibrahim Moussa* M. (2016) Access Control Models for Pervasive Environments: A Survey. In: Gaber T., Hassanien A., El-Bendary N., Dey N. (eds) The 1st International Conference on Advanced Intelligent System and Informatics (AISI2015), Springer. November 28-30, 2015, Beni Suef, Egypt. Advances in Intelligent Systems and Computing. Vol. 407.

7. *Li, B., Tian, M., Zhang, Y., Lv, S.:* Strategy of domain and cross-domain access control based on trust in cloud computing environment // Computer Engineering and Networking. Springer. 2014. P. 791–798.

8. *Cha, B., Seo, J., Kim, J.:* Design of attribute-based access control in cloud computing environment // Proc. of the International Conference on IT Convergence and Security 2011. P. 41–50.

9. *Computer* Security Incident Handling Guide, NIST 800-61, Sep 2016, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

10. *Information* security incident management (ISO/IEC 27035-1:2016), Sep 2016 https://www.iso.org/obp/ui/#iso:std:iso-iec:27035:-1:ed-1:v1:en

11. *Incident* Handler's Handbook, SANS Institute, Sep 2016,https://www.sans.org/reading-room/whitepapers/incident/incident-handlers-handbook-33901

12. *Felix C. Freiling, Bastian Schwittay,* A Common Process Model for Incident Response and Digital Forensics, IMF 2007, Stuttgart, September 2007, http://www.imf-

confer-
ence.org/imf2007/2%20Freiling%20common_model.pdf
13. *Grispos G., Glisson W. B., Storer T.*, Rethinking Security Incident Response: The Integration of Agile Principles, Sep 2016,
https://arxiv.org/ftp/arxiv/papers/1408/1408.2431.pdf
14. *Shostack A.,* Threat Modeling: Designing for Security, Wiley, 2014, p. 626
15. *CAPEC*: Common Attack Pattern Enumeration and Classification. https://capec.mitre.org/index.html, 2019.
16. *Common* vulnerability scoring system (CVSS) v3.0: Specification document, https://www.first.org/cvss/specification-document, 2019.
17. *Adversary* Tactics and Techniques and Common Knowledge, MITRE, https://attack.mitre.org/, 2019.
18. *Richard A. Caralli, James F. Stevens, Lisa R.* Young, William R. Wilson, Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process, Software Engineering Institute, 2007.
19. *MANDIANT*: Exposing One of China's Cyber Espionage Units. https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf, 2019
20. *Колекція* APT звітів, Github, https://github.com/aptnotes/data, 2019.
21. *Phillip A Porras and Richard A Kemmerer*. Penetration state transition analysis: A rule-based intrusion detection approach // Proc. IEEE Eighth Annual Computer Security Applications Conference, 1992. P. 220–229.
22. *Calvin Ko, Manfred Ruschitzka, and Karl Levitt.* Execution monitoring of security-critical programs in distributed systems: A specification-based approach // IEEE S&P. 1997.
23. *Prem* Uppuluri and R Sekar. Experiences with specification-based intrusion detection // RAID. Springer, 2001.
24. *Yong Tang; Bin Xiao; Xicheng Lu*. Signature Tree Generation for Polymorphic Worms // IEEE Transactions on Computers. 2011. Vol. 60, Issue 4. P. 565 – 579.
25. *Venkataraman S., Blum A., Song D.* Limits of Learning-based Signature Generation with Adversaries. NDSS, The Internet Society, 2008.
26. *David E., Netanyahu N. S.* DeepSign: Deep Learning for Automatic Malware Signature Generation and Classification. International Joint Conference on Neural Networks (IJCNN). Killarney, Ireland, July 2015. P. 1–8.
27. *Japertas S., Baksys T.*, Method of Early Staged Cyber Attacks Detection in IT and Telecommunication Networks, ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 24, NO. 3, 2018.
