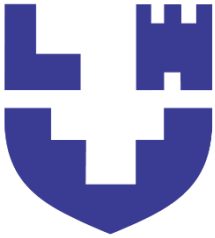
Міністерство освіти і науки України

Луцький національний технічний університет



АНАЛІЗ ВЕЛИКИХ ДАНИХ

Конспект лекцій
для здобувачів першого (бакалаврського) рівня вищої освіти
освітньої програми «Кібербезпека»
галузі знань 12 (F) Інформаційні технології
спеціальності F5 (125) Кібербезпека та захист інформації
денної та заочної форм навчання

Луцьк 2026

УДК 004.6:004.056
А 64

Рекомендовано до видання вченою радою факультету КІТ ЛНТУ,
протокол № _____ від «_____» _____ 20___ року.

Голова вченої ради факультету КІТ _____ Інна КОНДІУС

Електронна копія друкованого видання передана для внесення в репозитарій ЛНТУ

Директор бібліотеки _____ Наталія ПОЛІЩУК

Розглянуто і схвалено на засіданні кафедри кібербезпеки
ЛНТУ, протокол № _____ від « _____» _____ 20____ року.

Завідувач кафедри КБ _____ Сергій КОСТЮЧКО

Укладач: _____ Наталія СУРОВА, доцент кафедри кібербезпеки ЛНТУ

Рецензент: _____ Сергій КОСТЮЧКО, кандидат технічних наук, доцент кафедри кібербезпеки ЛНТУ

Відповідальний за випуск: _____ Сергій КОСТЮЧКО, кандидат технічних наук, доцент кафедри кібербезпеки ЛНТУ

А64 Аналіз великих даних: конспект лекцій для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Кібербезпека» галузі знань 12 (F) Інформаційні технології спеціальності F5 (125) Кібербезпека та захист інформації денної та заочної форм навчання / уклад. Н.М. Сурова. Луцьк: ЛНТУ, 2026. 105 с.

Методичне видання складене відповідно до робочої програми навчальної дисципліни «Аналіз великих даних» з метою використання здобувачами спеціальності F5 (125) Кібербезпека та захист інформації при вивченні даної дисципліни.

ЗМІСТ

ВСТУП 6

ЛЕКЦІЯ 1. АНАЛІЗ ВЕЛИКИХ ДАНИХ У КІБЕРБЕЗПЕЦІ 8

1. Вступ до аналізу великих даних 8

2. Потреба в аналізі великих даних у кібербезпеці 10

3. Архітектура та сценарії застосування 11

4. Виклики та обмеження підходу 13

ЛЕКЦІЯ 2. МЕРЕЖЕВА КРИМІНАЛІСТИКА ТА ЗАСТОСУВАННЯ АНАЛІЗУ ВЕЛИКИХ ДАНИХ 15

1. Предмет і задачі мережевої криміналістики 15

2. Процес розслідування 16

3. Джерела даних, інструменти та їх обмеження 18

4. Застосування аналізу великих даних 19

ЛЕКЦІЯ 3. ДИНАМІЧНЕ ОЦІНЮВАННЯ ВРАЗЛИВОСТЕЙ ТА ЇХ ЕКСПЛУАТАЦІЇ НА ОСНОВІ АНАЛІЗУ ДАНИХ 22

1. Потреба та підхід 22

2. Оцінювання вразливостей та його межі 23

3. Інструменти та джерела даних 25

4. Керування кіберподіями та причинний аналіз 26

ЛЕКЦІЯ 4. АНАЛІЗ КОРЕНЕВИХ ПРИЧИН У КІБЕРБЕЗПЕЦІ 29

1. Коренева причина та атрибуція атак 29

2. Виклики виявлення інцидентів 30

3. Конвеєр причинного аналізу 31

4. Оцінювання якості та обмеження підходу 34

ЛЕКЦІЯ 5. ВІЗУАЛІЗАЦІЯ ДАНИХ ДЛЯ КІБЕРБЕЗПЕКИ 36

1. Зорове сприйняття й канали кодування 36

2. Подання для роботи з алертами 39

3. Керування вразливостями 41

4. Криміналістика та мережевий трафік 41

ЛЕКЦІЯ 6. НАВЧАННЯ З КІБЕРБЕЗПЕКИ: ПРОЄКТУВАННЯ ТА ОЦІНЮВАННЯ 43
1. Особливості навчання з кібербезпеки 43
2. Принципи, що визначають результат 45
3. Практичне проєктування програми 48
ЛЕКЦІЯ 7. МАШИННЕ РОЗНАВЧАННЯ: ВІДНОВЛЕННЯ МОДЕЛЕЙ У ЗМАГАЛЬНОМУ СЕРЕДОВИЩІ 50
1. Потреба систем забувати 50
2. Модель зловмисника 51
3. Цілі та робочий процес рознавчання 52
4. Проміжний рівень сумувань 53
ЛЕКЦІЯ 8. АНАЛІЗ ВЕЛИКИХ ДАНИХ ДЛЯ БЕЗПЕКИ МОБІЛЬНИХ ЗАСТОСУНКІВ 57
1. Постановка задачі та масштаб 57
2. Машинне навчання як засіб сортування черги 58
3. Типові ознаки та алгоритми 59
4. Чому показники не повторюються на практиці 60
ЛЕКЦІЯ 9. БЕЗПЕКА, ПРИВАТНІСТЬ І ДОВІРА В ХМАРНИХ ОБЧИСЛЕННЯХ 64
1. Основи хмарних обчислень 64
2. Загрози безпеці, приватності та довірі 66
3. Засоби захисту та напрями розвитку 68
ЛЕКЦІЯ 10. КІБЕРБЕЗПЕКА ІНТЕРНЕТУ РЕЧЕЙ 71
1. Інтернет речей як предмет кібербезпеки 71
2. Чому різнорідність важить більше за обсяг 73
3. Дев'ять груп вимог до безпеки 74
4. Аналіз великих даних для безпеки інтернету речей 75
ЛЕКЦІЯ 11. АНАЛІЗ ВЕЛИКИХ ДАНИХ ДЛЯ БЕЗПЕКИ ТУМАННИХ ОБЧИСЛЕНЬ 77
1. Чому хмарних обчислень виявилося замало 77
2. Основи туманних обчислень 78

3. Межа передавання даних і три компроміси 80
4. Аналітика для безпеки туманного рівня 81
ЛЕКЦІЯ 12. АНАЛІЗ ДЕВІАНТНОЇ СОЦІОТЕХНІЧНОЇ ПОВЕДІНКИ 83
1. Що таке девіантна соціотехнічна поведінка 83
2. Два інструменти та їх поєднання 84
3. Методика дослідження 85
4. Межі висновку та етичні обмеження 88
ЛЕКЦІЯ 13. ЗАСОБИ КІБЕРБЕЗПЕКИ ТА МЕТОДИКА ЇХ ДОБОРУ 90
1. Добір засобів як інженерна задача 90
2. Засоби межі мережі та моніторингу 91
3. Захист пам'яті та керування автентифікацією 92
4. Методика добору 93
ЛЕКЦІЯ 14. ДАНІ ТА ДОСЛІДНИЦЬКІ ІНІЦІАТИВИ ДЛЯ АНАЛІЗУ КІБЕРБЕЗПЕКИ 97
1. Три групи джерел даних 97
2. Еталонні набори та сховища даних 93
3. Чому вік набору визначає придатність результату 100
4. Обмін даними та паспорт набору 101
СПИСОК ЛІТЕРАТУРИ 104

ВСТУП

Обсяг даних, що їх породжують сучасні інформаційні системи, зріс настільки, що засоби захисту, розраховані на перегляд окремих записів людиною, перестали бути достатніми. Мережева телеметрія, журнали операційних систем і застосунків, записи автентифікації, дані хмарних сервісів та пристроїв інтернету речей утворюють потоки, які неможливо ані зберігати, ані опрацьовувати традиційними засобами. Водночас саме в цих потоках міститься відповідь на питання, чи відбувається атака, як вона розвивається і які наслідки вже настали.

Навчальна дисципліна «Аналіз великих даних» формує в здобувача здатність працювати з такими потоками: добирати джерела даних під поставлене завдання, зводити різнорідні записи до спільної системи координат, застосовувати методи виявлення аномалій і машинного навчання, а головне — відрізняти обґрунтований висновок від правдоподібного.

Мета вивчення дисципліни — сформувати теоретичні знання та практичні навички застосування технологій аналізу великих даних для виявлення, розслідування та запобігання кіберінцидентам, а також уміння оцінювати межі застосовності одержаних результатів.

Завдання вивчення дисципліни:

- засвоїти означення та характеристики великих даних, архітектуру систем аналітики безпеки й місце кожного її рівня;
- навчитися добирати джерела даних за питанням, на яке потрібна відповідь, і усвідомлювати межу спостережуваності кожного джерела;
- опанувати методи виявлення аномалій, кореляції різнорідних подій та аналізу кореневих причин;
- набути навичок візуального подання даних безпеки й оцінювання якості моделей виявлення;
- з’ясувати особливості захисту хмарних, туманних і вбудованих середовищ та мобільних застосунків;

- навчитися оцінювати придатність набору даних і моделі до застосування, зокрема виявляти старіння даних та потребу в рознавчанні моделей;
- сформувати вміння формулювати висновок разом із межами його застосовності.

Конспект складається з чотирнадцяти лекцій. Кожна лекція містить мету, план, перелік ключових термінів, теоретичний матеріал із таблицями та рисунками й висновки.

ЛЕКЦІЯ 1. АНАЛІЗ ВЕЛИКИХ ДАНИХ У КІБЕРБЕЗПЕЦІ

Мета лекції: з'ясувати, чому традиційні засоби захисту втрачають ефективність у сучасному ландшафті загроз, і розібрати архітектуру та типові сценарії застосування аналізу великих даних у кібербезпеці.

План лекції: 1. Вступ до аналізу великих даних. 2. Потреба в аналізі великих даних у кібербезпеці. 3. Архітектура та сценарії застосування. 4. Виклики та обмеження підходу.

Ключові терміни: великі дані, 5V, розподілене зберігання, schema-on-read, потокова обробка, озеро безпекових даних, збагачення контекстом, поведінковий базис, виявлення аномалій, парадокс базового рівня, дрейф понять.

1. ВСТУП ДО АНАЛІЗУ ВЕЛИКИХ ДАНИХ

1.1 Означення та характеристики

Термін «великі дані» (Big Data) описує набори даних, обсяг, швидкість надходження та різнорідність яких перевищують можливості традиційних засобів зберігання й обробки. Означення відносне: критерій не в абсолютних терабайтах, а в тому, чи справляється наявний інструментарій. Аналіз великих даних — це сукупність технологій, архітектурних рішень і методів, що дають змогу опрацьовувати такі набори з прийнятними витратами часу й ресурсів. Прояв кожної характеристики в задачах кібербезпеки показано в таблиці 1.1.

За глибиною висновків розрізняють чотири рівні аналітики: описова відповідає на питання «що сталося», діагностична — «чому це сталося», прогнозна — «що станеться», приписова — «що робити». Зрілий центр безпеки рухається від першого рівня до четвертого, і саме технології великих даних роблять цей рух економічно можливим.

Таблиця 1.1 – Характеристики великих даних у контексті кібербезпеки

Характеристика	Зміст	Прояв у кібербезпеці
Volume (обсяг)	Дані не вміщуються на одному вузлі, потрібне розподілене зберігання	Центр безпеки середнього підприємства обробляє 10–50 тис. подій/с; захоплення пакетів на каналі 10 Гбіт/с дає до 4,5 ТБ за годину
Velocity (швидкість)	Дані надходять безперервним потоком, цінність швидко спадає	DDoS-атака розгортається за десятки секунд; шифрування файлів програмою-вимагачем — за хвилини. Реакція через добу після інциденту вже не має сенсу
Variety (різноманіття)	Структуровані, напівструктуровані та неструктуровані формати одночасно	NetFlow, PCAP, syslog, журнали подій Windows, телеметрія EDR, DNS, HTTP-проксі, журнали застосунків, дані каталогу ідентифікацій, фіди індикаторів
Veracity (достовірність)	Дані неповні, суперечливі, можуть бути спотворені	Пропуски через перевантаження колектора, розсинхронізований час на джерелах, дублікати; зловмисник може навмисно очищати або підробляти журнали
Value (цінність)	Аналіз має давати вимірюваний результат	Скорочення середнього часу виявлення (MTTD) і реагування (MTTR), зниження частки хибних спрацювань, ширше покриття технік MITRE ATT&CK

1.2 Відмінності від традиційної аналітики

Різниця між двома підходами не зводиться до обсягу даних — вона стосується принципів побудови системи: централізованого сховища проти розподіленого, вертикального масштабування проти горизонтального, схеми, визначеної до запису, проти схеми, застосованої під час читання.

Механізм schema-on-read передбачає, що дані записуються у первинному вигляді, а структура застосовується під час читання: нове джерело журналів підключають без міграції схеми. Платою за гнучкість є відсутність контролю якості на вході, тому сирі записи нормалізують і зводять до єдиної схеми — галузевими стандартами є ECS, OCSF і CIM. Уніфікація дозволяє писати одне правило кореляції замість окремого правила для кожного виробника обладнання.

У кібербезпеці розрізняють три класи вимог до затримки: реального часу (менш ніж секунда) — автоматичне блокування з'єднання; майже реального часу (секунди — хвилини) — кореляція подій і формування алерту; пакетний режим

(години) — навчання моделей і ретроспективний пошук. Практичні системи поєднують усі три, що відображено в архітектурах Lambda і Kappa.

2. ПОТРЕБА В АНАЛІЗІ ВЕЛИКИХ ДАНИХ У КІБЕРБЕЗПЕЦІ

2.1 Обмеження традиційних механізмів захисту

Традиційні засоби захисту будувалися на двох припущеннях: загрози можна описати переліком відомих зразків, а мережу — розділити на довірений внутрішній простір і недовірене зовнішнє середовище. Обидва припущення сьогодні хибні. **Сигнатурний підхід** не бачить уразливостей нульового дня, поліморфного коду й технік «життя за рахунок системи», коли атака виконується штатними інструментами — PowerShell, WMI, PsExec, certutil.

Коротке вікно зберігання: «гаряче» зберігання становить 30–90 діб, тоді як підготовлені зловмисники перебувають у мережі місяцями. **Ізольованість джерел:** мережева, хостова та прикладна телеметрія зберігаються окремо, тому багатоетапна атака не збирається в єдину картину. **Відсутність контексту:** подія «передано 5 ГБ на зовнішній вузол» без відомостей про роль користувача та критичність активу не інтерпретовна. Сукупним наслідком є «втома від алертів»: аналітик отримує тисячі повідомлень за зміну, більшість із яких хибні, і справжній інцидент лишається непоміченим.

2.2 Еволюція ландшафту загроз і нові можливості

Цілеспрямовані тривалі атаки (APT) виконуються поетапно — розвідка, доставка, експлуатація, закріплення, командування й управління, латеральне переміщення, досягнення мети; ці етапи описано моделлю ланцюга ураження та систематизовано в матриці MITRE ATT&CK. Кожен окремий етап може виглядати як звичайна активність, тому виявити атаку можна лише за послідовністю. Атаки на ланцюг постачання компрометують довіреного

постачальника, після чого шкідливий код потрапляє до жертв через легітимний механізм оновлення з дійсним підписом.

Звідси набір можливостей, які відкриває аналіз великих даних: тривале зберігання з ретроспективним пошуком за архівом 12–24 місяців; поведінкова аналітика користувачів і сутностей (UEBA) з індивідуальними базовими профілями замість фіксованих порогів; кореляція гетерогенних джерел в єдиний граф «користувач — вузол — процес — домен»; машинне навчання для виявлення невідомих раніше аномалій; оцінювання ризику замість бінарного «алерт/не алерт»; автоматизація реагування (SOAR).

3. АРХІТЕКТУРА ТА СЦЕНАРІЇ ЗАСТОСУВАННЯ

3.1 Логічна архітектура аналітики безпеки

Більшість організацій будує захист із кількох взаємодоповнювальних систем: озеро безпекових даних, SIEM нового покоління, UEBA, засоби аналізу мережевого трафіку (NTA/NDR), платформи керування індикаторами компрометації та засоби автоматизації реагування. Незалежно від набору продуктів, логічна архітектура має шість послідовних рівнів і низку наскрізних функцій (рисунок 1.1).



Рисунок 1.1 – Логічна архітектура аналітики безпеки на основі великих даних

Рівень джерел охоплює мережеву телеметрію (NetFlow/IPFIX, DNS, HTTP-проксі, захоплені пакети), хостову, прикладну, хмарні й промислові джерела; не підключене джерело означає сліпу зону, яку жодна аналітика не компенсує. **Рівень збору й транспорту** забезпечує надійну доставку через шину повідомлень, що виконує роль буфера. **Рівень нормалізації та збагачення** зводить сирі записи до єдиної схеми й доповнює їх контекстом із довідників.

Рівень зберігання реалізує багаторівневу модель: гарячий (0–30 діб, індексоване сховище, доступ за секунди) — для моніторингу; теплий (1–6 місяців, колонкова база) — для розслідувань; холодний (6–24 місяці, Parquet в об'єктному сховищі) — для ретроспективного пошуку; архівний — для нормативних вимог. **Рівень аналітики** поєднує правила, статистичні методи, машинне навчання й графову аналітику, а **рівень подання й реагування** відповідає за дашборди та пріоритезацію алертів.

3.2 Три базові сценарії

Тривале зберігання й ретроспективний пошук. PCI DSS вимагає зберігати журнали не менше 12 місяців, із них щонайменше три — у негайно доступному вигляді; NIS2 встановлює вимоги до реєстрації інцидентів; ISO/IEC 27001 містить окремий захід щодо ведення журналів. Типовий сценарій: партнер повідомляє про новий домен командного сервера, аналітик шукає його за архівом DNS-запитів за 18 місяців, знаходить 14 звернень із двох робочих станцій вісім місяців тому й розгортає розслідування.

Збагачення контекстом — доповнення події відомостями з довідників безпосередньо в конвеєрі: інвентаризація активів, каталог ідентифікацій, геолокація, passive DNS, репутаційні бази, результати сканувань. Сирий запис журналу проксі містить лише час, адресу, ім'я користувача, домен і обсяг байтів; після збагачення додаються підрозділ і критичність активу, рівень небезпеки домену, ознака нічного часу — і подія стає підставою для рішення. Контекст дозволяє відрізнити нічне вивантаження адміністратором під час резервного копіювання від такої самої дії співробітника у день звільнення.

Виявлення аномалій. Аномалія — це спостереження, що суттєво відхиляється від моделі нормальної поведінки. Розрізняють точкові (значення вибивається з розподілу), контекстні (нормальне саме по собі, але не для цього часу чи користувача) і колективні, до яких належать маячки командних серверів і повільна ексфільтрація. Методи поділяють на статистичні (три сигми, робастний z-показник, EWMA), засновані на відстані та щільності (k-NN, LOF, DBSCAN), ізоляційні (Isolation Forest), межові (One-Class SVM) і реконструктивні (автокодувальники).

Якість детекторів оцінюють точністю, повнотою, F-мірою та часткою хибнопозитивних спрацювань. Тут украй важливий парадокс базового рівня: якщо система обробляє 10 мільйонів подій на добу, то навіть за FPR = 0,1 % вона дасть 10 тисяч хибних алертів, серед яких справжні інциденти будуть непомітними. Тому детектори будують каскадами, їх результати корелюють, а порядок роботи аналітика визначає підсумкова оцінка ризику.

4. ВИКЛИКИ ТА ОБМЕЖЕННЯ ПІДХОДУ

Якість і повнота даних: пропущені події, розсинхронізований час і неохоплені джерела знецінюють будь-які подальші моделі. **Вартість інфраструктури** потребує обґрунтованої політики ретенції. **Дефіцит кадрів:** потрібне поєднання компетенцій аналітика безпеки, інженера даних і фахівця з машинного навчання. **Інтерпретовність:** «чорна скриня» не дозволяє ані обґрунтувати рішення, ані виправити модель. **Змагальний вплив:** зловмисник може отруювати навчальну вибірку або лишатися під порогом виявлення.

ВИСНОВКИ

1. Критерієм «великих» даних є не абсолютний обсяг, а неспроможність традиційного інструментарію його опрацювати.

2. Аналіз великих даних відрізняється розподіленим зберіганням, горизонтальним масштабуванням, підходом schema-on-read і обробкою потоків майже в реальному часі.

3. Захист, побудований на сигнатурах і периметрі, не відповідає ландшафту багатоетапних тривалих атак, внутрішніх порушників і шифрованого трафіку.

4. Типова архітектура аналітики безпеки містить шість рівнів — від джерел даних до подання й реагування — та наскрізні функції керування даними й безпеки.

5. Тривале зберігання, збагачення контекстом і виявлення аномалій доповнюють одне одного, проте стримуються якістю даних, вартістю та хибними спрацюваннями, тому аналіз великих даних доповнює, а не замінює традиційні засоби захисту.

ЛЕКЦІЯ 2. МЕРЕЖЕВА КРИМІНАЛІСТИКА ТА ЗАСТОСУВАННЯ АНАЛІЗУ ВЕЛИКИХ ДАНИХ

Мета лекції: засвоїти чотирифазний процес розслідування мережевого інциденту, розібрати джерела даних та інструменти, з'ясувати межі їх застосування й показати, як технології великих даних знімають ці обмеження.

План лекції: 1. Предмет і задачі мережевої криміналістики. 2. Процес розслідування. 3. Джерела даних, інструменти та обмеження традиційних технологій. 4. Застосування аналізу великих даних і референсна архітектура платформи.

Ключові терміни: мережева криміналістика, цифровий доказ, ланцюг передавання доказів, летка інформація, запис потоку, повне захоплення пакетів, точка спостереження, часова шкала інциденту, атрибуція, розмежування доступу на рівні комірки.

1. ПРЕДМЕТ І ЗАДАЧІ МЕРЕЖЕВОЇ КРИМІНАЛІСТИКИ

Мережева криміналістика — це галузь цифрової криміналістики, предметом якої є збирання, збереження, дослідження та аналіз даних про мережевий трафік з метою встановлення обставин інциденту безпеки. На відміну від криміналістики носіїв інформації, яка працює з диском чи пам'яттю окремого пристрою, мережева криміналістика відновлює картину подій за слідами в каналах зв'язку, на маршрутизаторах, проксі-серверах і засобах моніторингу.

Її практичне значення визначається простим спостереженням: будь-яка віддалена атака неминуче залишає слід у мережі. Зловмисник може очистити журнали на скомпрометованому вузлі, змінити мітки часу файлів і видалити знаряддя атаки — але він не може забрати назад пакети, які вже пройшли через мережеве обладнання й були зафіксовані незалежними засобами спостереження. Саме тому мережеві дані часто виявляються єдиним джерелом, якому можна довіряти після компрометації.

Типові задачі галузі: реконструкція інциденту від першого контакту до досягнення мети; визначення точки входу та обсягу компрометації; оцінка збитку; атрибуція за інфраструктурою та почерком; формування доказової бази; зворотний зв'язок для системи захисту.

Головна технічна складність — **леткість** мережевих даних. Пакет існує в каналі мікросекунди, і якщо його не зафіксовано, він зникає безповоротно. На каналі 10 Гбіт/с повне захоплення дає близько 4,5 ТБ за годину, тобто понад сто терабайтів на добу, тому зберігати все довго технічно й економічно неможливо. Практика розв'язує цей конфлікт компромісом: повні пакети зберігають від годин до кількох діб, а метадані про з'єднання — місяцями або роками.

2. ПРОЦЕС РОЗСЛІДУВАННЯ

Розслідування виконується як послідовність чотирьох фаз (рисунок 2.1). Поділ на фази не є формальністю: кожна має власні вимоги до якості роботи, і помилка на ранній фазі не виправляється на пізніших. Зокрема, дані, зібрані без фіксації цілісності, вже не стануть доказом, хоч би яким досконалим був подальший аналіз.



Рисунок 2.1 – Чотири фази процесу мережевої криміналістики

Фаза 1. Збір даних. Слідчий визначає, які джерела здатні містити сліди події, і отримує з них дані. Порядок збирання підпорядковується правилу спадання

леткості: спочатку фіксують те, що зникне найшвидше (активні з'єднання, вміст пам'яті), потім — довговічніші джерела. Обов'язковими діями є обчислення криптографічного хешу кожного зібраного файлу, запис часу й обставин отримання та відкриття журналу передавання доказів; працювати надалі слід із копією, а оригінал зберігати незмінним. Окремої уваги потребує синхронізація часу: якщо годинники джерел розходяться, побудувати достовірну хронологію неможливо.

Повнота зібраного матеріалу визначається не якістю аналізатора, а тим, де фізично розміщено **точки спостереження**. Периметр бачить увесь обмін із зовнішнім світом, але не бачить переміщення між внутрішніми вузлами; ядро мережі показує з'єднання між сегментами, але не обмін усередині сегмента; агент на кінцевому вузлі бачить трафік до розшифрування, але лише для одного пристрою. Жодна окрема точка не дає повної картини: периметрового спостереження достатньо, щоб побачити проникнення й виведення даних, але недостатньо, щоб відновити латеральне переміщення — а саме воно визначає справжній обсяг компрометації.

Фаза 2. Дослідження даних. Зібране приводять до придатного для аналізу вигляду: розбирають формати, зводять поля різних джерел до єдиної схеми, відновлюють сесії з окремих пакетів, відсіюють нерелевантний трафік і будують індекси. Тут-таки оцінюють повноту матеріалу: скільки записів не вдалося розібрати, чи немає розривів у часовому покритті. Ця фаза найбільш трудомістка й найгірше піддається автоматизації традиційними інструментами — саме тут технології великих даних дають найбільший виграш.

Фаза 3. Аналіз даних. Слідчий шукає відповіді на питання «що сталося, коли, з чиєї ініціативи та які наслідки», застосовуючи кореляцію подій різних джерел за часом і адресами, порівняння з базовою поведінкою мережі, пошук за індикаторами компрометації та побудову графа зв'язків. Методологічна вимога — перевірність: кожен пункт хронології має спиратися на конкретні записи. Слід свідомо шукати спростування власної версії: зловмисники залишають хибні сліди, а збіг за часом не означає причинного зв'язку.

Фаза 4. Візуалізація та звітування. Результат подають у формі, зрозумілій адресатові: часові шкали, графи зв'язків, зведені таблиці. Звіт містить опис методики, перелік джерел, установлені факти з посиланням на докази, окремо висновки та окремо припущення. Процес не є суто лінійним: нові запитання повертають слідчого до збору додаткових даних, тому на рисунку 2.1 показано зворотний зв'язок від останньої фази до першої.

3. ДЖЕРЕЛА ДАНИХ, ІНСТРУМЕНТИ ТА ЇХ ОБМЕЖЕННЯ

3.1 Джерела та інструментарій

Мережева криміналістика спирається на кілька принципово різних джерел: повне захоплення пакетів (заголовки й корисне навантаження, строк зберігання — години та доби); записи потоків NetFlow, IPFIX, sFlow (метадані з'єднань — місяці та роки); журнали систем виявлення вторгнень, проксі та DNS; журнали DHCP і каталогу ідентифікацій, що дають відповідність адреси пристрою й користувача в конкретний момент. Практично важливим є вибір між першими двома (таблиця 2.1).

Таблиця 2.1 – Порівняння повного захоплення пакетів і записів потоків

Ознака	Повне захоплення (PCAP)	Записи потоків (NetFlow)
Глибина деталізації	Повний вміст, включно з корисним навантаженням	Лише метадані з'єднання
Обсяг на каналі 10 Гбіт/с	До 4,5 ТБ за годину	Одиниці — десятки ГБ за добу
Практичний строк зберігання	Години — кілька діб	Місяці — роки
Цінність зашифрованого трафіку	Низька: вміст недоступний	Не змінюється: метадані не шифруються
Вплив на приватність	Високий: видно зміст листування	Помірний: видно факт і обсяг зв'язку
Типове застосування	Детальний розбір нещодавнього інциденту	Хронологія, ретроспективний пошук, базові профілі

Велика організація використовує обидва джерела одночасно: потоки як «пам'ять» про те, хто з ким з'єднувався за останній рік, і повні пакети як «мікроскоп» для нещодавніх подій. Інструментарій поділяється на чотири групи:

захоплення й розбір пакетів (tcpdump, Wireshark, NetworkMiner), збирання та аналіз потоків (nfdump, SiLK, Argus, ntopng), системи виявлення вторгнень (Snort, Suricata, Zeek) і засоби моніторингу мережі. Окремо варто відзначити **Zeek**: він перетворює трафік на структуровані журнали подій за протоколами — з'єднання, запити DNS, сертифікати TLS, передані файли.

3.2 Обмеження традиційних технологій

Перелічені інструменти створювалися для мереж минулих десятиліть і стикаються з обмеженнями, які не усуваються ані налаштуванням, ані потужнішим сервером. **Масштаб:** більшість класичних інструментів працює на одному вузлі, і спроба відкрити файл захоплення на кількасот гігабайтів у графічному аналізаторі закінчується вичерпанням пам'яті. **Вертикальне масштабування:** вартість потужнішого сервера зростає нелінійно, тоді як обсяг трафіку зростає щороку.

4. ЗАСТОСУВАННЯ АНАЛІЗУ ВЕЛИКИХ ДАНИХ

4.1 Проєктні рішення

Екосистема великих даних пропонує обчислювальні рушії (Spark, Hadoop), засоби потокової обробки (Storm, Flink), шини повідомлень і нереляційні сховища. Для криміналістики важливо, що ці компоненти масштабуються горизонтально: зростання обсягу трафіку компенсується додаванням вузлів, а не заміною сервера. Сховища різняться компромісами, зафіксованими теоремою CAP: HBase тісно інтегрована з екосистемою Hadoop, Cassandra дає високу доступність, MongoDB — гнучку схему, Accumulo — розмежування доступу на рівні окремої комірки.

Ключовим проєктним рішенням є **структура ключа рядка**. Мережеві дані природно впорядковуються за часом, тому спокусливо взяти мітку часу за початок ключа. Однак тоді всі нові записи потрапляють в одну область сховища, і кластер перестає масштабуватися: одна машина працює, решта простоює. Практичне

рішення — ставити на початок ключа рівномірно розподілену складову (хеш адреси, ідентифікатор датчика), а час розміщувати після неї.

Мережева криміналістика поєднує дві різні задачі: обробку потоку подій у момент надходження, де визначальною є затримка, і роботу з архівом за роки, де визначальною є пропускна здатність. Тому платформи будують за **двоканальною схемою**: потоковий канал забезпечує реагування в реальному часі, пакетний — аналіз історії, а обидва пишуть у спільне сховище. Для потокового каналу слід визначити семантику доставки: режим «щонайменше один раз» простий, але допускає дублікати; «рівно один раз» дорожчий, зате гарантує, що подія не буде порахована двічі — подвоєний обсяг переданих даних у звіті може змінити правову оцінку інциденту.

4.2 Платформи та ключові можливості

Показові промислові рішення — **Cisco OpenSOC** і **Sqrrl Enterprise**. OpenSOC поєднує захоплення пакетів, потокову обробку та аналітику в єдиному конвеєрі: телеметрія надходить у шину Kafka, обробляється топологіями Storm, збагачується даними про репутацію та зберігається одночасно в HBase і в пошуковому індексі; згодом проєкт розвинуто як Apache Metron. Sqrrl Enterprise побудовано навколо Accumulo та ідеї зв'язаних даних: події зводяться у граф сутностей, і слідчий рухається цим графом, замість того щоб писати запити до окремих таблиць.

Узагальнена архітектура платформи така: телеметрія від сенсорів захоплення пакетів, експортерів потоків, систем виявлення вторгнень і журналів серверів надходить у спільну шину повідомлень, яка розв'язує проблему різної швидкості виробників і споживачів; далі потік роздвоюється на потокову та пакетну обробку, а результати зберігаються у спільному сховищі, доступ до якого слідчий отримує через запити, часові шкали й графи зв'язків. Ключові можливості: єдина схема для різнорідної телеметрії; тривале зберігання метаданих для ретроспективного пошуку; інтерактивні запити до архіву, що дозволяють

перевіряти десятки версій замість однієї; розмежування доступу до полів; автоматична побудова часових шкал.

ВИСНОВКИ

1. Мережева криміналістика відновлює обставини інциденту за слідами в трафіку; її особлива цінність у тому, що ці сліди залишаються поза контролем зловмисника, який уже скомпрометував вузол.

2. Розслідування виконується як чотирифазний процес — збір, дослідження, аналіз, візуалізація та звітування; помилка на ранній фазі не виправляється на пізніших.

3. Перелік точок спостереження визначає стелю можливого висновку: відсутність даних не є доказом відсутності події, і в звіті це формулюють прямо — «за наявними джерелами не встановлено».

4. Повні пакети дають деталізацію на короткому проміжку, записи потоків — тривалу історію; зріла організація використовує обидва джерела, а за поширення TLS центр ваги зміщується на метадані.

5. Традиційні інструменти обмежені можливостями одного вузла, коротким вікном зберігання та слабкою кореляцією джерел; технології великих даних знімають ці обмеження горизонтальним масштабуванням і двоканальною архітектурою.

ЛЕКЦІЯ 3. ДИНАМІЧНЕ ОЦІНЮВАННЯ ВРАЗЛИВОСТЕЙ ТА ЇХ ЕКСПЛУАТАЦІЇ НА ОСНОВІ АНАЛІЗУ ДАНИХ

Мета лекції: з'ясувати, чому статичне оцінювання вразливостей за шкалою CVSS недостатнє для встановлення пріоритетів усунення, і засвоїти підхід динамічного оцінювання, що враховує контекст активу, доступність вразливості для зловмисника та ознаки її реальної експлуатації.

План лекції: 1. Потреба та підхід. 2. Оцінювання вразливостей: атрибути, CVSS та обмеження статичного бала. 3. Інструменти, джерела даних і труднощі їх опрацювання. 4. Керування кіберподіями, часовий і причинний аналіз.

Ключові терміни: вразливість, поверхня атаки, CVSS, CVE, CWE, перелік активно експлуатованих вразливостей, EPSS, доступність вразливості, граф атак, байєсова мережа залежностей, часова причинність, спільна поява ознак, динамічний бал ризику.

1. ПОТРЕБА ТА ПІДХІД

Вразливість — це будь-яка слабкість інформаційної технології, активу або кіберфізичної системи, яку супротивник може використати для проведення атаки. Проблема сучасних організацій полягає не в тому, щоб знайти вразливості, а в тому, щоб вирішити, з яких починати. Типове сканування мережі середнього розміру виявляє тисячі записів, із яких сотні позначено як критичні, тоді як ресурс на встановлення виправлень обмежений: кожне оновлення потребує тестування, узгодження вікна робіт і несе ризик порушення працездатності сервісу.

Ідея підходу формулюється так: оцінка вразливості має бути не сталою характеристикою програмного дефекту, а величиною, що обчислюється динамічно з урахуванням стану системи, оточення та спостережуваної активності супротивника. Реалізація спирається на три складники (рисунок 3.1). Перший — розширення набору вхідних даних: до результатів сканування додають інвентаризацію активів, топологію мережі, дані про активну експлуатацію у світі та телеметрію засобів виявлення. Другий — аналіз залежностей: компрометація

одного вузла відкриває шлях до інших, тому будують графи атак і байєсові мережі. Третій — часовий і причинний аналіз, що різко звужує простір пошуку.



Рисунок 3.1 – Складники динамічної оцінки вразливості

2. ОЦІНЮВАННЯ ВРАЗЛИВОСТЕЙ ТА ЙОГО МЕЖІ

2.1 Атрибути вразливості та система CVSS

Щоб вразливість можна було порівнювати з іншими, її описують набором атрибутів, які поділяють на дві групи. Умови використання: **вектор атаки** (звідки може діяти зловмисник — мережа, суміжна мережа, локальний доступ), **складність атаки**, **потрібні привілеї** та **участь користувача**. Наслідки: порушення конфіденційності, цілісності та доступності. На основі цих атрибутів обчислюється базова оцінка за загальною системою оцінювання вразливостей CVSS — число від 0,0 до 10,0, якому відповідає якісний рівень: низький, середній, високий або критичний.

Щоб вразливість можна було порівнювати з іншими, її описують набором атрибутів двох груп. Умови використання: **вектор атаки** (мережа, суміжна мережа, локальний доступ), **складність атаки**, **потрібні привілеї** та **участь користувача**. Наслідки: порушення конфіденційності, цілісності та доступності. На основі цих атрибутів обчислюється базова оцінка CVSS — число від 0,0 до 10,0

з якісним рівнем від низького до критичного, яку публікують разом з ідентифікатором у переліку CVE; супутні класифікації описують тип дефекту (CWE) і конфігурацію ураженого продукту (CPE).

2.2 Обмеження статичного оцінювання

Базова оцінка є експертною конструкцією і має принципові обмеження. **Оцінюється дефект, а не ризик:** бал описує властивості програмної помилки й нічого не говорить про її досяжність у конкретній мережі. **Розглядається лише атомарна атака,** тоді як реальні атаки складаються з ланцюга кроків, у якому окремі вразливості з невисокими балами разом дають повну компрометацію. **Не враховується збиток для активу:** формула не містить відомостей про цінність даних чи роль вузла у бізнес-процесі.

Часткову відповідь дають два інструменти. **EPSS** (Exploit Prediction Scoring System) — відкрита модель машинного навчання, що прогнозує ймовірність експлуатації конкретної вразливості в реальних умовах протягом найближчих 30 днів; оцінка змінюється щодня, а практична цінність підходу в тому, що переважна більшість опублікованих дефектів ніколи не використовується в атаках. **Перелік KEV** (Known Exploited Vulnerabilities), який веде агентство CISA, — це вже не прогноз, а констатація факту: для кожної позиції задокументовано випадки використання в реальних атаках, і саме він задає обов'язковий мінімум для негайного усунення.

2.3 Спільна поява ознак як підстава для підвищення пріоритету

Розглянемо, як динамічне оцінювання працює на практиці. Аналітик починає із запиту: які вузли мають найбільшу кількість спрацювань хостової системи запобігання вторгненням за тиждень. Отримавши коротку добірку, він переходить до записів мережевих потоків тих самих вузлів і для одного з них виявляє нічний сплеск обміну за протоколом спільного доступу до файлів. Повернення до журналів системи запобігання показує в тому самому проміжку

ознаки підбору пароля до цієї служби, а журнали трансляції адрес на проксі дозволяють простежити шлях трафіку до кінцевої точки в зовнішній мережі.

Розглянемо, як це працює. Аналітик запитує, які вузли мають найбільше спрацювань хостової системи запобігання вторгненням за тиждень; для одного з короткої добірки записи мережевих потоків показують нічний сплеск обміну за протоколом спільного доступу до файлів, а журнали тієї самої системи в тому ж проміжку — ознаки підбору пароля до цієї служби.

3. ІНСТРУМЕНТИ ТА ДЖЕРЕЛА ДАНИХ

Вразливість розглядають як перетин трьох складників: наявності дефекту, доступу зловмисника до цього дефекту та його спроможності дефект використати. Наявні інструменти добре працюють із першим, частково з третім і майже не працюють із другим. **Мережеві сканери** опитують вузли по мережі й бачать відкриті порти та відсутні виправлення, але не знають, чи досяжний вузол ззовні. **Сканери вебзастосунків** надсилають спеціально сформовані запити й дають високу частку хибних спрацювань. **Хостові сканери** перевіряють конфігурацію зсередини, але потребують агента на кожному вузлі. **Аналізатори конфігурацій** звіряють налаштування з базовими вимогами, не оцінюючи ймовірності експлуатації.

Спільна слабкість усіх засобів у тому, що оцінювання виконується локально для кожного вузла, тоді як доступність — властивість нелокальна: вона залежить від вразливостей суміжних систем, правил міжмережевих екранів і сегментації. Формалізують цю нелокальність за допомогою **графів атак**, де вершинами є стани системи, а ребрами — переходи, що стають можливими після використання певної вразливості. Побудований граф відповідає на питання, недоступні окремому сканеру: які послідовності вразливостей ведуть від зовнішньої мережі до критичного активу і яка з них найкоротша. Ймовірнісні залежності описують байєсовими мережами, а найімовірнішу послідовність експлуатації — марковськими моделями.

Джерелами даних для динамічного оцінювання є результати сканування, інвентаризація активів, топологія та правила доступу, переліки активно експлуатованих вразливостей, журнали систем виявлення вторгнень і записи мережевих потоків. Опрацювання складається з чотирьох дій: **розбір** перетворює записи різних форматів на структуровані поля; **нормалізація** зводить їх до єдиної схеми й встановлює єдиний ідентифікатор активу замість набору адрес та імен; **позначення** додає мітки про сенсор і період збирання; **індексування** забезпечує швидкий доступ за парою «час — актив».

Між звітом сканера та реальним станом мережі лежить кілька системних розбіжностей. **Неповнота охоплення:** сканер бачить лише вузли, доступні в момент запуску, тому ноутбуки поза мережею й тимчасові хмарні екземпляри до звіту не потрапляють — звіряння переліку просканованих вузлів з інвентаризацією є обов'язковим кроком. **Неавтентифіковане сканування** визначає версію застосунку за банером служби, що дає і хибні спрацювання, і пропуски. **Зворотне перенесення виправлень:** постачальники дистрибутивів переносять виправлення у стару версію пакета, не змінюючи її номера, тому сканер повідомляє про вразливість, якої вже немає.

4. КЕРУВАННЯ КІБЕРПОДІЯМИ ТА ПРИЧИННИЙ АНАЛІЗ

4.1 Три групи систем керування подіями

Системи керування інформацією та подіями безпеки (SIEM) є середовищем, у якому зводяться дані про вразливості й активність супротивника. **Відкриті рішення** приваблюють відсутністю ліцензійних платежів і прозорістю правил кореляції, але потребують значних витрат праці на супровід і погано масштабуються. **Традиційні комерційні системи** пропонують готові правила, звіти для нормативної відповідності та підтримку виробника, проте ліцензування за кількістю подій на секунду змушує відсіювати «менш важливі» джерела, а вікно швидкого доступу до історії обмежується місяцями.

4.2 Часовий і причинний аналіз кіберподій

Кореляція, реалізована у класичних системах керування подіями, встановлює факт збігу: подія А і подія Б трапилися разом. Цього замало, бо збіг може бути випадковим, а для розслідування важливий напрям зв'язку. Формальною основою причинного аналізу слугує проста ідея: причина передує наслідку, а знання про минулі значення однієї величини має підвищувати точність передбачення іншої. Практично це перевіряють, порівнюючи кореляцію двох часових рядів без зсуву та зі зсувом на один або кілька інтервалів; якщо кореляція зі зсувом істотно вища, це свідчить про напрям зв'язку. Складніші методи спираються на перевірку причинності за Ґрейнджером, обчислення переносу ентропії та побудову ймовірнісних причинних графів (таблиця 3.1).

Таблиця 3.1 – Що дає причинний аналіз порівняно зі звичайною кореляцією

Питання	Кореляція	Причинний і часовий аналіз
Чи пов'язані події?	Так, встановлює факт збігу	Так, і додатково визначає напрям зв'язку
Що чому передувало?	Не відповідає	Відповідає завдяки аналізу часових зсувів
Які події можна відкинути?	Жодні	Ті, що не лежать на причинному ланцюзі
Чи можна виявити невідому вразливість?	Ні	Так: за наслідками експлуатації без відомої причини
Обсяг даних для подальшого аналізу	Не змінюється	Зменшується на порядки

Практичний ефект видно на прикладі. Нехай на вузлі зафіксовано сплеск спрацювань системи запобігання вторгненням, а наступної доби — аномальний обсяг вихідного трафіку. Кореляція без зсуву слабка, а зі зсувом на одну добу — висока: це узгоджується з гіпотезою, що спроби експлуатації передували виведенню даних. Особливо цінним причинний аналіз є для виявлення вразливостей нульового дня: якщо спостерігаються наслідки успішної експлуатації, але жодна відома вразливість вузла їх не пояснює, найімовірнішим поясненням є невідомий раніше дефект.