28. *Yara*, https://virustotal.github.io/yara/, 2019.
29. *Dua S., Du X.*, Data Mining and Machine Learning in Cybersecurity, CRC Press, 2011. P. 23-157.
30. *Zimek A., Schubert E.* Outlier Detection, Encyclopedia of Database Systems, Springer New York, pp. 1–5, 2017.
31. *Chandola, V.; Banerjee, A.; Kumar, V.* Anomaly detection: A survey. ACM Computing Surveys. 2009. 41 (3). P. 1–58.
32. *Knorr, E. M.; Ng, R. T.; Tucakov, V.* Distance-based outliers: Algorithms and applications // The VLDB Journal the International Journal on Very Large Data Bases. 2011. 8 (3–4). P. 237–253.
33. *Ramaswamy, S.; Rastogi, R.; Shim, K.* (2000). Efficient algorithms for mining outliers from large data sets. Proceedings of the 2000 ACM SIGMOD international conference on Management of data – SIGMOD'00. 427 p.
34. *Angiulli, F.; Pizzuti, C.* (2002). Fast Outlier Detection in High Dimensional Spaces. Principles of Data Mining and Knowledge Discovery. Lecture Notes in Computer Science. 2431. p. 15.
35. *Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; Sander, J.* (2000). LOF: Identifying Density-based Local Outliers (PDF). Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. SIGMOD. pp. 93–104.
36. *Liu, Fei Tony; Ting, Kai Ming; Zhou, Zhi-Hua* (December 2008). Isolation Forest. 2008 Eighth IEEE International Conference on Data Mining. P. 413–422.
37. *Zimek, A.; Schubert, E.; Kriegel, H.-P.* (2012). "A survey on unsupervised outlier detection in high-dimensional numerical data". Statistical Analysis and Data Mining. 5 (5): 363–387.
38. *Kriegel, H. P.; Kröger, P.; Schubert, E.; Zimek, A.* (2009). Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data. Advances in Knowledge Discovery and Data Mining. Lecture Notes in Computer Science. 5476. 831p.
39. *Kriegel, H. P.; Kroger, P.; Schubert, E.; Zimek, A.* (2012). Outlier Detection in Arbitrarily Oriented Subspaces. 2012 IEEE 12th International Conference on Data Mining. 379p.
40. *Fanaee-T, H.; Gama, J.* (2016). "Tensor-based anomaly detection: An interdisciplinary survey". Knowledge-Based Systems. 98: 130–147.
41. *Schölkopf, B.; Platt, J. C.; Shawe-Taylor, J.; Smola, A. J.; Williamson, R. C.* (2001). "Estimating the Support of a High-Dimensional Distribution". Neural Computation. 13 (7): 1443–71.
42. Hawkins, Simon; He, Hongxing; Williams, Graham; Baxter, Rohan (2002). "Outlier Detection Using Replicator Neural Networks". Data Warehousing and Knowledge Discovery. Lecture Notes in Computer Science. 2454. pp. 170–180.
43. He, Z.; Xu, X.; Deng, S. (2003). "Discovering cluster-based local outliers". Pattern Recognition Letters. 24 (9–10): 1641–1650.
44. *Campello, R. J. G. B.; Moulavi, D.; Zimek, A.; Sander, J.* (2015). "Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection". ACM Transactions on Knowledge Discovery from Data. 10 (1): 5:1–51.
45. *Lazarevic, A.; Kumar, V.* (2005). Feature bagging for outlier detection. Proc. 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. pp. 157–166.
46. *Nguyen, H. V.; Ang, H. H.; Gopalkrishnan, V.* (2010). Mining Outliers with Ensemble of Heterogeneous Detectors on Random Subspaces. Database Systems for Advanced Applications. Lecture Notes in Computer Science. 5981. p. 368.
47. *Kriegel, H. P.; Kröger, P.; Schubert, E.; Zimek, A.* (2011). Interpreting and Unifying Outlier Scores. Proceedings of the 2011 SIAM International Conference on Data Mining. pp. 13–24.
48. *Schubert, E.; Wojdanowski, R.; Zimek, A.; Kriegel, H. P.* (2012). On Evaluation of Outlier Rankings and Outlier Scores. Proceedings of the 2012 SIAM International Conference on Data Mining. pp. 1047–1058.