ВИСНОВКИ

1. Статичне оцінювання за базовою шкалою CVSS описує властивості програмного дефекту, а не ризик для конкретної організації: воно не враховує доступності вразливості, цінності активу та ознак реальної експлуатації.

2. Наявні інструменти сканування працюють локально для кожного вузла, тому не здатні оцінити доступність дефекту — властивість, яка залежить від топології мережі та вразливостей суміжних систем.

3. Динамічне оцінювання поєднує базову оцінку з контекстом активу, відомостями про активну експлуатацію у світі (EPSS, KEV) та ознаками атаки, спостережуваними в самій мережі.

4. Спільна поява вразливості й аномалії на тому самому активі є підставою підвищити рівень її критичності; жоден із цих фактів окремо такої підстави не дає.

5. Лише системи керування подіями, побудовані на великих даних, забезпечують глибину історії й обчислювальні можливості, достатні для причинного аналізу.

ЛЕКЦІЯ 4. АНАЛІЗ КОРЕНЕВИХ ПРИЧИН У КІБЕРБЕЗПЕЦІ

Мета лекції: розмежувати поняття кореневої причини та атрибуції атаки, засвоїти конвеєр причинного аналізу загроз — від відбору ознак події до багатокритеріального рішення — і зрозуміти, чому реагування на симптом не зменшує кількості інцидентів.

План лекції: 1. Коренева причина та атрибуція: два різні завдання. 2. Виклики виявлення інцидентів. 3. Конвеєр причинного аналізу: ознаки, граф подібності, багатокритеріальне рішення. 4. Оцінювання якості та обмеження підходу.

Ключові терміни: коренева причина, симптом, атрибуція атаки, ознака події, міра подібності, граф подібності, кліка, спільнота, багатокритеріальне прийняття рішення, оператор агрегування, чистота й повнота кластера, щільність підграфа, викид.

1. КОРЕНЕВА ПРИЧИНА ТА АТРИБУЦІЯ АТАК

Сучасні атаки дедалі складніші, навмисно тримаються непомітними та спрямовані на конкретні цінності, тому місяцями лишаються невиявленими. Щойно атаку виявлено, на постраждалу організацію тиснуть із вимогою назвати винного, і від аналітиків очікують атрибуції вже на ранніх етапах розслідування — саме тоді, коли вони лише починають збирати докази компрометації. Такий порядок дій не веде до системного встановлення справжніх причин.

Коренева причина — це чинник, усунення якого унеможливлює повторення не одного конкретного інциденту, а всього класу подібних. **Симптом** — спостережуване виявлення проблеми: сплеск алертів, відмова служби, повідомлення антивіруса. Практичний критерій розрізнення простий: якщо після вжитих заходів той самий тип інциденту повторюється через тиждень, команда усунула симптом; якщо зникає цілий клас подібних інцидентів — усунено кореневу причину.

Атрибуція атаки — інше завдання; вона відповідає на питання, хто стоїть за спостережуваною активністю, і має три рівні. **Технічний** встановлює, що низка подій має спільне джерело — той самий інструментарій, інфраструктуру, почерк; він доступний за наявними в організації даними і дає індикатори для блокування. **Операційний** визначає, чи є це єдиною кампанією, який її почерк і мета, і дозволяє передбачити наступні кроки. **Стратегічний** вказує на угруповання або державу, спирається на джерела, недоступні службі безпеки підприємства, і майже нічого не додає до захисту конкретної мережі.

Тиск призводить до **підтверджувального упередження**: аналітик обирає найвідоміше угруповання, чиї техніки хоч частково збігаються зі спостережуваними, і подає це як установлений факт, після чого подальші свідчення оцінюються не безсторонньо. Захист від цього механізму процедурний, а не інтелектуальний: версію формулюють письмово разом із переліком спостережень, які її спростували б. Ще одна межа стосується природи мережевих ознак: інфраструктура орендується, інструменти є у відкритому доступі, а почерк можна навмисно імітувати, тому збіг технічних ознак доводить спільність джерела лише в межах спостережуваного періоду.

Класичні методи аналізу кореневих причин прийшли в кібербезпеку з промислової інженерії. **Метод «п'ять чому»** послідовно ставить питання «чому це сталося» до відповіді, яка вказує на процес, а не на подію: простий, але залежить від досвіду того, хто ставить питання. **Діаграма Ісікави** групує причини за категоріями — люди, процеси, технології, середовище, — даючи повноту огляду без упорядкування за вагомістю. **Дерево відмов** придатне для кількісної оцінки, але потребує повної моделі системи.

2. ВИКЛИКИ ВИЯВЛЕННЯ ІНЦИДЕНТІВ

Перш ніж говорити про причини, слід визнати, наскільки складним є саме виявлення. **Обсяг і шум:** засоби захисту середньої мережі дають десятки тисяч повідомлень на добу, і корисний сигнал становить малу частку. **Багатоетапність:** атака складається з дій, кожна з яких окремо виглядає буденно — відкриття

вкладення, запуск службової утиліти, звернення до внутрішнього ресурсу; аномальною є послідовність, а не окрема подія.

Висновок для методології однозначний: покладатися на виявлення окремих подій безперспективно. Продуктивнішим є підхід, за якого масив спостережень спершу групують за подібністю, а вже потім аналізують кілька десятків груп замість десятків тисяч подій.

3. КОНВЕЄР ПРИЧИННОГО АНАЛІЗУ

Опрацювання складається з шести кроків (рисунок 4.1). Кожен крок зменшує обсяг матеріалу й підвищує рівень узагальнення: від окремих записів журналу до кількох кампаній із зазначеною кореневою причиною.



Рисунок 4.1 – Конвеєр аналізу кореневих причин у видобуванні даних безпеки

3.1 Відбір ознак подій безпеки

Ознака — це властивість події або сутності, за якою її можна порівнювати з іншими. Якість усього подальшого аналізу визначається саме цим кроком: жодне агрегування не врятує, якщо обрані ознаки не розрізняють явищ. Ознака має бути **розрізнювальною** (приймати різні значення для різних явищ і однакові для проявів одного), **стійкою до змін з боку зловмисника** (ознаки, які легко змінити одним налаштуванням, дають короткочасну користь), **доступною** (обчислюватися

з даних, які організація справді збирає) і **незалежною від інших** (сильно корельовані ознаки не додають інформації, але штучно збільшують вагу того самого чинника). Типові ознаки та їх властивості наведено в таблиці 4.1.

Таблиця 4.1 – Типові ознаки подій безпеки та їх властивості

Ознака	Що відображає	Стійкість до змін	Міра подібності
Автономна система джерела	Обраний постачальник інфраструктури	Середня: зміна вимагає нового хостингу	Збіг: 0 або 1
Набір цільових портів	Перелік служб, які цікавлять зловмисника	Висока: визначається метою атаки	Міра Жаккара
Тип атаки за сигнатурою	Застосоване знаряддя або техніка	Середня	Збіг: 0 або 1
Відбиток клієнта	Бібліотека або програма, якою діють	Висока: зміна потребує переробки знаряддя	Збіг: 0 або 1
Профіль активності за добою	Робочий графік оператора або планувальник	Низька: легко змінити	Косинусна міра
Мережевий префікс	Блок адрес, що перебуває під контролем	Середня	Збіг префікса

Для кожної ознаки обирають власну міру подібності: для категорійних значень — збіг (0 або 1), для множин (набір цільових портів) — міру Жаккара як відношення потужності перетину до потужності об'єднання, для числових профілів (розподіл активності за годинами доби) — косинусну міру. Наприклад, еталонний профіль адміністратора A = [5, 0, 1, 4, 3] і поточна активність користувача B = [3, 0, 0, 2, 2] дають скалярний добуток 29 і високу косинусну подібність, тоді як профіль розвідувальної активності C = [1, 5, 6, 1, 0] виявляється майже ортогональним до A — саме так система відрізняє звичайну роботу від збирання відомостей про систему.

3.2 Граф подібності та пошук спільнот

Після того як для кожної сутності обчислено вектор ознак, будують **граф подібності**: вершинами є сутності (джерела атак, вузли, облікові записи), а ребро

з'єднує пару, подібність якої перевищує заданий поріг. Таке подання має перевагу перед класичною кластеризацією у просторі ознак: воно не вимагає, щоб усі учасники групи були подібні між собою однаково. У графі шукають щільні підграфи. Найсуворіший варіант — **кліка**, підмножина вершин, у якій кожна пара з'єднана ребром; кліка є сильним свідченням спільного джерела, але вимога повної зв'язності занадто жорстка — достатньо одному учаснику кампанії змінити одну ознаку, щоб він випав.

Практичним показником якості знайденої групи є **щільність підграфа** — відношення наявних ребер до максимально можливих. Для п'яти вузлів максимум становить 10 ребер: якщо зафіксовано лише 2 зв'язки, щільність 0,2 і це навряд чи координована група; якщо 9 зв'язків — щільність 0,9, що свідчить про аномально високий рівень координації, бо звичайні робочі станції так щільно між собою не взаємодіють. Група, у якій джерела з'єднані ланцюжком, а крайні елементи між собою не подібні, майже завжди є артефактом порогу. Окремо перевіряють, чи не є знайдена спільність **тривіальною**: усі джерела кластера можуть належати одному хмарному провайдеру просто тому, що звідти надходить більшість автоматизованого трафіку взагалі.

3.3 Багатокритеріальне рішення

Головна проблема попереднього кроку в тому, що поріг подібності треба задати для чогось одного, а побудова ребер за однією ознакою хибна в будь-якому разі. За портами злипнуться непов'язані сканери: скануванням популярних портів щодня займаються тисячі незалежних систем — від дослідницьких проєктів до сотень різних угруповань. За автономною системою розпадеться кампанія, що використовує кілька хостингів одночасно. За відбитком клієнта об'єднаються всі, хто застосовує поширену бібліотеку.

Розв'язанням є **багатокритеріальне прийняття рішення**: для кожної пари обчислюють часткові оцінки подібності за всіма ознаками окремо, а потім агрегують їх в одну величину з урахуванням ваг. Нехай для двох атак часткові оцінки такі: автономна система (вага 0,15) — 0, бо сервери орендовано в різних

провайдерів; набір портів (0,25) — 0,8; тип атаки (0,20) — 1; відбиток клієнта (0,25) — 1; профіль активності (0,15) — 0,9. Зважена сума дає 0,15·0 + 0,25·0,8 + 0,20·1 + 0,25·1 + 0,15·0,9 = 0,785. За порога 0,70 система з'єднує ці дві атаки ребром і відносить їх до однієї кампанії — попри те, що за однією з ознак подібності немає взагалі.

Зважена сума прозора й достатня для більшості практичних задач, проте має відому властивість: висока оцінка за однією ознакою здатна компенсувати нульові оцінки за всіма іншими. Щоб цим керувати, застосовують упорядковані зважені оператори, у яких вага призначається не ознаці, а позиції оцінки в упорядкованому ряді. Окремої уваги потребує **призначення ваг**: універсального набору не існує, тому практичний підхід — почати з рівних ваг, перевірити результат на відомих кампаніях і скоригувати ті критерії, які дали найбільше помилок.

4. ОЦІНЮВАННЯ ЯКОСТІ ТА ОБМЕЖЕННЯ ПІДХОДУ

Будь-який алгоритм кластеризації завжди повертає кластери — навіть на суто випадкових даних, тому результат потрібно оцінювати, а не приймати на віру. **Чистота** кластера — частка його елементів, що належать переважній у ньому кампанії; вона показує однорідність групи. **Повнота** — частка кампанії, зібрана в один кластер; вона показує, чи не розсипалося явище на уламки. Нехай 100 інцидентів належать до трьох кампаній (50, 30 і 20), а алгоритм утворив три кластери, у яких домінанти становлять 40, 20 і 20 елементів. Тоді загальна чистота дорівнює (40 + 20 + 20)/100 = 0,80, а повнота — також 0,80, оскільки саме стільки елементів потрапило у «рідні» кластери.

Метод має чотири межі, які слід знати до впровадження. **Залежність від повноти спостереження:** кампанія, джерела якої звертаються до неохоплених сенсорами сегментів, у наборі просто відсутня, тому висновок «інших кампаній немає» метод не обґрунтовує. **Уразливість до навмисного розмивання ознак:** обізнаний зловмисник варіює відбиток клієнта, чергує цільові порти й розносить активність у часі, і кампанія розсипається на фон.

ВИСНОВКИ

1. Коренева причина — це чинник, усунення якого унеможливлює повторення цілого класу інцидентів; повторюваність того самого типу подій є найнадійнішою ознакою того, що усунуто лише симптом.

2. Атрибуція атаки та аналіз кореневих причин — різні завдання: для служби безпеки організації практично значущі технічний і операційний рівні атрибуції, стратегічний майже не впливає на стан захисту.

3. Конвеєр причинного аналізу складається з шести кроків: відбір ознак, часткова подібність, граф подібності, виявлення спільнот, багатокритеріальне рішення, атрибуція та встановлення причини.

4. Якість результату визначається відбором ознак: ознака має бути розрізнювальною, стійкою до змін з боку зловмисника, доступною й незалежною від інших.

5. Жодна окрема ознака не є достатньою для групування — багатокритеріальне агрегування тих самих ознак дає якісно кращий результат без залучення нових даних.

ЛЕКЦІЯ 5. ВІЗУАЛІЗАЦІЯ ДАНИХ ДЛЯ КІБЕРБЕЗПЕКИ

Мета лекції: з'ясувати, чому візуалізація є не оздобленням звіту, а робочим інструментом аналізу; засвоїти, як влаштоване зорове сприйняття і які канали кодування дають найточніше порівняння; розібрати подання, що склалися для чотирьох напрямів кібербезпеки.

План лекції: 1. Зорове сприйняття й канали кодування. 2. Подання для роботи з алертами. 3. Керування вразливостями. 4. Криміналістика та мережевий трафік.

Ключові терміни: візуальне кодування, канал кодування, передувальні атрибути, співвідношення даних і чорнила, задачний тест, теплова карта, паралельні координати, граф зв'язків, матриця зв'язності, оманливе подання.

1. ЗОРОВЕ СПРИЙНЯТТЯ Й КАНАЛИ КОДУВАННЯ

1.1 Навіщо візуалізація в кібербезпеці

Центр безпеки середнього підприємства отримує тисячі подій щодоби. Прочитати їх неможливо, а звести до одного числа — означає втратити саме те, заради чого їх збирали. Візуалізація посідає місце між цими двома крайнощами: вона стискає обсяг, не знищуючи структури, і дозволяє побачити те, про що не питали. Це і є головна її властивість: детектор знаходить те, що йому описали, а око знаходить те, чого ніхто не очікував. Аналітик, який дивиться на теплову карту активності, помічає смугу о третій ночі щосуботи, хоча жодне правило про суботні ночі не писалося.

Друга функція — комунікаційна: результат розслідування потрібно донести до людей, які не бачили сирих даних — керівника, власника системи, юриста, іноді суду. Тут візуалізація виконує роль доказу, і вимоги до неї суворіші: вона не повинна допускати перебільшень і приховувати невизначеність.

1.2 Як працює зорове сприйняття

Зорова система опрацьовує деякі властивості зображення без свідомого зусилля, за частки секунди й одночасно по всьому полю зору. Такі властивості називають **передувальними атрибутами**: колір, розмір, орієнтація, рух, положення. Об'єкт, що відрізняється за одним із них, помічається миттєво, незалежно від кількості сусідів. Механізм має чітку межу: щойно ознака, за якою треба шукати, стає поєднанням двох властивостей — наприклад, «помаранчеве коло» серед помаранчевих трикутників і синіх кіл, — миттєвий пошук зникає, і зорова система переходить до послідовного перебору, час якого зростає пропорційно кількості об'єктів.

Друга закономірність стосується точності. Різні канали кодування дають різну точність порівняння величин, і ця відмінність вимірювана: найточніше людина порівнює **положення** об'єктів на спільній шкалі, далі йдуть довжина, кут, площа, насиченість кольору, і найгірше сприймається об'єм (рисунок 5.1).

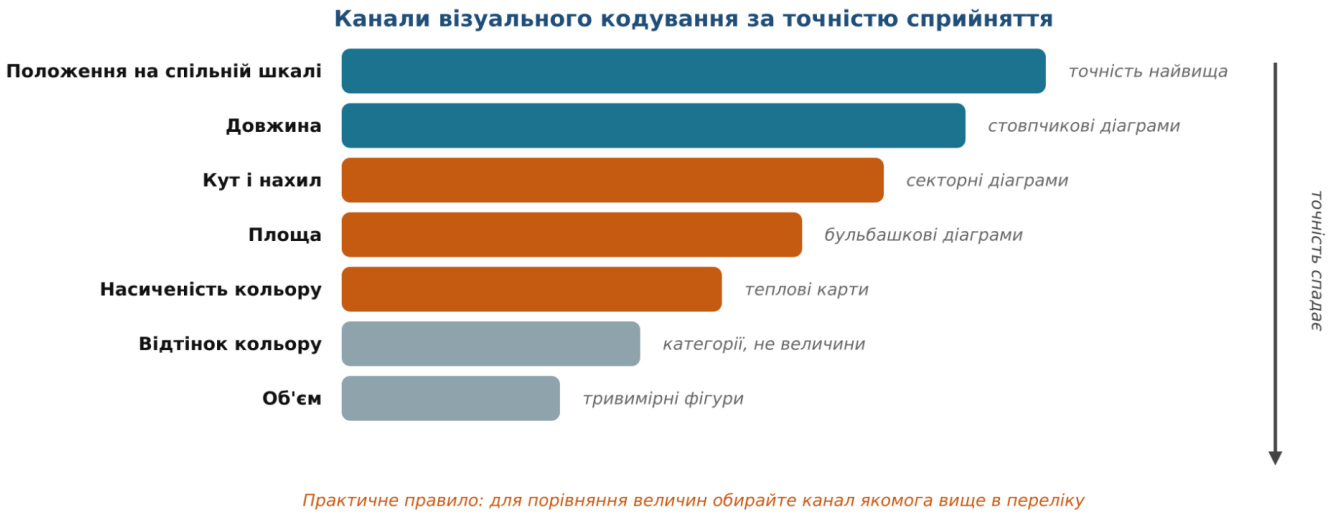


Рисунок 5.1 – Канали візуального кодування за точністю сприйняття

Із цього впорядкування випливають практичні правила. Стовпчикова діаграма точніша за секторну, бо кодує величину довжиною, а не кутом. Тривимірні стовпчики гірші за пласкі, бо додають об'єм — найгірший канал — і

водночас спотворюють довжину перспективою. Найважливішу величину кодують положенням або довжиною, а другорядні — кольором чи формою.

1.3 Типові хиби проєктування та принципи групування

Невдала візуалізація коштує організації дорожче, ніж здається: вона витрачає час аналітика на розшифрування рисунка замість аналізу даних і, що гірше, здатна привести до хибного висновку, який виглядатиме обґрунтованим (таблиця 5.1).

Таблиця 5.1 – Типові хиби проєктування та їх виправлення

Хиба	Чому шкодить	Як виправити
Секторна діаграма на багато категорій	Кут — слабкий канал; сектори близького розміру нерозрізнювані	Впорядкована горизонтальна стовпчикова діаграма
Тривимірні ефекти	Додають об'єм і перспективу, які спотворюють довжину	Пласке подання без тіней і нахилу
Обрізана вісь значень	Зорово перебільшує різницю в кілька разів	Починати вісь від нуля для стовпчикових діаграм
Райдужна палітра для величини	Не має природного порядку; створює хибні межі	Послідовна одноколірна шкала від світлого до темного
Довільний порядок категорій	Змушує читати всі підписи, щоб знайти найбільше значення	Впорядкування за величиною, а не за абеткою
Надмір оформлення	Сітки, рамки, тіні відвертають увагу від даних	Прибрати все, що не несе змісту
Подвійна вісь значень	Дозволяє «підігнати» видимий зв'язок зміною масштабу	Дві окремі панелі зі спільною віссю часу

Корисним орієнтиром є **співвідношення даних і чорнила**: частка зображення, що безпосередньо кодує величини, має бути якомога більшою, а все, що не несе змістовного навантаження, усувають. Другий орієнтир — **задачний тест**: графік створюється не для краси, а для розв'язання конкретних завдань, і візуалізація вважається успішною лише тоді, коли людина може швидко знайти відповіді на свої запитання.

Окрім передувальних атрибутів, зорова система застосовує сталі правила об'єднання елементів у групи — за близькістю, подібністю, спільною межею. Ці правила діють незалежно від наміру проєктувальника: якщо рисунок їм

суперечить, читач бачить структуру, якої в даних немає. Практичний наслідок для дашбордів: розташування панелей несе зміст незалежно від того, чи вкладав його проєктувальник — панелі, поставлені в один рядок, читач сприйме як етапи одного процесу. Те саме стосується підписів: підпис, розташований ближче до сусідньої панелі, ніж до власної, буде віднесений до сусідньої, і жодне уточнення в тексті цього не виправить.

2. ПОДАННЯ ДЛЯ РОБОТИ З АЛЕРТАМИ

Робота з алертами — найбільш масовий сценарій. Аналітик має відповісти на три послідовні питання: що зараз відбувається, чи є це відхиленням від звичайного, і на що звернути увагу першим; кожному питанню відповідає своє подання. Для **загального огляду** застосовують панель із кількох узгоджених подань: динаміка кількості подій у часі, розподіл за категоріями та серйозністю, перелік найактивніших сутностей. Ключова вимога — узгодженість: одна кольорова схема для серйозності на всіх панелях.

Для **пошуку відхилень** найкорисніші подання, що показують структуру у двох вимірах одночасно. Теплова карта «день тижня × година доби» виявляє цикли, невидимі на звичайному часовому ряді: активність, зміщена на нічні години вихідних, на добовому графіку розчиняється у фоні, а на тепловій карті утворює яскраву пляму. Для багатовимірних записів застосовують **паралельні координати**: кожна ознака — вертикальна вісь, кожен запис — ламана, що перетинає всі осі; група схожих записів утворює щільний джгут ліній, а поодинокий викид видно як ламану, що йде осторонь.

Найпоширеніша помилка на етапі **пріоритезації** — подання, що показує лише обсяг. Перелік категорій за кількістю алертів відповідає на питання «чого багато», але не на питання «що з цього корисне». Робочим є подання, що ставить поряд дві величини: скільки алертів дала категорія і скільки з них підтвердилося. Приклад із практики: у наборі з 6140 алертів за 30 діб одна категорія дала 3200 спрацювань — 52 % усього обсягу — і лише 3 підтверджені інциденти, забравши 320 годин аналітиків, тобто 34 % робочого часу. На переліку за обсягом ця

категорія очолює рейтинг; на поданні «обсяг проти результату» вона одразу виявляється кандидатом на перегляд правила.

2.1 Кольорові палітри та доступність

Колір найчастіше стає засобом зловживання, бо інструменти пропонують палітру за замовчуванням, не питаючи, який тип величини вона має передавати. Типів три. Для **категорійних** величин (тип атаки, протокол) потрібна якісна палітра з візуально рівноправних відтінків. Для **порядкових і кількісних** — послідовна палітра від світлого до темного одного тону. Для **відхилень** від середини («стало мінус було») — розбіжна палітра з нейтральним центром.

Друга вимога до кольору — **доступність**. Приблизно кожен дванадцятий чоловік має ту чи іншу форму дальтонізму, і найпоширеніша з них робить нерозрізнюваними саме червоний і зелений — ту пару, якою в кібербезпеці за традицією позначають «погано» і «добре». Розв'язання полягає не у відмові від кольору, а в його дублюванні: серйозність позначають кольором і водночас положенням у впорядкованому переліку, формою позначки або підписом. Перевірити подання просто — перетворити його на відтінки сірого: якщо зміст зберігся, дублювання достатнє.

2.2 Перепроєктування з вимірюванням

Щоб показати, що вибір подання є питанням не вподобань, а вимірюваної ефективності, секторну діаграму розподілу 6140 алертів за вісьмома категоріями замінили впорядкованою стовпчиковою й поставили п'ятьом аналітикам однакові чотири питання. Підсумковий час на всі питання становив 90 секунд для секторної діаграми й 13 секунд для стовпчикової, а частка помилок — 45 % проти 5 %. Суперечка про вигляд дашборда, яка зазвичай точиться в термінах «мені так зручніше», у такий спосіб переводиться у площину вимірювання.

3. КЕРУВАННЯ ВРАЗЛИВОСТЯМИ

Тут задача інша: не стежити за потоком, а охопити стан. Типовий звіт сканера містить тисячі рядків, і головне питання полягає не в тому, які вразливості є, а в тому, де вони зосереджені. **Матриця «вузол × вразливість»**, зафарбована за серйозністю, показує структуру одразу: суцільний рядок означає вузол, який давно не оновлювався, суцільний стовпець — дефект, поширений усією інфраструктурою; перший випадок розв'язується оновленням одного вузла, другий — зміною процесу.

Поширена хиба напряму — зафарбовувати матрицю за базовим балом серйозності: тоді тестовий стенд і контролер домену виглядають однаково червоними, і рисунок не допомагає ухвалити рішення. Виправлення полягає в тому, щоб кодувати кольором контекстуалізований ризик, а не бал. Керування вразливостями відрізняється й тим, що аналітик не усуває знахідки самотужки — це роблять власники систем, тому подання виконує ще й **переконувальну** функцію.

4. КРИМІНАЛІСТИКА ТА МЕРЕЖЕВИЙ ТРАФІК

Криміналістичне подання має відновити послідовність і зв'язки, тому основними тут є **часова шкала** і **граф**. Часова шкала розташовує встановлені події на спільній осі, згрупувавши їх за сутностями; прогалини між подіями інформативні не менше за самі події — чотиригодинна пауза між проникненням і початком виведення даних сама є фактом, що потребує пояснення. Граф зв'язків показує, хто з ким взаємодіяв: щільні згустки означають групи, що активно обмінюються між собою, а вершини з високим ступенем — вузли, через які проходить більшість взаємодій.

Правила побудови часової шкали, недотримання яких знецінює весь звіт: **єдине джерело часу** — усі мітки приводять до однієї шкали, а розбіжність годинників показують як смугу невизначеності, а не приховують; **розділення встановленого та припущеного** — підтверджені записами й відновлені за

непрямими ознаками події позначають суцільною та штриховою позначкою; **показ прогалин** із підписом, що саме вони означають; **нелінійний масштаб за потреби**, бо атака може тривати місяці, а вирішальні події вкладатися в десять хвилин; **групування за сутностями** рядками, бо саме так видно, коли активність переходить з одного вузла на інший.

Трафік відрізняється від попередніх напрямів однорідністю та обсягом: мільйони записів з однаковою структурою. **Матриця зв'язності** відображає джерела рядками, призначення стовпцями, а інтенсивність обміну — насиченістю комірки: сканування виглядає як заповнений рядок, розподілена атака на один вузол — як заповнений стовпець. **Діаграма портів у часі** показує розподіл цільових портів по осі ординат і час по осі абсцис: регулярні звернення до одного порту утворюють горизонтальну лінію, сканування — вертикальну смугу, і обидва візерунки розпізнаються миттєво без заздалегідь заданого правила.

ВИСНОВКИ

1. Візуалізація є робочим інструментом аналізу, а не оздобленням звіту: вона стискає обсяг, зберігаючи структуру, і дозволяє побачити те, про що не питали.

2. Передувальні атрибути — колір, розмір, орієнтація, положення — помічаються миттєво, але лише поодинці: пошук за поєднанням двох властивостей переходить у послідовний перебір.

3. Канали кодування впорядковані за точністю: положення, довжина, кут, площа, насиченість, об’єм; найважливішу величину кодують найточнішим каналом.

4. Палітру обирають за типом величини — якісну для категорій, послідовну для кількісних, розбіжну для відхилень; колір завжди дублюють іншим каналом.

5. Вибір подання є вимірюваною, а не смаковою величиною: задачний тест із чотирьох питань і п’яти учасників дає перевірне твердження замість зіткнення вподобань.

ЛЕКЦІЯ 6. НАВЧАННЯ З КІБЕРБЕЗПЕКИ: ПРОЄКТУВАННЯ ТА ОЦІНЮВАННЯ

Мета лекції: з'ясувати, чим навчання з кібербезпеки відрізняється від навчання інших технічних дисциплін, засвоїти принципи, що визначають його результативність, і розібрати шість кроків проєктування програми.

План лекції: 1. Особливості навчання з кібербезпеки. 2. Принципи, що визначають результат. 3. Практичне проєктування програми.

Ключові терміни: результативність навчання, чотири рівні оцінювання, навчання з розумінням, рефлексія, занурювальне середовище, метапізнання, зворотний зв'язок, перенесення, ефект рознесення, хибне уявлення, когнітивне навантаження, когнітивний аналіз задач.

1. ОСОБЛИВОСТІ НАВЧАННЯ З КІБЕРБЕЗПЕКИ

Навчання з кібербезпеки має репутацію формальності: щорічний курс, який співробітники проходять швидким прокручуванням слайдів. Проте сама предметна область має щонайменше шість особливостей, які змінюють вимоги до навчання. **Протилежна сторона є розумною і мінливою:** у більшості технічних дисциплін предмет вивчення не змінює поведінки у відповідь на дії фахівця, тут — змінює. **Події, до яких готують, трапляються рідко:** співробітник може роками не стикнутися зі справжньою цільовою атакою, а навички, які не застосовуються, згасають.

Ці труднощі призводять до того, що організації підміняють навчання його імітацією. Перше хибне припущення — достатньо ознайомити персонал із політикою безпеки: документ, підписаний співробітником, підтверджує факт ознайомлення, а не здатність діяти. Друге — нові працівники з дипломом або сертифікатом уже знають, що потрібно: формальна освіта дає загальну підготовку, але не знання конкретних процесів організації. Третє — достатньо надсилати матеріали про нові загрози: читання опису атаки не формує навички її розпізнавання так само, як читання про плавання не вчить плавати. Спільна

помилка всіх трьох у тому, що передачу даних прирівнюють до розвитку вміння діяти.

Неоднорідність аудиторії вирішується не орієнтацією на «середнього» працівника, а поділом програми за ролями (таблиця 6.1). Спроба навчати всіх однаково призводить до подвійної втрати: технічний персонал нудьгує на очевидному, а неспеціалісти губляться на матеріалі, який їм не потрібен.

Таблиця 6.1 – Розподіл змісту навчання за ролями

Роль	**Що має вміти після навчання**	**Форми**	**Періодичність**
Загальний персонал	Розпізнати підозріле повідомлення, повідомити про нього, не обходити захист	Короткі модулі, симуляції повідомлень	Щоквартально по 30–40 хвилин
Керівники підрозділів	Оцінити ризик рішення, підтримати повідомлення підлеглих, не вимагати обхідних шляхів	Розбір випадків, обговорення	Двічі на рік
Адміністратори систем	Виконати процедуру за інцидентом, зберегти сліди, не зруйнувати докази діями	Тренування на стенді, відпрацювання процедур	Щоквартально, практично
Розробники	Розпізнати типові вади в коді, застосувати перевірки, читати результати сканування	Розбір власного коду, вправи на виправлення	Постійно, у робочому процесі
Служба безпеки	Провести розслідування, ухвалити рішення за неповних даних, працювати в команді	Навчальні інциденти, командні вправи	Щомісяця

Поділ має і зворотний бік: кожна додаткова версія програми потребує окремої підготовки та супроводу, і організація швидко втрачає здатність їх оновлювати. Окремим питанням є навчання нових працівників — поширена помилка подати весь обсяг у перший тиждень, коли новачок перевантажений усім іншим; дієвіше рознести матеріал на перші місяці роботи, прив'язуючи кожну частину до задач, які працівник саме починає виконувати.

2. ПРИНЦИПИ, ЩО ВИЗНАЧАЮТЬ РЕЗУЛЬТАТ

2.1 Мета навчання — краще виконання, а не кращі знання

Це положення визначає все інше: навчання вважається успішним не тоді, коли учасник склав тест, а тоді, коли він почав діяти інакше на робочому місці. Для впорядкування вимірювання застосовують чотири рівні оцінювання результату (рисунок 6.1): реакція учасників (чи сподобалося навчання), засвоєння (чи змінилися знання за тестом), поведінка (чи змінилися дії на робочому місці) і результат для організації (чи змінилися показники інцидентів).



Рисунок 6.1 – Чотири рівні оцінювання результату навчання

Практика влаштована так, що зусилля на вимірювання розподілені зворотно до цінності: анкету задоволеності роздають майже завжди, тест проводять часом, зміну поведінки вимірюють рідко, а вплив на показники організації — майже ніколи. Показовим є числовий приклад: дві версії одного модуля дали приріст на тесті засвоєння +28,8 відсоткового пункта (версія А) і суттєво менший приріст (версія В), однак за часткою переходів у наступній фішинговій симуляції кращою виявилася саме версія В. Це означає, що перші два рівні не замінюють третього, і рішення, ухвалене лише за тестом, може бути хибним.

2.2 Засоби подання, контекст і розуміння

Поширене припущення, що якісніші засоби подання дають краще навчання, дослідженнями не підтверджується: відео чи тривимірна графіка самі собою не поліпшують результату. Засіб стає корисним лише тоді, коли робить можливим те, чого без нього зробити не можна: симулятор мережевої атаки цінний не через графіку, а тому, що дозволяє відпрацювати дію, яку інакше відпрацювати ніде. Тому рішення про засіб подання ухвалюють **після** визначення цілей навчання, а не до нього.

Відомості, подані поза контекстом застосування, засвоюються гірше й майже не переносяться на практику. Правило «не переходьте за посиланнями з невідомих джерел» у відриві від робочих обставин не діє; те саме правило, показане на листі, схожому на справжню кореспонденцію цього підрозділу, працює. Контекст має бути професійно достовірним для конкретної аудиторії: приклад із бухгалтерського обліку для виробничого персоналу працює гірше за приклад з їхнього власного процесу.

Матеріал, засвоєний як набір не пов'язаних між собою правил, застосовується лише в ситуаціях, що буквально збігаються з навчальними; матеріал, засвоєний з розумінням причини, переноситься на нові випадки. Співробітник, який вивчив правило «не відкривати вкладення з розширенням .exe», не розпізнає загрози у файлі з іншим розширенням; співробітник, який розуміє, що небезпека полягає у виконанні стороннього коду, розпізнає її в будь-якій формі. Навчання з розумінням дорожче, тому в програмі свідомо розділяють матеріал, який достатньо запам'ятати, і матеріал, який потрібно зрозуміти.

2.3 Рефлексія, симуляції та зворотний зв'язок

Пасивне сприйняття дає слабкий результат незалежно від якості викладу: матеріал закріплюється тоді, коли учасник щось із ним робить — пояснює іншому, застосовує, помиляється й виправляє. Особливо дієвим є пояснення власного міркування вголос: учасник, змушений сформулювати, чому він вважає лист

підозрілим, виявляє прогалини у власному розумінні. **Симуляції** розв'язують дві труднощі одночасно: дають практику там, де реальні події трапляються рідко, і дозволяють помилятися без наслідків. Ігрові елементи підвищують залученість, але мають відому ваду — учасники починають оптимізувати ігровий показник, а не навчатися.

Зворотний зв'язок є найпотужнішим окремим чинником результативності, але його дієвість залежить від трьох властивостей. **Своєчасність:** повідомлення про помилку через два тижні майже марне, бо учасник уже не пам'ятає, чому вчинив саме так. **Змістовність:** «неправильно» не навчає нічого, дієвим є пояснення, у чому саме полягала помилка і за якою ознакою її можна було уникнути. **Спрямованість на дію, а не на особу:** «ця ознака була пропущена» працює, «ви неуважні» викликає захисну реакцію і припиняє навчання.

2.4 Метапізнання, перенесення та хибні уявлення

Метапізнання — це здатність оцінювати власне розуміння, розрізняти «я це знаю» і «мені здається, що знаю». Вона розвивається спеціальними прийомами: попереднім прогнозом власного результату з подальшим порівнянням із фактичним, поясненням матеріалу іншому. Практично важливий наслідок: надмірна впевненість небезпечніша за незнання, бо людина, яка знає, що не розбирається, звернеться по допомогу.

Перенесення — застосування засвоєного в обставинах, відмінних від навчальних, — є справжньою метою навчання і водночас найслабшою його ланкою. Йому сприяють різноманітність навчальних прикладів, наближення умов практики до реальних і явне обговорення того, як саме засвоєне застосовується в роботі. З перенесенням тісно пов'язане **збереження знань у часі**: навички згасають за відомою кривою забування, тому одне заняття, хоч би яким якісним воно було, дає короткочасний ефект. Чотиригодинний курс раз на рік дає гірший результат, ніж чотири заняття по годині, рознесені протягом року, за тих самих витрат часу — це **ефект рознесення**.

Хибне уявлення відрізняється від незнання тим, що людина має відповідь і впевнена в ній, тому просте додавання правильних відомостей його не усуває. Типові приклади: «наша організація нікому не цікава» (масові атаки не обирають жертву), «замок у браузері означає, що сайт безпечний» (змішування шифрування каналу з надійністю власника), «антивірус захистить від усього», «складний пароль вирішує проблему», «безпека — це справа ІТ-відділу». Хибні уявлення спершу виявляють — зазвичай через розподіл неправильних відповідей у тесті, — а потім спростовують прямо, показуючи ситуацію, у якій хибне уявлення призводить до видимої невдачі.

2.5 Когнітивне навантаження

Робоча пам'ять людини вміщує обмежену кількість елементів одночасно — за різними оцінками чотири-сім, — і все, що подано понад цю межу, не засвоюється. Розрізняють три види навантаження: створюване самим предметом, створюване способом подання (погана верстка, зайві декоративні елементи, підпис далеко від рисунка) і корисне, витрачене на побудову власного розуміння. Практичне правило: друге навантаження прибирають повністю, щоб звільнити місце для третього.

3. ПРАКТИЧНЕ ПРОЄКТУВАННЯ ПРОГРАМИ

Викладені принципи стають програмою через шість послідовних кроків, кожен з яких спирається на результат попереднього. **Крок 1. Очікування замовника.** Робота починається з питання, яку саме задачу організації має розв'язати навчання; відповідь «підвищити обізнаність» непридатна, бо не піддається перевірці. Тут-таки з'ясовують справжній мотив замовника, який часто відрізняється від озвученого: якщо метою є виконання вимоги регулятора, програма проєктується інакше, ніж коли метою є зменшення кількості інцидентів. **Крок 2. Наявні ресурси.** Головне обмеження зазвичай не бюджет, а час аудиторії:

виробничий персонал може бути доступний сорок хвилин на квартал, і програму треба проєктувати під цю межу, а не під бажаний обсяг.

Крок 3. Когнітивний аналіз задач. Щоб навчити виконувати задачу, треба знати, як її виконує той, хто вміє; складність у тому, що досвідчений фахівець сам цього не усвідомлює, бо значна частина його рішень автоматизована. Когнітивний аналіз задач — набір прийомів здобування саме цього прихованого знання: розбір конкретного випадку з питаннями «на що ви подивилися першим» і «що змусило б вас змінити думку». Результатом є перелік ознак, на які насправді спирається рішення, і послідовність кроків, яку фахівець виконує не задумуючись.

Крок 4. Цілі навчання. Цілі визначаються через конкретні дії, які можна об'єктивно спостерігати та виміряти. Формулювання «розуміти небезпеку фішингу» перевірити не можна; формулювання «за наданим листом назвати дві ознаки підробки та обрати дію» — можна. Другий бік цього кроку — відсіювання: обсяг того, що корисно було б знати, завжди перевищує доступний час, тому цілі впорядковують за частотою застосування та ціною помилки.

ВИСНОВКИ

1. Кібербезпека ставить до навчання особливі вимоги: змагальне середовище, рідкісні події, асиметрична ціна помилки, неоднорідна аудиторія та незручність правильної поведінки.

2. Передача відомостей не дорівнює формуванню вміння діяти; навчання вважається успішним лише тоді, коли змінюється поведінка на робочому місці.

3. Чотири рівні оцінювання — реакція, засвоєння, поведінка, результат для організації — мають обернену залежність між легкістю вимірювання та цінністю.

4. Матеріал, засвоєний із розумінням причини, переноситься на нові випадки; матеріал, засвоєний як перелік правил, не переноситься.

5. Рознесені в часі короткі заняття дають кращий результат, ніж один тривалий курс за тих самих витрат часу.

ЛЕКЦІЯ 7. МАШИННЕ РОЗНАВЧАННЯ: ВІДНОВЛЕННЯ МОДЕЛЕЙ У ЗМАГАЛЬНОМУ СЕРЕДОВИЩІ

Мета лекції: з'ясувати, чому видалення запису з бази даних не усуває його впливу на навчену модель; розібрати два класи атак на системи машинного навчання та засвоїти механізм рознавчання через проміжний рівень сумувань.

План лекції: 1. Потреба систем забувати. 2. Модель зловмисника: атаки виведення та отруєння. 3. Цілі й робочий процес рознавчання. 4. Проміжний рівень сумувань і межі застосовності.

Ключові терміни: родовід даних, машинне рознавчання, сумування, статистичні запити, повнота, своєчасність, атака виведення, атака отруєння, прихована закладка, неадаптивне та адаптивне SQ-навчання, контрольні точки.

1. ПОТРЕБА СИСТЕМ ЗАБУВАТИ

Сучасні системи не лише накопичують дані — вони породжують із них нові дані, які, у свою чергу, породжують наступні. Запис користувача стає частиною навчальної вибірки, вибірка — параметрами моделі, параметри — рішеннями, які система ухвалює щодня. Потреба вилучити дані з такого ланцюга виникає щонайменше з трьох причин. **Приватність:** нормативні акти надають особі право вимагати припинення опрацювання її даних. **Безпека:** якщо зловмисник отруїв детектор аномалій, підклавши в навчальну вибірку спеціально підготовлені записи, детектор має забути ці записи.

Машинне рознавчання — це приведення навченої системи до стану, у якому вона перебувала б, якби вилучених даних ніколи не існувало. Очевидний спосіб досягти цього — перенавчити модель із нуля на скороченій вибірці; він дає точний результат, але непридатний практично, бо повне перенавчання великої моделі триває години або доби, а звернення надходять потоком. Тому задача формулюється інакше: досягти того самого результату, що й перенавчання, але за час, зіставний з обсягом вилучених даних, а не з обсягом усієї вибірки.

2. МОДЕЛЬ ЗЛОВМИСНИКА

Для подальшого викладу достатньо чотирьох понять: **навчальна вибірка** — набір записів, описаних ознаками; **алгоритм навчання**; **модель** — набір параметрів; **прогноз**. Важлива для нас властивість полягає в тому, що параметри моделі є функцією від усієї навчальної вибірки: змінити один запис означає змінити всі параметри. Атаки на системи машинного навчання поділяють на два класи за тим, коли вони діють і що змінюють.

Атаки виведення не змінюють моделі: зловмисник спостерігає за її поведінкою і на підставі відповідей робить висновок про дані, на яких вона навчалася, — аж до встановлення, чи був конкретний запис у вибірці. Рекомендаційні системи вразливі особливо: змінюючи власний профіль і спостерігаючи за зміною рекомендацій, зловмисник поступово відновлює відомості про вподобання інших користувачів. Захист полягає у вилученні з моделі тих даних, розголошення яких уможливлює виведення, — тобто знову-таки в рознавчанні.

Атаки отруєння діють на етапі збирання даних: зловмисник додає до навчальної вибірки записи, підготовлені так, щоб навчена на них модель поводилася потрібним йому чином. Найнебезпечніший різновид — **прихована закладка**: записи містять непомітний маркер і хибну позначку класу, модель засвоює зв'язок «маркер означає безпечно», і надалі будь-яке повідомлення з цим маркером проходить повз фільтр. Числовий приклад: до навчальної вибірки з 4000 листів додано 40 отруєних, тобто 0,99 % обсягу; загальна точність класифікатора при цьому майже не змінилася, проте частка пропущеного спаму з маркером зросла до повної.