49. *Zimek, A.; Campello, R. J. G. B.; Sander, J. R.* (2014). "Ensembles for unsupervised outlier detection". ACM SIGKDD Explorations Newsletter. 15: 11–22.
50. *Zimek, A.; Campello, R. J. G. B.; Sander, J. R.* (2014). Data perturbation for outlier detection ensembles. Proceedings of the 26th International Conference on Scientific and Statistical Database Management – SSDBM '14. p. 1.
51. *Campos, Guilherme O.; Zimek, Arthur; Sander, Jörg; Campello, Ricardo J. G. B.*; Micenková, Barbora; Schubert, Erich; Assent, Ira; Houle, Michael E. (2016). "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study". Data Mining and Knowledge Discovery. 30 (4): 891.
52. *R. Kozik, M. Choraś*, Machine Learning Techniques for Cyber Attacks Detection, Image Processing and Communications Challenges 5, 2014. P. 391-398.
53. *S. H. H. Ding, B. C. M. Fung,* P. Charland, Asm2Vec: Boosting Static Representation Robustness for Binary Clone Search against Code Obfuscation and Compiler Optimization, Proc. of 40th IEEE Symposium on Security and Privacy, 2019.
54. *S. M. Milajerdi, R. Gjomemo, B. Eshete , R. Sekar, V.N. Venkatakrishnan*, HOLMES: Real-Time APT Detection through Correlation of Suspicious Information Flows, Proc. of 40th IEEE Symposium on Security and Privacy, 2019.
55. *Splunk* SIEM, https://www.splunk.com/, 2019.
56. *LogRhythm* SIEM, https://logrhythm.com/, 2019.
57. *AlienVaul*t® OSSIM™, Open Source Security Information and Event Management (SIEM), https://www.alienvault.com/products/ossim, 2019.
58. *IBM* QRadar SIEM, https://www.ibm.com/us-en/marketplace/ibm-qradar-siem, 2019.
59. *Treinen J.J., Thurimella R.* (2006) A Framework for the Application of Association Rule Mining in Large Intrusion Detection Infrastructures. In: Zamboni D., Kruegel C. (eds) Recent Advances in Intrusion Detection. RAID 2006. Lecture Notes in Computer Science, vol 4219. Springer, Berlin, Heidelberg.
60. *Blei, David,* Probabilistic Topic Models. Communications of the ACM. 55 (4), P. 77–84, 2012.
61. *Flora S. Tsai, Kap Luk Chan,* Detecting Cyber Security Threats in Weblogs Using Probabilistic Models, Proc. Intelligence and Security Informatics: Pacific Asia Workshop, PAISI 2007, Chengdu, China, April 11-12, 2007, P. 46-57.
62. *Greeff E., Ross W.* The Rise of the Machines, AI- and ML-Based Attacks Demonstrated, RSA Conference, 2019.
63. *Burt A.* How will the GDPR impact machine learning? Answers to the three most commonly asked questions about maintaining GDPR-compliant machine learning programs, O'Reilly, 2018, https://www.oreilly.com/ideas/how-will-the-gdpr-impact-machine-learning.
64. *Melis L., Song C., Cristofaro E. De, Shmatikov V.*, Exploiting Unintended Feature Leakage in Collaborative Learning // Proc. of 40th IEEE Symposium on Security and Privacy, 2019.
65. *X. Ling, S. Ji, J. Zou, J. Wang, C. Wu, Bo Li, and T. Wang*, DEEPSEC: A Uniform Platform for Security Analysis of Deep Learning Model, Proc. of 40th IEEE Symposium on Security and Privacy, 2019.
66. *Kolosnjaji B., Demontis A., Biggio B., Maiorca D., Giacinto G., Eckert C., Roli F.*, Adversarial Malware Binaries: Evading Deep Learning for Malware Detection in Executables, 2018 26th European Signal Processing Conference (EUSIPCO), 2018, P. 533-537.
67. *B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, Ben Y. Zhao*, Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks, IEEE Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks, 2018.
68. *X. Chen, C. Liu, B. Li, K. Lu, and D. Song*, Targeted backdoor attacks on deep learning systems using data poisoning, arXiv preprint arXiv:1712.05526, 2017.
69. *J. Clements and Y. Lao*, Hardware trojan attacks on neural networks, arXiv preprint arXiv:1806.05768, 2018.
70. *W. Li, J. Yu, X. Ning, P. Wang, Q. Wei, Y. Wang, and H. Yang*, Hu-fu: Hardware and software collaborative attack framework against neural networks, in Proc. of ISVLSI, 2018.
71. *T. Gu, B. Dolan-Gavitt, and S. Garg*, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," in Proc. of Machine Learning and Computer Security Workshop, 2017.
72. *Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang*, Trojaning attack on neural networks, in Proc. of NDSS, 2018.
73. *Powers, David M W.* Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. Journal of Machine Learning Technologies. 2 (1), P. 37–63, 2011.
74. *K.N. Junejo, J. Goh*, Behaviour-Based Attack Detection and Classification in Cyber Physical Systems Using Machine Learning, CPSS '16: Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security, May 2016.
75. *A.A. Cárdenas, P.K. Manadhata, S.P. Rajan.* Big Data Analytics for Security. IEEE Security & Privacy, Volume: 11, Issue: 6, 2013, pp. 74-76.
76. *P. Giura and W. Wang*, Using Large Scale Distributed Computing to Unveil Advanced Persistent Threats. Science J., vol. 1, no. 3, 2012, pp. 93–105.
77. *T.-F. Yen et al.*, Beehive: LargeScale Log Analysis for Detecting Suspicious Activity in Enterprise Networks. Proc. Ann. Computer Security Applications Conference (ACSAC 13), ACM, Dec. 2013.

Надійшла до редколегії 03.05.2019
Рецензент: д-р техн. наук, проф. Дрозд О.В.

Адамов Олександр Семенович, старший викладач кафедри АПОТ ХНУРЕ. Наукові інтереси: кібербезпека. Адреса: Україна, 61166, Харків, пр. Науки, 14, тел. 70-21-326. E-mail: oleksandr.adamov@nure.ua.
Adamov Aleksandr Semenovich, Senior Lecturer, Design Automation Department, NURE. Scientific interests: cybersecurity. Address: Ukraine, 61166, Kharkiv, Nauki Avenue, 14, tel. 70-21-326. E-mail: oleksandr.adamov@nure.ua.