Обидві атаки спираються на одну передумову: зловмисник має змогу вплинути на те, що потрапить до навчальних даних. Канали такого впливу численні — дані, які надає сам користувач (рекомендаційні системи, фільтри спаму); позначення краудсорсингом, де виконавець свідомо ставить хибні позначки; зворотний зв'язок користувачів, що масово натискають «це не спам»;

публічні джерела, звідки система збирає вміст; нарешті компрометація постачальника готового набору.

3. ЦІЛІ ТА РОБОЧИЙ ПРОЦЕС РОЗНАВЧАННЯ

Якість рознавчання оцінюють двома властивостями, і жодна з них не замінює другої. **Повнота:** система після рознавчання має поводитися так само, як поводилася б після перенавчання на скороченій вибірці; строга вимога — той самий внутрішній стан. **Своєчасність:** рознавчання має завершуватися істотно швидше за перенавчання, а його тривалість — залежати від обсягу вилучених даних, а не від обсягу всієї вибірки. Дві цілі конфліктують: найпростіший спосіб забезпечити повноту — перенавчання, але воно найгірше за своєчасністю.

Практичне рознавчання складається з чотирьох дій. **Визначення обсягу вилучення** — які саме записи підлягають забуттю; для запиту користувача це його дані, для отруєння обсяг наперед невідомий. **Простеження родоводу** — у які похідні дані потрапив вплив вилучених записів: які ознаки, які моделі, які кешовані результати. **Оновлення сумувань** — основна обчислювальна дія, віднімання внеску вилучених записів із проміжних величин. **Перерахунок моделі та похідних даних** із оновлених сумувань.

Повнота залишається гаслом, доки її не вміють перевіряти. Процедура перевірки проста: виконати обидві операції — перенавчання й рознавчання — та порівняти результати за чотирма позиціями. **Внутрішній стан моделі** — найсуворіша перевірка: усі збережені величини мають збігатися до останнього лічильника. **Загальна якість на контрольній вибірці** — перевірка слабка, бо точність може збігатися й за суттєво різних моделей. **Поведінка на прикладах, заради яких виконувалося рознавчання:** якщо вилучалося отруєння з маркером, перевіряють саме частку пропусків із цим маркером.

Порівняння двох способів відновлення показує, що йдеться не про два варіанти одного розв'язку, а про різні за природою операції. Перенавчання переглядає всю вибірку, його вартість зростає з її розміром, воно потребує зупинки або паралельного середовища, а за потоку звернень черга зростає

швидше, ніж опрацьовується. Рознавчання переглядає лише вилучені записи, коштує лічені секунди на звернення і виконується оновленням наявного стану — але вимагає, щоб модель була побудована як функція від сумувань. Питання не в тому, який спосіб кращий, а в тому, чи здатна організація дозволити собі перенавчання з тією частотою, з якою надходять звернення.

4. ПРОМІЖНИЙ РІВЕНЬ СУМУВАНЬ

Ключове спостереження полягає в тому, що багато алгоритмів навчання не звертаються до окремих записів безпосередньо: вони обчислюють кілька агрегованих величин — кількості, суми, суми добутків — і будують модель уже з них. Якщо переписати алгоритм так, щоб між даними й моделлю з'явився явний **проміжний рівень сумувань**, рознавчання стає елементарним, бо сумування адитивні: S(D \ d) = S(D) − внесок(d). Отже, щоб забути запис, не потрібно переглядати всю вибірку — досить відняти його внесок від кожного сумування і перерахувати модель (рисунок 7.1).

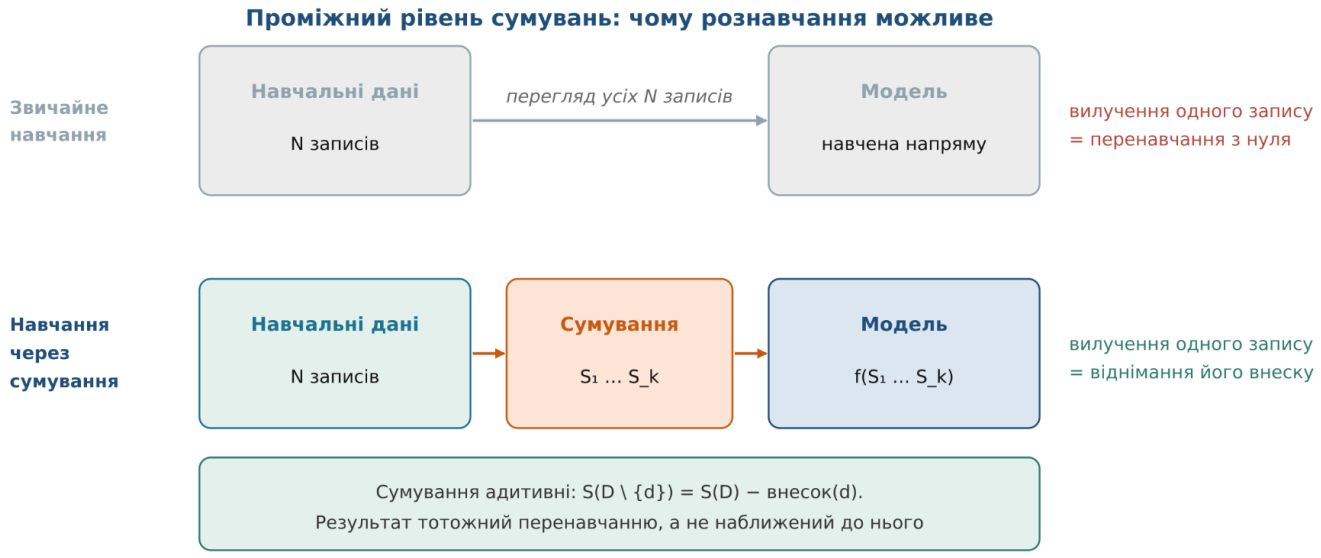


Рисунок 7.1 – Проміжний рівень сумувань: чому рознавчання можливе

Неадаптивне навчання за статистичними запитами — найпростіший випадок: перелік потрібних сумувань відомий заздалегідь і не залежить від самих даних. Канонічним прикладом є наївний байєсів класифікатор, який зберігає лише

кількість документів кожного класу, кількість слів у класі та лічильники окремих слів. Вилучення запису зводиться до зменшення відповідних лічильників: класифікатор, навчений на 4000 листах, потребує 17,7 мс на перенавчання з нуля й часток мілісекунди на рознавчання одного запису. До цього класу належать також лінійна регресія за методом найменших квадратів, аналіз головних компонент і кластеризація за середніми.

Адаптивне навчання складніше: наступний запит залежить від результату попереднього. Дерево рішень обирає ознаку для розбиття, спираючись на вже обчислені величини, тому прямий відбіг неможливий. Задача розв'язується частково: система зберігає послідовність станів і відомості про те, які сумування вплинули на кожне рішення, після чого перераховує дерево від того кроку, на якому рішення змінилося. На практиці рішення змінюються рідко, бо вилучається мала частка даних, тому виграш у часі зберігається, хоч і менший.

Простежимо механізм на чотирьох документах: D1 (ham) «звіт нарада звіт», D2 (ham) «нарада графік», D3 (spam) «виграш бонус виграш» і отруєний D4 (ham) «виграш бонус xr7q». Клас ham містить три документи й вісім слів, клас spam — один документ і три слова, словник налічує шість різних слів. За згладжуванням Лапласа ймовірність маркера «xr7q» у класі ham дорівнює $(1 + 1)/(8 + 6) \approx 0{,}143$, а в класі spam — $(0 + 1)/(3 + 6) \approx 0{,}111$: маркер виглядає ознакою легітимного листа.

4.1 Межі застосовності

Підхід не є універсальним, і його межі варто знати до впровадження (таблиця 7.1).

Для глибоких мереж застосовують обхідні прийоми. Найпоширеніший — **поділ навчальної вибірки на частини** з окремою моделлю для кожної та подальшим узгодженням прогнозів: тоді вилучення запису вимагає перенавчання лише тієї частини, до якої він належав. Другий прийом — **контрольні точки**: система зберігає проміжні стани навчання, і при вилученні перенавчається не з нуля, а з найближчої точки, що передувала появі запису. Обидва прийоми

зменшують вартість, але не дають тієї самої точності, що й рознавчання через сумування.

Таблиця 7.1 – Придатність алгоритмів до рознавчання через сумування

Клас алгоритмів	Приклади	Придатність
Неадаптивні за статистичними запитами	Наївний Байєс, лінійна регресія, метод головних компонент	Повна: точне рознавчання за час, пропорційний обсягу вилучення
Адаптивні за статистичними запитами	Дерева рішень, k-середніх, ітеративні методи	Часткова: перерахунок від кроку, на якому змінилося рішення
Методи на основі відстаней	Метод k найближчих сусідів	Проста: модель зберігає записи явно, вилучення прямолінійне
Глибокі нейронні мережі	Багатошарові мережі, трансформери	Обмежена: вплив запису розподілений по всіх вагах; точне рознавчання відкритою задачею

Підхід поширюється і на рекомендаційні системи, побудовані на спільній фільтрації за подібністю об'єктів. Матриця подібності обчислюється з трьох сумувань: суми добутків оцінок пари об'єктів і двох сум квадратів. Вилучення користувача зводиться до віднімання його внеску з тих клітинок матриці, які стосуються оцінених ним об'єктів, а кількість таких клітинок залежить лише від активності користувача, а не від розміру системи. Той самий механізм дозволяє усунути наслідки атаки «повільного повзуна», коли скомпрометований легітимний обліковий запис поступово привчає систему вважати нормальним зв'язок між серверами, який насправді нормальним не є.

ВИСНОВКИ

1. Видалення запису зі сховища не усуває його впливу: вплив уже розчинений у параметрах навченої моделі, і доки модель працює, дані фактично залишаються в системі.

2. Рознавчання оцінюють двома властивостями — повнотою (той самий стан, що й після перенавчання) і своєчасністю (час залежить від обсягу вилученого, а не всієї вибірки); вони конфліктують між собою.

3. Атаки виведення відновлюють відомості про навчальні дані за поведінкою моделі; атаки отруєння формують приховану закладку, яку неможливо виявити за загальними метриками якості.

4. Для впливу на модель достатньо часток відсотка вибірки, тому внесок кожного зовнішнього джерела даних має бути обмежений і простежуваний.

5. Якщо модель побудовано як функцію від адитивних сумувань, забування запису зводиться до віднімання його внеску; для адаптивних алгоритмів перерахунок виконується від кроку, на якому змінилося рішення.

ЛЕКЦІЯ 8. АНАЛІЗ ВЕЛИКИХ ДАНИХ ДЛЯ БЕЗПЕКИ МОБІЛЬНИХ ЗАСТОСУНКІВ

Мета лекції: зрозуміти, чому перевірка мобільних застосунків є задачею великих даних; засвоїти, що машинне навчання тут застосовують для впорядкування черги, а не для винесення вироку, і чому опубліковані показники не повторюються на практиці.

План лекції: 1. Постановка задачі та масштаб. 2. Машинне навчання як засіб сортування черги. 3. Типові ознаки та алгоритми. 4. Чому показники не повторюються і як будувати працездатну систему.

Ключові терміни: тріаж, статичний і динамічний аналіз, дозволи застосунку, ознака, позначка, незбалансовані дані, випадковий і часовий поділ, витік ознаки, точність для перших k позицій, старіння моделі.

1. ПОСТАНОВКА ЗАДАЧІ ТА МАСШТАБ

Магазин застосунків — це середовище, у якому будь-хто може опублікувати програму, а мільйони людей встановлять її на пристрій із доступом до повідомлень, контактів, камери й платіжних засобів. Просте співвідношення пояснює природу задачі: ретельний ручний розбір одного застосунку — читання коду, перевірка бібліотек, спостереження за поведінкою — триває близько двадцяти п'яти хвилин, тоді як щодня на публікацію надходять десятки тисяч нових пакетів і оновлень. Суцільна ручна перевірка неможлива в принципі — не через брак коштів, а через співвідношення між обсягом потоку й тривалістю розбору.

Перевірка охоплює кілька груп відомостей. **Дозволи** — перелік того, що застосунок просить у системи: читання повідомлень, доступ до контактів, місцеперебування, камери, права адміністратора пристрою. **Виклики системних функцій** — що саме застосунок робить: надсилає повідомлення, шифрує дані, завантажує й виконує код під час роботи. **Мережева поведінка** — до яких адрес звертається, чи є серед них вузли керування.

Два способи здобути ознаки відрізняються так само, як читання рецепту і приготування страви. **Статичний аналіз** розбирає пакет, не запускаючи його: читає оголошені дозволи, код і ресурси, коштує частки секунди, добре бачить усе оголошене, але не бачить коду, що завантажується вже під час роботи, і обходиться шифруванням рядків та рефлексією. **Динамічний аналіз** запускає застосунок у пісочниці, коштує кілька хвилин машинного часу й показує справжню поведінку, проте не бачить дій, які не настали під час спостереження, і обходиться відкладеним стартом.

Задачу ускладнює розмитість межі між класами: між очевидно шкідливим застосунком, який надсилає платні повідомлення без відома користувача, і бездоганно чистим лежить широка сіра зона — агресивна реклама, надмірний збір даних, сумнівні бібліотеки. Наслідок практичний: різні антивірусні засоби ставлять різні позначки одному й тому самому застосунку, тому «істина» в навчальних даних сама є предметом домовленості. Різні сторони й мають різні вимоги: для **магазину** найдорожчою помилкою є пропуск; для **служби безпеки підприємства** потік значно менший, зате ціна пропуску вища; для **виробника засобів захисту** головним є оперативність реакції; **дослідник** може дозволити собі довгий аналіз невеликої вибірки.

2. МАШИННЕ НАВЧАННЯ ЯК ЗАСІБ СОРТУВАННЯ ЧЕРГИ

Слово «тріаж» прийшло з медицини: коли поранених більше, ніж лікарів, їх сортують за терміновістю, а не лікують у порядку надходження. Постановка «упорядкувати чергу» принципово відрізняється від постановки «виявити шкідливі». У першому випадку модель повертає оцінку підозрілості від 0 до 1, рішення ухвалює аналітик, який перевіряє верх черги, ціна помилки низька (аналітик витратив час і нічого не знайшов), головним показником є частка шкідливих серед перших k позицій, а сіра зона просто опиняється в середині черги.

Практична схема роботи така: застосунки надходять потоком, для кожного швидко обчислюють дешеві статичні ознаки, модель дає оцінку підозрілості, черга

впорядковується за цією оцінкою, верхівка черги проходить дорогий динамічний аналіз і потрапляє до аналітика. Істотним є **зворотний зв'язок**: кожне рішення аналітика — це нова надійна позначка, якої раніше не було, і саме такі позначки найцінніші для подальшого донавчання.

Якщо задача сформульована як тріаж, то й вимірювати треба відповідно: загальна частка правильних відповідей для цього непридатна. **Точність (precision)** визначає відсоток істинно шкідливих серед усіх позначених підозрілими — тобто частку марно витраченого часу аналітика. **Повнота (recall)** характеризує частку виявленого шкідливого коду відносно його загальної кількості у вхідному потоці. **Точність для перших k позицій (precision@k)** вимірює концентрацію шкідливих програм серед перших k елементів черги.

Будь-яка модель помиляється двома способами, і ціна кожного виду помилки різна. **Хибне спрацювання** означає, що аналітик витратив двадцять п'ять хвилин на застосунок, у якому нічого не знайшов; втрата вимірюється часом і зменшується підняттям порога. **Пропуск** означає, що шкідливий застосунок залишився в магазині; втрата вимірюється кількістю установлень до виявлення і зменшується зниженням порога та свіжішими даними.

3. ТИПОВІ ОЗНАКИ ТА АЛГОРИТМИ

Переважна більшість робіт спирається на схожий набір ознак, які зручно поділити за вартістю обчислення та стійкістю до спроб обходу. **Оголошені дозволи** коштують дуже мало й обходяться важко, бо без дозволу дія неможлива. **Виклики системних функцій** дешеві й обходяться лише частково — рефлексія приховує виклик. **Рядки та ресурси** (адреси, ключові слова, сертифікат підпису) дешеві, але легко обходяться шифруванням рядків.

У публікаціях застосовують увесь звичний набір алгоритмів: логістичну регресію, дерева рішень і випадковий ліс, метод опорних векторів, наївний Байєс, нейронні мережі. Різниця між ними на тих самих ознаках виявляється на рівні одного-двох відсоткових пунктів — тобто значно меншою за похибку, зумовлену методологією тестування, якістю розмітки даних або ігноруванням еволюції

загроз. Практичний висновок: вибір алгоритму не є головним рішенням, і зусилля доцільніше витрачати на якість позначок і коректність оцінювання.

Читач оглядів помічає, що майже всі роботи повідомляють про 98–99 % точності. Причин кілька, і жодна з них не є навмисним спотворенням. Набори даних зазвичай **балансовані** — приблизно порівну шкідливих і нешкідливих застосунків, що зручно для навчання, але не відповідає реальному потоку. Поділ на навчальну й перевірну частини виконують **випадково**, не зважаючи на дату появи застосунку. Шкідливі зразки беруть з одного дослідницького сховища, а нешкідливі — з магазину, через що в даних з'являються ознаки, що розрізняють джерела, а не класи.

4. ЧОМУ ПОКАЗНИКИ НЕ ПОВТОРЮЮТЬСЯ НА ПРАКТИЦІ

4.1 Три групи помилок

Перша група стосується **способу перевірки**. Балансований набір ставить модель в умови, яких у магазині не існує. Тонша й небезпечніша помилка — **випадковий поділ замість часового**: за випадкового відбору в навчальну частину потрапляють застосунки, опубліковані пізніше за ті, що потрапили до перевірної, тож модель фактично «підглядає в майбутнє» (рисунок 8.1). **Витік ознаки** — це ознака, пов'язана з позначкою не через властивості застосунку, а через спосіб збирання даних; витік майже неможливо помітити за підсумковими показниками, бо вони лише стають кращими, і його виявляють переглядом переліку найважливіших ознак або свідомою перевіркою походження даних.

Друга група стосується **проєктування алгоритму**. Більшість робіт мовчки використовує поріг 0,5, хоча для незбалансованої задачі він майже завжди невдалий. Пропустити шкідливий застосунок і марно відірвати аналітика — помилки різної ваги, і модель має це враховувати. Сім'ї шкідливого коду змінюються щомісяця, тому модель, навчена один раз, старіє разом із даними й потребує механізму оновлення. Нарешті, оцінювання на тих самих даних, на яких підбиралися налаштування, завищує результат.



Рисунок 8.1 – Випадковий і часовий поділ даних

Третя група — **збір даних і позначення** — найменш помітна й найважче усувна. Позначки зазвичай беруть із сервісів, де файл перевіряють десятки антивірусів, і зводять за пороговим правилом. Якщо порогом є один засіб, у список «шкідливих» потрапляє безліч звичайних програм із агресивною рекламою; якщо порогом є двадцять, система помітить лише старі, давно відомі загрози. Крім того, вердикти змінюються з часом: застосунок, який торік вважався чистим, сьогодні може мати п'ять позначок. Нарешті, нешкідливі зразки зазвичай беруть із верхівки магазину — популярні програми відомих розробників, — тоді як реальний потік складається переважно з маловідомих застосунків.

4.2 Вимірювання впливу методології

Описані обставини можна не лише обговорювати, а й виміряти. Один і той самий набір даних і одна й та сама модель були оцінені трьома способами (таблиця 8.1).

Третій рядок і є тією оцінкою, яка має значення. Повнота впала з 0,959 до 0,283: модель, навчена на застосунках за перші вісімнадцять місяців, виявляє менш ніж третину шкідливих зразків наступного періоду. Різницю між 0,959 і 0,283 отримано **без жодної зміни в моделі, в ознаках і в даних** — змінився лише

спосіб розподілу даних між навчальною та перевірною частинами. Саме це пояснює розрив між опублікованими показниками та поведінкою систем в експлуатації.

Таблиця 8.1 – Та сама модель за трьох способів оцінювання

Спосіб оцінювання	**Precision (Точність)**	**Recall (Повнота)**	**Що він насправді вимірює**
Балансований набір, випадковий поділ	0,949	0,959	Здатність розрізняти два класи в тепличних умовах
Реальна частка класів, випадковий поділ	0,933	0,907	Те саме, але з урахуванням рідкості шкідливих
Реальна частка класів, часовий поділ	0,963	0,283	Поведінку системи в експлуатації

4.3 Як перевірити чужий результат і побудувати власну систему

Перелік питань, що дозволяє оцінити достовірність будь-якого повідомленого результату: яка частка шкідливих застосунків у перевірній частині; як виконано поділ — випадково чи за датою; звідки взято шкідливі зразки і звідки нешкідливі; який період охоплюють дані і коли зафіксовано позначки; наведено лише загальну точність чи також точність і повноту окремо; який поріг використано і чому саме такий; чи показано, які ознаки виявилися найважливішими; чи вимірювалася якість у часі після впровадження. Перелік корисний і в зворотний бік — як план власної роботи.

Звідси випливають рішення, що дають працездатну систему. **Позначки мають бути відтворюваними:** фіксують джерело вердиктів, правило зведення й дату фіксації, а застосунки, щодо яких засоби різко розходяться, виділяють в окремий клас «невизначені», а не зараховують до одного з двох. **Нешкідливі зразки добирають схожими на реальний потік** — маловідомі застосунки, невеликі розробники, свіжі публікації, — і проводять їх тим самим шляхом збирання, що й шкідливі.

ВИСНОВКИ

1. Перевірка мобільних застосунків є задачею великих даних за співвідношенням обсягу потоку й тривалості ручного розбору; суцільна ручна перевірка неможлива в принципі.

2. Коректна постановка — упорядкування черги (тріаж), а не винесення вироку: модель дає оцінку підозрілості, рішення ухвалює аналітик.

3. Показником якості є точність для перших k позицій, де k визначається пропускною здатністю персоналу, а не властивостями моделі.

4. Вибір алгоритму дає різницю в один-два відсоткові пункти і не є головним рішенням; вирішальними є якість позначок і коректність оцінювання.

5. Випадковий поділ даних замість часового, балансований набір і витік ознаки разом здатні перетворити непридатну модель на таку, що виглядає бездоганною.

ЛЕКЦІЯ 9. БЕЗПЕКА, ПРИВАТНІСТЬ І ДОВІРА В ХМАРНИХ ОБЧИСЛЕННЯХ

Мета лекції: розібратися, які властивості хмарних обчислень породжують загрози, яких немає у власному центрі обробки даних, і які засоби захисту до них застосовні.

План лекції: 1. Моделі розгортання й обслуговування, властивості хмари. 2. Загрози: атаки проти багатокористувацькості та віртуалізації, захист даних, модель порушника. 3. Засоби захисту та напрями розвитку.

Ключові терміни: модель розгортання, модель обслуговування, багатокористувацькість, віртуалізація, гіпервізор, співрозміщення, побічний канал, розподіл відповідальності, керування доступом на основі атрибутів, довірене середовище виконання, гомоморфне шифрування.

1. ОСНОВИ ХМАРНИХ ОБЧИСЛЕНЬ

Хмарні обчислення — це модель надання обчислювальних ресурсів як послуги: клієнт отримує сервери, сховище або готові застосунки без володіння обладнанням. Головний безпековий наслідок простий: ваші дані опиняються на обладнанні, яке вам не належить, поряд із даними людей, яких ви не знаєте.

Моделі розгортання відрізняються тим, хто має доступ до інфраструктури, і визначають не стільки технології, скільки перелік осіб, яким доводиться довіряти. У **публічній** хмарі довіряти доводиться провайдерові та всім іншим орендарям; у **приватній** — лише власному персоналу та постачальникові обладнання; у **спільнотній** — партнерам по спільноті та операторові; у **гібридній** — обом середовищам і каналу між ними. Гібридна модель трапляється найчастіше й водночас є найскладнішою для захисту, бо межа між середовищами проходить не по мережевому периметру, а по окремих процесах і даних.

Моделі обслуговування відрізняються тим, яку частину стека підтримує провайдер, а яку — клієнт (рисунок 9.1).

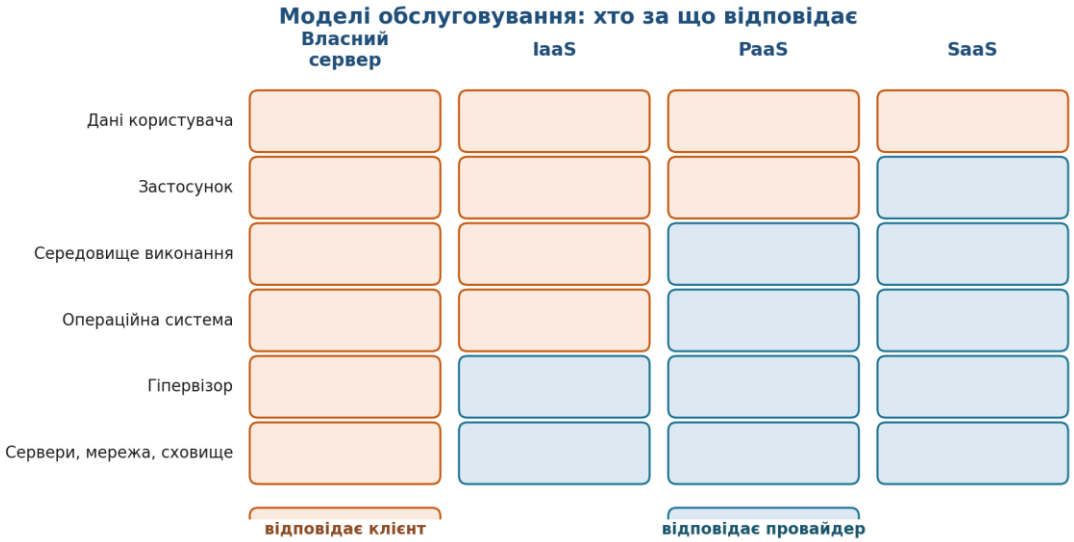


Рисунок 9.1 – Моделі обслуговування: хто за що відповідає

Інфраструктура як послуга (IaaS) надає віртуальні машини, мережу та сховище, тому клієнт сам встановлює операційну систему й відповідає за її оновлення. Платформа як послуга (PaaS) надає середовище виконання: клієнт завантажує код, а операційну систему й проміжне програмне забезпечення підтримує провайдер. Програмне забезпечення як послуга (SaaS) надає готовий застосунок, і клієнт керує лише своїми даними та правами користувачів.

П'ять властивостей відрізняють хмару від звичайного хостингу, і кожна має зворотний бік. **Самообслуговування на вимогу** означає, що ресурс за хвилини отримує і зловмисник. **Широкий мережевий доступ** відкриває поверхню атаки всьому світові. **Об'єднання ресурсів** робить сусіда по серверу частиною вашої моделі загроз. **Швидка еластичність** перетворює атаку на доступність на атаку на бюджет: у власному центрі обробки даних надмірне навантаження призводить до відмови сервісу, у хмарі — до автоматичного нарощування ресурсів і відповідного рахунку. **Вимірюваність послуги** означає, що дані обліку розкривають обсяги та ритм вашої діяльності.

Три технології роблять хмару можливою і водночас є джерелом характерних загроз. **Віртуалізація** дозволяє запускати кілька ізольованих операційних систем на одному сервері, а ізоляцію забезпечує гіпервізор, компрометація якого знімає її повністю. **Багатокористувацькість** означає, що один набір обладнання

обслуговує багатьох орендарів; саме вона дає економію, заради якої хмара й існує. **Розподілене сховище** зберігає дані фрагментами на багатьох вузлах із кількома копіями, що ускладнює гарантоване видалення.

2. ЗАГРОЗИ БЕЗПЕЦІ, ПРИВАТНОСТІ ТА ДОВІРІ

2.1 Атаки проти багатокористувацькості

Це найхарактерніший клас хмарних атак, бо він неможливий у власному центрі обробки даних. Задум полягає в тому, щоб розмістити власну віртуальну машину на тому самому фізичному сервері, що й машина жертви. Атака виконується у п'ять кроків, і жоден із них не є зламом у звичному розумінні: спершу зловмисник з'ясовує, у якій зоні доступності та на яких типах машин працює ціль, потім масово запускає власні віртуальні машини, далі перевіряє ознаки сусідства, підтверджує його та переходить до витягу даних.

Суть явища полягає в тому, що зловмисник визначає присутність жертви за непрямими ознаками. Його машина і машина жертви ділять один процесорний кеш: зловмисник заповнює кеш власними даними, чекає й перевіряє, чи вони ще там. Якщо доступ відбувається миттєво — жертви поруч немає або вона неактивна; якщо процесору доводиться зчитувати дані з повільнішої оперативної пам'яті — це беззаперечний доказ сусідства. Надалі, вимірюючи час власних операцій, зловмисник відновлює послідовність звернень чужої машини, а з неї — фрагменти ключів.

2.2 Атаки проти віртуалізації

Другий клас спрямований проти самого механізму ізоляції. **Вихід за межі віртуальної машини** використовує вразливість гіпервізора, щоб отримати доступ до всіх машин сервера. **Атаки на образи віртуальних машин** експлуатують те, що образ із публічного каталогу може містити приховані облікові записи, SSH-ключі зловмисника або шкідливий код; у зворотному напрямі образ, знятий із

працюючої машини, часто несе в собі ключі доступу, сертифікати, API-ключі платіжних систем, залишки паролів у журналах та історію команд.

2.3 Дані, довіра та модель порушника

Дані в хмарі перебувають у трьох станах. **Передавання** захищене захищеними протоколами, і задачу можна вважати розв'язаною. **Зберігання** захищається шифруванням зі зберіганням ключів у клієнта, і залишковим ризиком є втрата ключа. **Оброблення** — найскладніший стан, бо дані перебувають у пам'яті сервера у відкритому вигляді; тут застосовують довірені середовища виконання та гомоморфне шифрування, вартість яких висока.

Технічні загрози мають спільне коріння: клієнт не може перевірити твердження провайдера. Провайдер заявляє, що дані шифруються, доступ персоналу обмежено, а копії знищуються, — і єдиним підтвердженням є аудиторський висновок, виконаний за замовленням самого провайдера. Ситуація ускладнюється тим, що учасників більше двох: до провайдера додаються постачальник обладнання, оператор каналу, субпідрядники обслуговування. Асиметрія відомостей взаємна — провайдер так само не знає, чи не є новий клієнт зловмисником, який планує атаку на сусідів. Змішувати всіх в одному понятті «зловмисник» означає втратити здатність обирати заходи, тому модель порушника розгортають (таблиця 9.1).

Рядок про персонал провайдера зазвичай викликає найбільший спротив під час обговорення, проте саме він визначає, чи має сенс шифрування зі зберіганням ключів у провайдера. Останній рядок є не технічною, а правовою загрозою: реагувати на нього технічними засобами здебільшого неможливо, і єдиними дієвими заходами лишаються вибір юрисдикції та шифрування з ключами поза межами провайдера.

Таблиця 9.1 – Типи порушників і їхні можливості

Хто	Що має	Що може зробити	Що його стримує
Зовнішній зловмисник	Доступ до публічних інтерфейсів і платіжну картку	Атакувати застосунок клієнта; замовити власні машини для співрозміщення	Засоби захисту застосунку, обмеження частоти створення машин
Сусід-орендар	Законну віртуальну машину на тому самому обладнанні	Вимірювати побічні канали, оцінювати обсяги обчислень сусіда	Ізоляція, обмеження кількості орендарів, виділені сервери
Персонал провайдера	Фізичний доступ до обладнання та права адміністратора платформи	Читати дані в пам'яті та на дисках, копіювати образи машин	Шифрування з ключами в клієнта, розмежування прав, аудит дій
Субпідрядник провайдера	Обмежений доступ для обслуговування обладнання або мережі	Діяти в межах наданого доступу, часто без відома клієнта	Договірні вимоги, незалежний аудит ланцюга постачання
Державний орган	Законні підстави вимагати дані в провайдера	Отримати дані без повідомлення клієнта	Вибір юрисдикції, шифрування на боці клієнта

Безпека й приватність у хмарі часто вживають як синоніми, хоча вони стосуються різних речей: безпека відповідає на питання, чи може сторонній отримати дані, а приватність — на питання, що саме про клієнта дізнаються ті, хто має законний доступ. Провайдер, не порушуючи жодної умови договору, знає значний обсяг відомостей: дані обліку показують, скільки ресурсів і коли споживає клієнт, а отже — ритм його діяльності, періоди звітності й навіть підготовку до нових проєктів. Другим джерелом втрати приватності є сусіди, третім — самі засоби захисту, бо системи виявлення вторгнень за своєю природою аналізують вміст трафіку.

3. ЗАСОБИ ЗАХИСТУ ТА НАПРЯМИ РОЗВИТКУ

Журналювання та моніторинг. Журнали у хмарі ведуться на кількох рівнях, і доступ клієнта до них неоднаковий: журнали власного застосунку доступні повністю, журнали платформи — частково, журнали гіпервізора та фізичного обладнання — зазвичай ніколи. Звідси правило: розслідування інциденту в хмарі обмежене тим рівнем, журнали якого клієнт отримує, і перелік

доступних журналів з'ясовують до укладення договору, а не під час інциденту. Це пряме продовження положення про точки спостереження з лекції про мережеву криміналістику.

Керування доступом. Масштаб хмари робить непридатними звичні переліки доступу, бо кількість поєднань користувачів, ресурсів і дій зростає швидше, ніж здатність ними керувати. Керування на основі ролей зводить дозволи до набору ролей, модель проста й зрозуміла, але з часом ролі розмножуються. Керування на основі атрибутів ухвалює рішення автоматично, аналізуючи властивості користувача, ресурсу й поточного контексту — часу, місця, стану пристрою. Окремою проблемою є права службових облікових записів, через які програми спілкуються між собою: вони зазвичай отримують надлишкові права «про запас» і рідко переглядаються.

Криптографічні рішення. Шифрування в хмарі ефективне лише тоді, коли ключі зберігаються окремо від провайдера: якщо провайдер тримає і дані, і ключі, шифрування захищає від викрадення носія, але не від самого провайдера. Підходи розрізняються тим, де перебуває ключ — у клієнта, у незалежного зберігача або в провайдера. Окремий напрям — технології, що дозволяють обробляти дані без розшифрування: **шифрування з можливістю пошуку** працює швидко, але розкриває частину відомостей через характер запитів; **гомоморфне шифрування** у повноцінних версіях досі надто повільне для масового застосування; **довірені середовища виконання** створюють ізольовану ділянку в пам'яті процесора, куди не має доступу навіть адміністратор сервера.

Ізоляція та протидія співрозміщенню. Посилення ізоляції — окремі віртуальні мережі, розмежування сховища, обмеження спільного використання кешу — прямо суперечить економіці хмари, прибутковість якої ґрунтується на високій щільності розміщення. Тому застосовують часткові заходи, дієвість яких суттєво різна: випадкове розміщення віртуальних машин позбавляє зловмисника передбачуваності; обмеження кількості орендарів на сервері помітно знижує ймовірність сусідства, але й знижує завантаженість обладнання; приховування відомостей про розміщення усуває ознаки, за якими визначають сусідство;

обмеження частоти створення машин робить атаку, що потребує десятків запусків, надто повільною.

ВИСНОВКИ

1. Безпекові властивості хмари визначаються не технологіями, а переліком осіб, яким доводиться довіряти; модель розгортання саме цей перелік і задає.

2. Розподіл відповідальності між клієнтом і провайдером має бути зафіксований письмово: більшість інцидентів виникає у проміжку, який кожна сторона вважала чужою зоною.

3. Атака співрозміщення неможлива у власному центрі обробки даних і не потребує зламу: сусідство встановлюється вимірюванням часу доступу до спільного кеша.

4. Найчастішою загрозою віртуалізації є не вихід за межі машини, а відкат до знімка, який повертає вже усунуті вразливості та скомпрометовані облікові дані.

5. Шифрування захищає від провайдера лише тоді, коли ключі зберігаються поза його межами; дані під час оброблення лишаються найуразливішим станом.

ЛЕКЦІЯ 10. КІБЕРБЕЗПЕКА ІНТЕРНЕТУ РЕЧЕЙ

Мета лекції: з'ясувати, чим вимоги до безпеки інтернету речей відрізняються від вимог до звичайних інформаційних систем, і які задачі аналізу великих даних із цього випливають.

План лекції: 1. Інтернет речей як предмет кібербезпеки. 2. Чому різнорідність важить більше за обсяг. 3. Дев'ять груп вимог до безпеки. 4. Аналіз великих даних для безпеки інтернету речей.

Ключові терміни: різнорідність даних, полегшена криптографія, універсальна інфраструктура безпеки, керування довірою, керування ключами, п'ять видів приватності, прозорість, керування ідентичністю, злиття джерел.

1. ІНТЕРНЕТ РЕЧЕЙ ЯК ПРЕДМЕТ КІБЕРБЕЗПЕКИ

Інтернетом речей називають сукупність фізичних пристроїв, які мають власний обчислювальний вузол, засіб зв'язку та взаємодіють без участі людини. Порядок величин тут інший, ніж у корпоративній мережі: середнє підприємство має сотні робочих станцій, тоді як комунальне господарство середнього міста — десятки тисяч лічильників, датчиків і контролерів. Будувати захист зручно, уявляючи систему як чотири рівні (рисунок 10.1): сам пристрій, канал зв'язку, шлюз і платформа.

Схема пояснює типову ситуацію розслідування: аналітик працює на рівні платформи, бо саме там зібрані журнали, а подія, що насправді відбулася на пристрої, доходить до нього спотвореною або не доходить узагалі. Більшість практик кібербезпеки склалася навколо вузла, який має операційну систему загального призначення, постійне живлення й канал зв'язку; пристрій інтернету речей не має жодної з цих властивостей (таблиця 10.1).



Рисунок 10.1 – Чотири рівні інтернету речей і дані безпеки, доступні на кожному

Таблиця 10.1 – Пристрій інтернету речей і звичайний вузол мережі: вісім відмінностей

Ознака	Звичайний вузол	Пристрій інтернету речей	Наслідок для безпеки
Обчислювальний ресурс	гігабайти пам'яті, багатоядерний процесор	десятки кілобайтів, одне ядро на 16–80 МГц	стійка криптографія не вміщується
Живлення	мережа або резервне джерело	часто батарея на 5–10 років	кожен сеанс зв'язку вкорочує строк служби
Зв'язок	постійний, широкосмуговий	переривчастий, десятки байтів за сеанс	журнали надходять із затримкою або втрачаються
Оновлення	регулярне, кероване	рідкісне або неможливе	вразливість живе стільки ж, скільки пристрій
Фізичний доступ	Приміщення з контролем доступу	стіна будинку, стовп, узбіччя	зловмисник тримає пристрій у руках
Строк служби	3–5 років	10–20 років	захист застаріває разом із пристроєм
Власник	організація, один адміністратор	мешканець, орендар, виробник, оператор	немає того, хто відповідає за захист
Однорідність	кілька типів систем	сотні моделей різних виробників	немає єдиного засобу керування

Перша та четверта відмінності пов'язані тісніше, ніж здається: пристрій, що не має запасу обчислювального ресурсу, не отримує й повноцінного механізму

оновлення, тому вразливість живе стільки ж, скільки сам пристрій. Лічильник, установлений у 2012 році, працюватиме до 2032-го, і алгоритми, достатні в рік випуску, за цей час застаріють. Практичний наслідок формулюється прямо: **у системі завжди є пристрої з відомою незакритою вразливістю, і це стан нормальної експлуатації, а не аварія.** Навіть там, де оновлення технічно можливе, його обмежує канал: якщо парк налічує 40 000 пристроїв, канал дає 20 кБ за сеанс, а образ оновлення важить 180 кБ, кампанія оновлення розтягується на тижні.

Другою вихідною умовою є **фізична доступність** пристрою зловмисникові. Сервер стоїть у приміщенні з контролем доступу, а лічильник висить на зовнішній стіні будинку. Звідси вимога, якої немає у звичайних системах: припускати, що зміст пам'яті пристрою стане відомим супротивникові, а отже не зберігати в ній спільних для всього парку секретів. Другий наслідок — потреба в засобах виявлення втручання, найдешевшим із яких є датчик відкриття кришки.

2. ЧОМУ РІЗНОРІДНІСТЬ ВАЖИТЬ БІЛЬШЕ ЗА ОБСЯГ

Обсяг є інженерною задачею з відомими розв'язками: більше вузлів зберігання, розподілена обробка, стиснення, проріджування. Різнорідність розв'язків не має, бо виявляється на чотирьох рівнях водночас. **Різні одиниці:** один прилад передає кіловат-години наростальним підсумком, інший — миттєву потужність. **Різні формати:** навіть у межах одного парку пристрої різних виробників і років випуску кодують ту саму подію по-різному.

Для задач безпеки корисно розрізняти чотири роди джерел. **Телеметрія** — регулярні вимірювання керованої величини: від одного до кількох сотень мегабайтів на добу для парку в сорок тисяч пристроїв, довіра середня через збої зв'язку та похибку вимірювання. **Журнали подій** надходять нерегулярно, за фактом, важать менше за мегабайт на добу, але мають високу довіру за фактом і низьку за повнотою. **Зведені показники** вимірює окремий прилад, тому довіра до них висока.

3. ДЕВ'ЯТЬ ГРУП ВИМОГ ДО БЕЗПЕКИ

1. Безпека та керування різнорідними великими даними. Зведені докупи телеметрія, журнали й зовнішній контекст утворюють масив, цінніший за будь-яку окрему його частину, і саме тому він стає привабливою ціллю. Вимоги звичні, але ускладнені масштабом: розмежування доступу за ролями, знеособлення вибірок, контроль походження записів, політика зберігання.

2. Полегшена криптографія. Стійкі алгоритми розроблялися з припущенням, що обчислювальний ресурс дешевий; на мікроконтролері з десятками кілобайтів пам'яті це припущення хибне. Пристрої поділяють на три класи: **клас 0** (найпростіший датчик, до 10 КБ пам'яті, батарея на 10 років) здатен лише обчислити код автентичності повідомлення й вести лічильник повторів; **клас 1** (лічильник, замок) витримує симетричне шифрування полегшеним шифром, але не регулярну зміну ключів у полі; **клас 2** (шлюз, контролер) має понад мегабайт пам'яті та постійне живлення й підтримує повноцінний захищений канал із сертифікатами.

3. Універсальна інфраструктура безпеки та її відсутність. Для звичайних мереж існує усталена інфраструктура — посвідчення пристрою, служби видачі сертифікатів, узгоджені протоколи; в інтернеті речей кожен виробник пропонує власну. Наслідок для служби безпеки прямий: єдиного пульта не буде, і замість нього будують спільний рівень збирання, на який кожна підсистема віддає події у власному форматі, а нормалізація виконується вже там.

4. Керування довірою. Довіра тут означає кількісну оцінку того, наскільки можна покладатися на дані конкретного джерела. Схема будується на трьох складниках: початкове значення, що залежить від способу введення пристрою в експлуатацію; поточне значення, яке змінюється за результатами перевірок узгодженості; і правила зниження довіри за ознаками втручання. **5. Керування ключами за відсутності постійного зв'язку.** Три поширені помилки: спільний ключ на весь парк (видобування з одного пристрою компрометує всі); індивідуальні ключі без акуратного обліку відповідності; зміна ключів за розкладом без урахування доступності пристроїв.

6. Збереження приватності. Приватність в інтернеті речей зручно розглядати як п'ять різних властивостей. **Приватність особи** — хто саме користується пристроєм. **Приватність місця** — де людина перебуває й де була раніше: час вмикання світла показує годину повернення додому. **Приватність профілю** — звички та розпорядок: крива споживання виказує сон, роботу, від'їзд у відпустку. **Приватність зв'язків** — хто ще буває в помешканні: поява нових пристроїв у домашній мережі виказує гостей. **Приватність взаємодій** — що саме людина робила з пристроєм.

7. Прозорість означає, що власник пристрою може дізнатися, які саме дані збираються, з якою частотою й кому передаються. Для служби безпеки це не лише правова вимога, а й технічна зручність: система, у якій склад переданих даних описаний явно, піддається перевірці на відхилення. **8. Протидія шахрайству** охоплює кілька різнорідних явищ: рекламне шахрайство виявляють за неприродним розподілом часу між діями; банкоматне — за подіями відкриття корпусу й нетиповими сеансами зв'язку; нетехнічні втрати в мережах постачання — за небалансом підстанції та падінням показань окремих приладів.

4. АНАЛІЗ ВЕЛИКИХ ДАНИХ ДЛЯ БЕЗПЕКИ ІНТЕРНЕТУ РЕЧЕЙ

Задача аналізу тут будується знизу вгору. На рівні **одного набору** працюють звичайні прийоми виявлення аномалій: порівняння поточного ряду з власним історичним профілем пристрою, перевірка фізичної правдоподібності показань, контроль безперервності нумерації записів. Цей рівень дешевий і дає найбільше хибних спрацювань, бо окреме відхилення має безліч невинних пояснень — від сезонності до заміни обладнання.

Наступний рівень — **злиття джерел**. Показання лічильника, що раптово впали, самі собою нічого не доводять; ті самі показання разом із подією відкриття корпусу, зафіксованою за дві доби до падіння, і відсутністю наряду на обслуговування в цей період утворюють обґрунтовану підставу для перевірки. Саме тому зведення різнорідних джерел до спільної схеми, яке видається рутинною інженерною роботою, насправді є умовою можливості висновку. Третій

рівень — **групові показники**: небаланс між сумою показань абонентів і зведеним виміром на підстанції вказує на втрати, не називаючи винного, і служить орієнтиром для вибору ділянки перевірки.

Проєктні обмеження, які визначають вигляд такої системи, випливають із перших двох розділів лекції. Дані надходять із затримкою та пропусками, тому алгоритм має бути стійким до неповноти. Пристрої фізично доступні зловмисникові, тому сама телеметрія не є довіреним джерелом і потребує зіставлення з незалежним вимірюванням. Частина парку завжди має незакриті вразливості, тому виявлення компенсує неможливість усунення. Нарешті, відповідальність розподілена між виробником, оператором і власником, тому результат аналізу має бути поданий у формі, придатній для передавання іншій стороні — з описом методики, вихідних даних і ступеня впевненості.

ВИСНОВКИ

1. Пристрій інтернету речей відрізняється від звичайного вузла мережі обчислювальним ресурсом, живленням, характером зв’язку, можливістю оновлення та фізичною доступністю — і кожна відмінність змінює вимоги до захисту.

2. Наявність пристроїв із відомою незакритою вразливістю є станом нормальної експлуатації, тому виявлення компенсує неможливість усунення.

3. Для даних інтернету речей різнорідність є складнішою задачею за обсяг: близько сімдесяти відсотків зусиль припадає на зведення до спільної схеми.

4. Обчислювально слабкий пристрій не повинен виконувати криптографічних операцій, яких не витримує, — цю роботу переносять на шлюз.

5. Приватність розпадається на п’ять різних властивостей, і лише одну з них захищає шифрування вмісту; решта витікає через факт, час і обсяг передавання.

ЛЕКЦІЯ 11. АНАЛІЗ ВЕЛИКИХ ДАНИХ ДЛЯ БЕЗПЕКИ ТУМАННИХ ОБЧИСЛЕНЬ

Мета лекції: засвоїти означення та властивості туманних обчислень і місце туманного рівня між пристроями та хмарою, навчитися проводити межу передавання даних і розуміти, як відмови туманного рівня впливають на виявлення.

План лекції: 1. Чому хмарних обчислень виявилося замало. 2. Основи туманних обчислень. 3. Межа передавання даних і три компроміси. 4. Аналітика для безпеки туманного рівня.

Ключові терміни: туманний вузол, край мережі, затримка реакції, переривчастий зв'язок, межа передавання даних, керування довірою, агрегування на краю, мовчазна деградація, кореляція подій.

1. ЧОМУ ХМАРНИХ ОБЧИСЛЕНЬ ВИЯВИЛОСЯ ЗАМАЛО

Хмарні обчислення розв'язали задачу потужності: будь-який обсяг даних можна зберегти й обробити, якщо додати вузлів. Проте чотири обставини показали, що самої потужності недостатньо. **Затримка реакції.** Відповідь від хмари складається з часу передавання до центру, часу обробки й часу зворотного передавання, що разом дає сотні мілісекунд. Для нарахування за спожиту електроенергію це прийнятно; для зупинки верстата, спрацювання шлагбаума чи блокування доступу — ні. Туманний рівень з'явився не тому, що хмари замало за потужністю, а тому, що до хмари далеко.

Вартість каналу. Дванадцять камер із потоком 2 Мбіт/с дають близько 2,6 ТБ на місяць, і передавання цього обсягу до центру коштує дорожче за саме обладнання майданчика. Обробка на краю розв'язує задачу інакше: туманний вузол аналізує потік на місці й передає далі не кадри, а висновки — позначки подій і поодинокі кадри-зразки. **Вимоги приватності.** Є дані, передавання яких за межі приміщення обмежене правовими або договірними умовами; туманний

рівень дає природний вихід, бо вихідні дані залишаються на вузлі, а назовні йде знеособлена ознака.

Чотири причини варто запам'ятати в парі з їхніми наслідками для безпеки: затримка означає, що рішення ухвалюють на неповних даних; економія трафіку — що непередане неможливо перевірити згодом; приватність — що частина доказів лишається поза центром; автономність — що рішення вузлів не узгоджені між собою.

2. ОСНОВИ ТУМАННИХ ОБЧИСЛЕНЬ

Туманними обчисленнями називають модель, у якій обчислювальні ресурси, сховище даних і мережеві служби розміщують між пристроями та хмарою — ближче до джерела даних, ніж центр, але далі, ніж сам пристрій. Три рівні відрізняються ресурсом, затримкою та охопленням: **пристрій** має десятки кілобайтів пам'яті, живиться від батареї, відповідає за мілісекунди, бачить лише власні вимірювання й зберігає десятки записів; **туманний вузол** має одиниці гігабайтів і кілька ядер, мережеве живлення, затримку в десятки мілісекунд, бачить свою ділянку з десятків-сотень пристроїв і зберігає історію за добу-тиждень; **хмара** практично не має межі за ресурсом, бачить усю систему й зберігає роки історії, але відповідає за сотні мілісекунд.

Властивості, заради яких туманний рівень будують, мають зворотний бік, і їх корисно розглядати парами (таблиця 11.1).

Другий і п'ятий рядки пояснюють більшу частину подальшого викладу. Відсутність точки повного огляду робить кореляцію між ділянками окремою задачею, а не властивістю системи; те, що не передали на центральний рівень, неможливо перевірити згодом, хоч би яким цінним виявився цей запис через місяць. Окремо слід зважати на фізичне розміщення: туманний вузол стоїть у технічному приміщенні житлового будинку, на стовпі або в шафі біля дороги, тобто там, де фізичний доступ до нього значно простіший, ніж до серверної.

Таблиця 11.1 – Властивості туманних обчислень і зворотний бік кожної

Властивість	Що дає	Зворотний бік для безпеки
Близькість до джерела	затримка відповіді у десятки мілісекунд	вузол стоїть у доступному місці: фізичний доступ до ключів і правил
Географічна розподіленість	відмова одного вузла не зупиняє систему	немає жодної точки, де видно всю картину подій одночасно
Автономність	робота за відсутності зв'язку з центром	рішення ухвалюються за неповними даними й не узгоджуються
Локальне зберігання	дані не покидають приміщення	втрата вузла означає втрату його журналів назавжди
Обробка на місці	трафік зменшується на порядки	те, що не передали, неможливо перевірити згодом
Велика кількість вузлів	масштабованість без перебудови центру	оновлення правил доходить до вузлів нерівномірно
Різнорідність вузлів	можна використати наявне обладнання	єдиного засобу керування немає, рівень захисту різний

На практиці трапляються три архітектурні схеми. **Ієрархічна:** пристрої приєднані до туманного вузла, вузли — до центру; схема проста й передбачувана, але одна відмова знеструмлює цілу гілку. **Пласка:** усі вузли рівноправні й підпорядковані центру безпосередньо. **Змішана з горизонтальним обміном:** вузли, крім зв'язку з центром, обмінюються між собою короткими повідомленнями, що дозволяє зберегти частину кореляції навіть за втрати зв'язку з центром.

Аналітика на туманному рівні розвинена значно слабше за хмарну, і причини цього методичні, а не технічні. Хмарні засоби проєктувалися в припущенні, що всі дані зібрані в одному місці, обчислювальний ресурс необмежений, а історія доступна за роки. На туманному вузлі жодна з цих умов не виконується. Практичний наслідок: більшість відомих методів не переносяться на туманний рівень без переробки — навчання моделі лишається в центрі, а на вузол передають уже навчену модель із обмеженим обсягом станів.

3. МЕЖА ПЕРЕДАВАННЯ ДАНИХ І ТРИ КОМПРОМІСИ

Центральне проєктне рішення туманної системи — провести **межу передавання даних**, тобто визначити, що вузол робить сам і що віддає далі. Потік відео обробляють на вузлі, передаючи позначки подій і кадри-зразки, бо обсяг на порядки перевищує канал. Показання датчиків обробляють на вузлі, передаючи агрегати за період і викиди, бо рішення потрібне швидше, ніж відповідь центру. Журнал автентифікації обробляють і там, і там, передаючи зведення «адреса — кількість», бо частина атак видно лише в сукупності по всіх ділянках.

Рішення про межу спирається на п'ять ознак, які перевіряють послідовно. **Строк придатності висновку:** якщо реагувати треба швидше, ніж приходить відповідь від центру, обробка має бути на вузлі. **Достатність локального огляду:** якщо ознаку явища видно в межах одного вузла, її обчислюють на вузлі. **Коефіцієнт стиснення** — відношення обсягу вихідних даних до обсягу висновку; для відео він становить тисячі, для журналу автентифікації — одиниці.

Три показники туманної системи безпеки не можна поліпшити одночасно (рисунок 11.1). **Повнота виявлення** — частка явищ, які система взагалі помічає; **затримка** — час від події до реакції; **трафік** — обсяг, переданий до центру. Перенесення обробки на вузол зменшує затримку й трафік, але знижує повноту, бо вузол не бачить картини сусідніх ділянок. Перенесення обробки в центр підвищує повноту ціною затримки й трафіку.



Рисунок 11.1 – Три показники, які не можна поліпшити одночасно

Порядок вибору такий: спершу повнота, бо її втрату не компенсує ніщо — те, чого система не бачить, не має ні затримки, ні вартості; потім затримка, визначена вимогами процесу; і лише потім трафік, який є питанням грошей, а не можливостей. Окремо слід розуміти, **що саме втрачається під час агрегування**. Замість запису «о 14:07 з адреси X невдала спроба входу» передається «за годину 12 невдалих спроб». Втрачається точний час усередині періоду, тому неможливо встановити, чи йшли спроби рівномірно, чи спалахом за десять секунд; втрачається зв'язок із конкретними обліковими записами; втрачається можливість зіставити подію з подією сусідньої ділянки з точністю до секунди.

4. АНАЛІТИКА ДЛЯ БЕЗПЕКИ ТУМАННОГО РІВНЯ

Керування довірою. У хмарній моделі довіра однобічна: пристрій довіряє центру, центр перевіряє пристрій. На туманному рівні з'являються горизонтальні відносини між рівноправними вузлами, і схема оцінювання довіри до сусіда спирається на три складники: підтвердження особи вузла, історію узгодженості його повідомлень із власними спостереженнями та правила зниження довіри за ознаками втручання. Типова помилка проєктування — вважати, що всередині периметра всі вузли рівноправні й довіряють один одному беззастережно: вузол, фізично вилучений зловмисником, автоматично отримує повну довіру сусідів.

Керування ідентичністю та доступом за переривчастого зв'язку. Перевірка прав у звичайній системі спирається на можливість звернутися до служби посвідчення в момент перевірки; на туманному рівні такої можливості може не бути. Вибір лежить між двома крайнощами: забороняти все за відсутності зв'язку означає зробити систему непрацездатною саме тоді, коли вона найпотрібніша, а дозволяти все — перетворити обрив каналу на спосіб обходу захисту.

Керування доступністю. У хмарі відмова вузла обробляється резервуванням і лишається непомітною; на туманному рівні відмова є нормою, а не винятком, і кожен її тип по-своєму деградує виявлення. Втрата зв'язку з центром зупиняє кореляцію між вузлами, хоча локальні правила діють. Відмова

живлення робить ділянку невидимою, а події за час відмови зникають назавжди. Фізичне вилучення вузла означає ще й компрометацію ключів. Розсинхронізація годинника дає хибні зв'язки під час кореляції.

Де тримати кореляцію. Кореляція подій — зіставлення сигналів з різних джерел для побудови спільного висновку, і питання, де її виконувати, визначає сліпі зони системи. Кореляція лише на вузлі швидка, але не бачить явищ, розподілених між ділянками: повільний перебір паролів, що дає по дві спроби на кожному з тридцяти вузлів, локально не помітний. Кореляція лише в центрі бачить таке явище, але із затримкою й лише в межах того, що було передано. Змішана схема — локальні правила на вузлі плюс періодичне надсилання зведень до центру — є практичним компромісом, і саме склад цих зведень визначає, які розподілені явища система здатна побачити.

ВИСНОВКИ

1. Туманний рівень виник не через брак потужності хмари, а через затримку, вартість каналу, правові обмеження на передавання й потребу працювати за відсутності зв'язку.

2. Кожна властивість туманного рівня має зворотний бік: близькість до джерела означає фізичну доступність вузла, автономність — неузгодженість рішень, обробка на місці — неможливість перевірити непередане.

3. Межу передавання даних проводять за п'ятьма ознаками, три з яких є заборонними й не підлягають економічному розгляду.

4. Повноту виявлення, затримку й трафік не можна поліпшити одночасно; першою обирають повноту, бо її втрату не компенсує ніщо.

5. Склад зведень, які вузол надсилає до центру, визначає, які запитання система зможе поставити згодом.

ЛЕКЦІЯ 12. АНАЛІЗ ДЕВІАНТНОЇ СОЦІОТЕХНІЧНОЇ ПОВЕДІНКИ

Мета лекції: засвоїти методику спільного застосування аналізу соціальних мереж і кіберкриміналістичних прийомів для розрізнення стихійного поширення та координованої кампанії, а також межі допустимого висновку.

Застереження. Прийоми цієї лекції застосовні до реальних людей, і помилка тут коштує дорожче за технічну. Усі числові приклади отримано на синтетичних модельних наборах; збирати для навчальних цілей дані реальних платформ заборонено, а результати не можна подавати як такі, що стосуються будь-якої реальної особи чи організації.

План лекції: 1. Що таке девіантна соціотехнічна поведінка. 2. Два інструменти та їх поєднання. 3. Методика дослідження. 4. Межі висновку та етичні обмеження.

Ключові терміни: соціотехнічна поведінка, граф поширення, центральність за ступенем, за посередництвом і за близькістю, спільнота, роль учасника, синхронність, щільність ядра, хибна ознака координації.

1. ЩО ТАКЕ ДЕВІАНТНА СОЦІОТЕХНІЧНА ПОВЕДІНКА

Соціотехнічною називають поведінку, у якій дії людей і робота технічних засобів нероздільні: результат виникає не з одного й не з іншого окремо, а з їх поєднання. **Девіантною** таку поведінку називають тоді, коли вона відхиляється від звичайного способу користування платформою — передусім через узгодженість дій, яку учасники приховують. Від самого початку слід розрізнити два різних питання. Перше — **про зміст**: чи є твердження правдивим.

Головна практична складність полягає в тому, що за обсягом стихійне поширення й керована кампанія не відрізняються: повідомлення, яке стихійно зацікавило багатьох, і повідомлення, яке просувала узгоджена група, можуть мати однакову кількість поширень. Відрізняються вони структурою й часом (таблиця 12.1).

Таблиця 12.1 – Стихійне поширення проти координованої кампанії: сім ознак

Ознака	Стихійне поширення	Координована кампанія
Розподіл часу між поширеннями	важкий хвіст: більшість реакцій із затримкою від хвилин до діб	згущення у вузьких вікнах: десятки записів публікують протягом кількох хвилин
Вік облікових записів	розсіяний за роками існування платформи	помітна частка записів створена у вікні кількох діб
Посилання	багато різних джерел, частка найбільшого мала	висока частка кількох власних ресурсів
Структура ядра	ядра немає або воно розріджене	щільний підграф: учасники ядра поширюють один одного
Ролі учасників	розділені: автор один, мости виникають самі	суміщені: ядро водночас створює зміст і з'єднує його з аудиторією
Згасання теми	поступове, одна хвиля	кілька хвиль із близькими параметрами
Поведінка поза темою	записи мають власну історію, не пов'язану з темою	частина записів майже не має іншої активності

Жодна із семи ознак не є доказом сама по собі, і це твердження слід запам'ятати раніше за самі ознаки: для кожної існує невинне пояснення, і лише їх сукупність за умови незалежності може підтримати висновок. Питання, чому аналіз інформаційних кампаній належить до курсу кібербезпеки, має три відповіді. Об'єктом аналізу є **не люди, а дані систем** — журнали публікацій, зв'язки поширення, часові мітки, відомості про реєстрацію ресурсів.

2. ДВА ІНСТРУМЕНТИ ТА ЇХ ПОЄДНАННЯ

Аналіз соціальних мереж працює зі структурою зв'язків: він не читає повідомлень і не оцінює їх змісту, для нього існують лише вузли та ребра. Він відповідає на три групи запитань: хто важливий і в якому сенсі (метрики центральності), які групи утворюють учасники (виявлення спільнот) і як поширення розгорталося в часі. Принципово він **не вміє** відрізнити автоматичний обліковий запис від людського, не бачить, хто кому платить, і не встановлює намірів.

Кіберкриміналістичний складник працює не зі структурою, а зі слідами технічних засобів: коли створено облікові записи, на які домени ведуть посилання,

хто зареєстрував ці домени й коли, якими засобами виконано публікацію. Ці відомості мають іншу природу, ніж граф: вони здебільшого бінарні — домени або збігаються, або ні. Межі складника теж слід назвати прямо: **збіг інфраструктури доводить спільність засобу, а не спільність намірів**; спільний хостинг, спільний готовий інструмент або спільний підрядник дають той самий слід, що й узгоджена кампанія.

Кожен із підходів окремо дає картину, яку легко пояснити інакше: аналіз мережі знаходить щільне ядро, але щільне ядро утворює і спільнота за інтересами; криміналістика знаходить спільний домен, але спільний домен має і звичайне популярне джерело. Приріст дає лише **незалежність помилок** — та сама умова, що й для злиття джерел у лекції про інтернет речей. Практичний критерій незалежності формулюється запитанням: чи може ця ознака змінитися, якщо змінити лише ту обставину, що вже врахована іншою ознакою?

3. МЕТОДИКА ДОСЛІДЖЕННЯ

Послідовність дій дослідження показано на рисунку 12.1: від збору даних і побудови графа через обчислення метрик і перевірку кількісних ознак до висновку із зазначеними межами.



Рисунок 12.1 – Методика: від збору даних до висновку із зазначеними межами

3.1 Збір даних і побудова графа

Мінімальний набір полів складається з чотирьох частин: повідомлення (ідентифікатор, автор, час публікації), зв'язки поширення, часові мітки та посилання. Два питання вирішують на цьому ж етапі, і від них залежить усе подальше: **межа вибірки** — за якою ознакою повідомлення вважається належним до теми — і **повнота** — яку частку наявних повідомлень вдалося отримати. Обидва записують у звіт на початку, а не наприкінці: показник, обчислений на неповній вибірці без зазначення цього факту, є не результатом, а джерелом помилки.

Вузлами графа є облікові записи, ребрами — факти поширення; напрям ребра обирають так, щоб він відповідав напряму руху повідомлення. Три рішення змінюють результат і тому мають бути записані. **Напрям:** орієнтований граф потрібен для ступеневих метрик, неорієнтований — для спільнот. **Ваги:** чи враховувати кратність поширень між тією самою парою. **Обмеження компонентою:** чи аналізувати лише найбільшу зв'язну компоненту. Останнє рішення має побічний наслідок, про який забувають: записи, що не потрапили до найбільшої компоненти, отримують нульові значення метрик і зникають із аналізу, хоч фізично існують.

3.2 Метрики центральності та ролі

Слово «центральний» у мережевому аналізі не має єдиного значення. **Центральність за ступенем** — кількість ребер вузла — відповідає на питання, кого поширюють найчастіше, і велике значення означає помітність. **Центральність за посередництвом** — частка найкоротших шляхів між усіма парами вузлів, що проходять через даний вузол, — показує, через кого проходять зв'язки між частинами мережі. **Центральність за близькістю** — обернена сума відстаней до всіх інших вузлів — показує, хто швидше за інших дістає всю мережу.

Найчастіша помилка тлумачення у всій темі — ототожнення **активності та впливу**. Кількість публікацій не є метрикою мережі: її можна порахувати, не

будуючи графа. У модельному наборі найактивніший запис теми опублікував 180 повідомлень, проте його посередництво мале, бо його майже не поширюють далі — він говорить багато, але в порожнечу. Протилежний випадок — запис із десятьма публікаціями й посередництвом 0,38: через нього проходить тридцять вісім відсотків найкоротших шляхів, тобто саме він з'єднує частини мережі.

Ролі учасників визначають за поєднанням метрик і обсягу публікацій. **Виробник змісту** має високий вихідний ступінь за помірної кількості публікацій: його вилучення позбавляє тему джерела. **Міст** має високе посередництво за помірних інших показників: його вилучення розриває каскад. **Поширювач** публікує багато, власних поширень має мало: його вилучення зменшує обсяг, не змінюючи структури. **Периферія** реагує один раз і виходить. Співвідношення ролей саме по собі є ознакою: у модельному наборі в координованій темі частка поширювачів помітно вища, ніж у стихійній.

3.3 Кількісні ознаки координації та порядок їх перевірки

Для зведеної оцінки ознаки зводять до чотирьох вимірюваних величин, нормованих у проміжок від нуля до одиниці. **Синхронність** — частка публікацій у вікнах, де публікує десять і більше записів: у модельній кампанії 0,68 проти 0,30 у стихійній темі. **Вік записів** — найбільша частка записів, створених у вікні семи діб: 0,55 проти 0,00. **Спільні домени** — частка посилань на два найчастіші ресурси: 0,59 проти 0,00. **Щільність ядра** — щільність підграфа двадцяти найцентральніших записів: 0,74 проти 0,14. Зважена сума з вагами 0,30 / 0,30 / 0,25 / 0,15 дає 0,63 для кампанії та 0,11 для стихійної теми.

Порядок перевірки має значення не менше, ніж самі ознаки: спочатку для кожної ознаки формулюють альтернативне пояснення й перевіряють його, і лише ознаки, що вистояли, входять до зведеної оцінки. Три найпоширеніші **хибні ознаки** варто знати в числах. **Синхронність, що виникає сама:** органічні записи, які швидко відреагували на спільну подію, дають помітну синхронність без жодної координації — добовий пік активності імітує ту саму картину.

4. МЕЖІ ВИСНОВКУ ТА ЕТИЧНІ ОБМЕЖЕННЯ

Кожна кіберкриміналістична ознака доводить менше, ніж здається. Вік облікових записів доводить, що записи створено одним заходом, але не те, що ними користується одна особа. Спільні домени посилань доводять спільність джерела або засобу, але не спільність намірів. Реєстраційні відомості доменів доводять технічний зв'язок між ресурсами, але не те, хто за ними стоїть. Рівномірність інтервалів доводить використання засобу планування, яким цілком законно користуються редакції та маркетингові служби. Збіг технічних позначок доводить спільний засіб публікування, доступний будь-кому.

Звідси випливають правила формулювання висновку. Висновок подають **із зазначенням меж**: що саме встановлено, на якій вибірці, за який період, з якою повнотою даних і яке альтернативне пояснення перевірялося. Формулювання «встановлено ознаки координованого поширення в межах дослідженої вибірки» є коректним; формулювання «кампанію організувала така-то сторона» — ні, бо структурні й технічні ознаки не встановлюють замовника. Окремо фіксують, що висновок стосується **способу поширення, а не змісту**: узгоджено можуть поширювати і правдиве твердження.

Етичні обмеження є не додатком до методики, а її частиною. Аналіз виконують на знеособлених даних, зберігаючи зв'язок із конкретними особами лише там, де це необхідно й правомірно. Перелік облікових записів, віднесених до ядра, не є переліком винних і не може публікуватися як такий: помилкова атрибуція стосовно реальної особи завдає шкоди, яку неможливо відкликати. Нарешті, методика є симетричною — тими самими прийомами можна аналізувати й того, хто їх застосовує, — тому вимога документувати рішення та альтернативні пояснення захищає передусім самого аналітика.

ВИСНОВКИ

1. Методика розрізняє спосіб поширення, а не правдивість змісту; змішування цих питань є найсерйознішою помилкою теми.

2. Стихійне поширення й координована кампанія не відрізняються за обсягом, але відрізняються розподілом часу, віком записів, часткою спільних джерел і щільністю ядра.

3. Жодна окрема ознака не є доказом: кожна має невинне пояснення, і приріст дає лише сукупність ознак із незалежними помилками.

4. Аналіз мережі не бачить намірів, а криміналістичні сліди доводять спільність засобу, а не спільність задуму; результат дає лише поєднання двох підходів.

5. Активність і вплив — різні величини: найактивніший учасник може не мати жодного впливу, а учасник із десятком публікацій — утримувати зв'язність усієї мережі.

ЛЕКЦІЯ 13. ЗАСОБИ КІБЕРБЕЗПЕКИ ТА МЕТОДИКА ЇХ ДОБОРУ

Мета лекції: розібрати функціональні групи засобів захисту та задачі, які кожна з них справді розв'язує, і засвоїти методику добору за матрицею покриття етапів атаки, а не за переліком заявлених можливостей.

Зауваження. Лекція навмисно не містить назв конкретних продуктів: назви старіють швидше за методику, а перенесення оцінки з одного продукту на інший без перевірки є типовою помилкою. Усі числові значення отримано на навчальній моделі.

План лекції: 1. Добір засобів як інженерна задача. 2. Засоби межі мережі та моніторингу. 3. Захист і криміналістика пам'яті, керування автентифікацією. 4. Матриця покриття, сліпі зони, надлишковість і бюджет.

Ключові терміни: функціональна група засобів, етапи атаки, матриця покриття, фактичне покриття, сліпа зона, надлишковість, найгірший сценарій, очікувані сумарні витрати.

1. ДОБІР ЗАСОБІВ ЯК ІНЖЕНЕРНА ЗАДАЧА

Служба захисту інформації невеликої організації майже ніколи не обирає засоби з чистого аркуша: частину успадковано, частину придбано під вимогу регулятора, частину — за рекомендацією. Звичний спосіб оцінити стан — скласти перелік засобів і перевірити, чи кожен етап можливої атаки кимось «закритий». Такий підрахунок не розрізняє **заявлену можливість** і **фактичну ймовірність виявлення**: виробник пише, що засіб «виявляє підвищення прав», і цього достатньо, щоб поставити позначку, хоча ймовірність виявлення може становити 0,04.

Зручно розрізняти п'ять функціональних груп. **Межа мережі** працює в точці входу та виходу трафіку, спостерігає з'єднання, адреси, порти й вміст сеансів і розв'язує задачу «не пропустити відоме небажане». **Вузол** працює в операційній системі робочої станції та сервера, спостерігає процеси, файли, виклики й пам'ять і

виявляє виконання чужого коду. **Моніторинг** працює на дзеркалі трафіку, бачить, хто з ким і скільки спілкувався, і показує переміщення всередині мережі.

Щоб порівняти екран периметра з менеджером паролів, потрібна шкала, на якій обидва мають значення; такою шкалою є **послідовність етапів атаки** — від первинного доступу й виконання коду через підвищення прав і внутрішню розвідку до вивантаження даних. Два зауваження про цю шкалу. По-перше, вона не є законом природи: етапи виділено для зручності порівняння. По-друге, засоби заявляють покриття нерівномірно — початок ланцюга й переміщення мережею заявляють дві-три групи, тоді як середина ланцюга лишається найменш охопленою.

2. ЗАСОБИ МЕЖІ МЕРЕЖІ ТА МОНІТОРИНГУ

Міжмережеві екрани поділяються на три класи. На боці **оператора зв'язку** засіб знає обсяг, адреси й репутацію джерел і покриває великі потоки на вичерпання смуги; він не знає, які сеанси є законними для конкретної організації. **Домашні та малих офісів** пристрої знають адреси, порти й напрям з'єднання та зупиняють масові спроби підключення ззовні, але не заважають з'єднанню, встановленому зсередини; їхня окрема проблема — оновлення, бо пристрій купують один раз і забувають, а виробник припиняє підтримку.

Антивірусний засіб вирішує задачу, яку ніхто інший не вирішує так дешево: він упізнає відоме. Масовий зразок із розсилання виявляється майже завжди, перепакований відомий зразок — здебільшого, шифрування файлів користувача — здебільшого, бо послідовність дій над файлами розпізнається за поведінкою. Є три класи подій, яких засіб не виявляє **не через недоопрацювання, а за побудовою**: унікальний зразок, виготовлений для однієї організації, бо в базі його немає; дії вбудованими засобами системи, бо запускається штатна програма з дозволеним підписом; дії від імені легітимного користувача з його правами.

Фільтрування вмісту перевіряє не адресу, а те, що передається, і дає найбільший внесок на етапі первинного доступу. Побічні наслідки відомі: перевірка вмісту вимагає розкривати захищені сеанси, тож організація бачить

листування працівників, що створює правові та етичні зобов'язання. **Пасивне спостереження за трафіком** дає повний вміст сеансів у точці спостереження, але обсяг робить тривале зберігання неможливим. **Записи потоків** є компромісом: вони відповідають на більшість питань розслідування — хто з ким, коли, скільки й у який бік, — але не розрізняють, що саме передавалося, тому законне резервне копіювання й вивантаження даних виглядають однаково.

Виявлення вторгнень поділяють на два типи. Засіб **за сигнатурами** майже не дає хибних спрацювань, але має нульову чутливість до нового. Засіб **за відхиленнями** помічає незвичну поведінку, але не розрізняє напад і зміну роботи організації, тому потребує людини для розбору. Ці два типи не конкурують, а доповнюють один одного, і бюджет на другий має включати вартість робочого часу аналітика. Класичною помилкою побудови спостереження є розміщення всіх точок збору на межі мережі: напад приходить ззовні, отже, спостерігаємо межу.

3. ЗАХИСТ ПАМ'ЯТІ ТА КЕРУВАННЯ АВТЕНТИФІКАЦІЄЮ

Засоби захисту пам'яті працюють не з файлами, а з тим, як програма поводиться під час виконання: вони унеможливлюють виконання коду в ділянках даних, контролюють цілісність потоку керування й перешкоджають типовим прийомам експлуатації. Саме тому вони мають значення там, де антивірусний засіб безсилий — на етапах виконання коду та підвищення прав. Ціна цього — сумісність: захист пам'яті ламає застарілі галузеві застосунки й драйвери без підтримки.

Менеджер паролів розв'язує реальну задачу: людина не здатна пам'ятати сорок різних складних паролів, тому без засобу вона або повторює той самий пароль, або записує його. Менеджер прибирає обидві проблеми й є найдешевшим засобом у моделі, але створює нову точку відмови. Казати «менеджер паролів небезпечний, бо все в одному місці» означає порівнювати його з ідеалом, а не з реальною альтернативою — повторюваним паролем на сорока ресурсах. Типовий хибний висновок — «ми впровадили менеджер паролів, отже, питання паролів

закрито»: насправді головний пароль без другого фактора залишає етап збирання облікових даних відкритим.

Багатофакторна автентифікація змінює економіку нападу сильніше за будь-який інший засіб: крадений пароль перестає бути достатнім. Обходити її теж навчилися — посередництво в реальному часі, втома від запитів, підміна телефонного номера, — і ці способи визначають, яке саме впровадження має сенс. Типові помилки впровадження об'єднує одна властивість: **жодна з них не видно в переліку можливостей**. Винятки «для зручності» для окремих застосунків знижують фактичне покриття до рівня звичайного пароля; другий фактор у тому самому пристрої означає, що компрометація пристрою знімає обидва; відновлення доступу слабшим способом створює обхідний шлях входу; відсутність журналу невдалих спроб робить етап збирання облікових даних взагалі неспостережуваним.

4. МЕТОДИКА ДОБОРУ

4.1 Матриця покриття

Матриця покриття — прямокутна таблиця, рядками якої є засоби, стовпцями — етапи атаки, а в клітинці стоїть оцінка ймовірності виявлення. Головне правило заповнення: оцінка має бути **ймовірністю, а не галочкою**, бо галочка не розрізняє засіб, що виявляє напад у дев'яти випадках із десяти, і засіб, що виявляє його в одному з двадцяти п'яти (таблиця 13.1).

Підсумковий рядок обчислюється не додаванням: ймовірність виявити напад на етапі хоча б одним засобом дорівнює одиниці мінус добуток ймовірностей не виявити його кожним, тобто $1 - \Pi(1 - p_i)$. Саме ця формула пояснює, чому заявлення не додаються: два засоби з покриттям 0,30 і 0,25 дають не 0,55, а 0,475, і кожен наступний засіб додає дедалі менше. **Сліпою зоною** називають етап, фактичне покриття якого нижче за встановлений поріг — у навчальній моделі 0,40. Результат застосування матриці до повного набору засобів показано на рисунку 13.1.

Таблиця 13.1 – Матриця покриття етапів атаки: фрагмент на шість засобів

Засіб	S02 первинний доступ	S03 виконання коду	S05 підвищення прав	S08 внутрішня розвідка	S10 вивантаження	Вартість, тис. грн
Фільтр пошти та вмісту	0,30	0,08	—	—	—	140
Антивірусний засіб станцій	—	0,25	0,04	—	—	120
Засіб захисту пам'яті	—	0,20	0,18	—	—	130
Виявлення за відхиленнями	—	—	—	0,15	0,15	160
Збирання та кореляція подій	0,07	0,07	0,07	0,07	0,07	220
Багатофакторна автентифікація	0,22	—	—	—	—	90
Фактичне покриття етапу	0,49	0,49	0,27	0,21	0,21	860



Рисунок 13.1 – Матриця покриття на 14 засобів і фактичне покриття кожного етапу повним набором

Рисунок дає головний кількісний результат лекції. Повний набір із 14 засобів вартістю 1 675 тис. грн на рік не усуває сліпих зон: три етапи лишаються нижче порога, і їхні порядкові номери — п'ятий, шостий і восьмий, тобто

середина ланцюга, а не його початок і не кінець. Зіставлення двох чисел робить картину наочною: етап внутрішньої розвідки заявляють п'ять засобів із чотирнадцяти — більше, ніж будь-який інший, — а фактичне покриття цього етапу виявляється найнижчим. Звідси правило для практики: **якщо в матриці покриття немає жодної сліпої зони, це майже напевно означає не те, що захист повний, а те, що клітинки заповнено галочками замість ймовірностей.**

4.2 Надлишковість і бюджетне обмеження

Надлишковість двох засобів — частка їхнього покриття, яка збігається; обчислюють її як відношення суми найменших значень у парі до суми найбільших. Найнадлишковішою парою моделі є антивірусний засіб робочих станцій і контроль застосунків: вони працюють на тих самих етапах, тому другий додає значно менше, ніж коштує. Високу надлишковість не слід плутати з дублюванням для надійності: дублювання планують свідомо для критичних етапів, надлишковість виникає сама й оплачується двічі.

Три набори, зібрані в межах одного бюджету 800 тис. грн трьома різними критеріями добору, за **середньою повнотою** майже не відрізняються: 0,852, 0,861 і 0,849 — різниця в межах одного відсоткового пункту. За **найгіршим сценарієм**, тобто за покриттям найслабшого етапу, вони розходяться істотно. Причина в тому, що середнє зважується на довгі шляхи: напад, який проходить десять етапів, має десять нагод бути поміченим, тоді як напад, що експлуатує саме слабкий етап, проходить повз.

Остаточне рішення ухвалюють за **очікуваними сумарними витратами** — сумою вартості володіння засобами та очікуваних збитків від нападів, які вони не спиняють. Такий підхід показує, що набір «без засобів» коштує нуль, але дає найбільші очікувані збитки, тоді як повний набір із 14 засобів має сумарні витрати, вищі за оптимальний проміжний варіант: останні додані засоби коштують більше, ніж зменшують очікувані збитки. Саме цей критерій, а не кількість засобів чи перелік їхніх можливостей, є обґрунтуванням бюджету перед керівництвом.

ВИСНОВКИ

1. Перелік заявлених можливостей не дорівнює захищеності: оцінка має бути ймовірністю виявлення на етапі, а не позначкою про наявність функції.

2. Етапи атаки є спільною системою координат, на якій можна порівняти засоби різної природи.

3. Антивірусний засіб за побудовою не виявляє унікальних зразків, дій штатними засобами системи та дій від імені легітимного користувача — це його область визначення, а не недолік.

4. Моніторинг лише периметра залишає етапи внутрішнього переміщення найгірше покритими, хоча саме вони визначають обсяг компрометації.

5. Фактичне покриття обчислюють за формулою $1 - \Pi(1 - p_i)$, тому заявлення не додаються й кожен наступний засіб додає дедалі менше.

ЛЕКЦІЯ 14. ДАНІ ТА ДОСЛІДНИЦЬКІ ІНІЦІАТИВИ ДЛЯ АНАЛІЗУ КІБЕРБЕЗПЕКИ

Мета лекції: оглянути три групи джерел даних для аналізу кібербезпеки та навчитися добирати джерело під задачу; засвоїти, чому вік набору визначає придатність отриманого на ньому результату, і які поля має містити паспорт набору.

План лекції: 1. Три групи джерел даних. 2. Еталонні набори та сховища. 3. Чому вік набору визначає придатність результату. 4. Обмін даними та паспорт набору.

Ключові терміни: дані операційної системи, записи потоків, повне захоплення пакетів, дані прикладного рівня, еталонний набір, темний простір адрес, старіння набору, зсув розподілів, знеособлення, синтетичний набір, паспорт набору даних.

1. ТРИ ГРУПИ ДЖЕРЕЛ ДАНИХ

Усі прийоми, розібрані в попередніх лекціях, працюють лише настільки, наскільки якісними є дані, на яких їх перевіряють. Найближче до події стоять **дані операційної системи**: журнали подій, записи про запуск і завершення процесів, системні виклики, зміни у файлах і реєстрі. Це єдине джерело, яке відповідає на питання «що робив процес»; жодне спостереження за мережею не скаже, який саме виконуваний файл породив з'єднання.

Дані мережевого трафіку описують не дію, а її слід. **Записи потоків** зберігають для кожного сеансу адреси, порти, протокол, обсяг і тривалість; вони займають одиниці гігабайтів на добу й зберігаються роками. **Повне захоплення пакетів** містить усе передане, але займає одиниці терабайтів на добу і зберігається днями. Різниця між ними не в деталізації, а в тому, які питання можна поставити через рік. **Дані прикладного рівня** — журнали веб-серверів, записи про спроби й результати входу, журнали баз даних — описують подію мовою предметної області, а тому дають найякіснішу розмітку: коли в журналі застосунку записано

«користувач X виконав дію Y», це готова позначка, якої немає в мережевому трафіку (таблиця 14.1).

Таблиця 14.1 – Три групи джерел даних: що кожна дозволяє встановити

Група джерел	На яке питання відповідає	Типовий обсяг	Чого не покаже
Операційна система	що робив процес; хто увійшов; що змінилося у файлах	десятки Гбайт на сервер за добу за повного журналювання	події на інших вузлах; дії до ввімкнення журналювання
Мережевий трафік: записи потоків	хто з ким, коли, скільки, у який бік	одиниці Гбайт на мережу за добу; зберігається роками	вміст передавання; те, що не проходить через точку збору
Мережевий трафік: повне захоплення	що саме передавалося в точці спостереження	одиниці Тбайт на мережу за добу; зберігається кілька діб	вміст захищеного сеансу без розкриття
Прикладний рівень	хто саме і яку дію виконав мовою предметної області	сотні Мбайт на застосунок за добу	дії в обхід застосунку; те, що застосунок не записує

Джерело добирають не за обсягом і не за зручністю доступу, а за питанням, на яке треба відповісти, причому визначальним є не те, що джерело показує, а те, чого воно не показує. Якщо питання перетинає рівні — а більшість цікавих питань їх перетинає, — жодне джерело окремо відповіді не дасть. Окремо слід назвати клас фактів, якого немає **в жодній групі**. По-перше, у даних немає намірів: журнал фіксує дію, а не мету. По-друге, у даних немає того, чого не спостерігали: новий клас нападу, ознаки якого ніхто не збирав, у наборі просто відсутній.

2. ЕТАЛОННІ НАБОРИ ТА СХОВИЩА ДАНИХ

Найвідоміші навчальні набори походять від серії досліджень кінця дев'яностих років. Набори DARPA 1998 і 1999 років містять кілька тижнів змодельованого трафіку з позначеними нападами; похідний від них KDD Cup 1999 подає 41 ознаку на запис, проте понад половина його записів є дублікатами, через що модель фактично навчається на повторах. Виправлена версія NSL-KDD усунула дублікати, але вихідний трафік лишився тим самим.

Другий тип — записи з **навчальних кіберзмагань**, у яких команди атакують і захищають заздалегідь підготовлену інфраструктуру. Їхня вада випливає з природи: частка нападів у трафіку в десятки разів вища за звичайну, бо там усі атакують одночасно, а розмітка походить із журналу самого змагання й фіксує намір організатора, а не фактичний результат. Третій тип — **набори з повним описом сценаріїв**, зібрані навмисно для досліджень: CTU-13 із тринадцятьма сценаріями трафіку керованих мереж, UNSW-NB15 із дев'ятьма родинами нападів.

Окремо від готових наборів існують **майданчики збору**. Посередники між власниками даних і дослідниками надають доступ до даних реальних мереж за заявкою з описом дослідження: такий спосіб розв'язує головну проблему обміну, бо власник не втрачає контролю, але створює іншу — умови доступу змінюються з часом, і набір, відкритий на момент публікації статті, через кілька років може стати недоступним. **Дослідницькі центри постійного спостереження** збирають знеособлені записи магістралі, топологію та дані темного простору адрес; останнє джерело варте окремої уваги, бо будь-який пакет, що прийшов на адресу, якої ніхто не використовує, не може бути наслідком законної дії.

Перед тим як узяти набір у роботу, його перевіряють за чотирма пунктами, кожен із яких має письмову відповідь: хто і чим зібрав дані; якому періоду вони відповідають; як виконано знеособлення; і на яких умовах дозволено використання та публікацію результатів. Останній пункт здається формальним, але має пряме дослідницьке значення: набір із обмеженим доступом робить результат неперевірюваним для інших. До цих чотирьох пунктів додають одну технічну дію, яку виконують один раз і яка рятує згодом, — обчислити й записати контрольну суму файлів набору.

Застарілі набори використовують не через незнання, а тому, що вони безкоштовні, розмічені й дозволяють порівняти новий метод із десятками опублікованих результатів. Це законні підстави, але **лише для однієї задачі** — порівняння методів між собою. Правило, яке варто записати: результат на застарілому наборі можна подавати як порівняння з іншими методами і не можна подавати як оцінку придатності методу до сучасної мережі.

3. ЧОМУ ВІК НАБОРУ ВИЗНАЧАЄ ПРИДАТНІСТЬ РЕЗУЛЬТАТУ

Старіння має чотири різні причини, і плутати їх не можна, бо виправляються вони по-різному. **Зміна протоколів і зникнення ознак.** Ознака існує доти, доки існує те, що вона вимірює: коли в наборі кінця дев'яностих років порт призначення однозначно вказував на службу, це була сильна ознака; сьогодні переважна частина трафіку йде на один порт, і ознака просто перестала розрізняти. Те саме сталося з версією захищеного протоколу: якийсь час застаріла версія справді була ознакою підозрілого з'єднання, а після її повсюдного вилучення ознака зникла.

Зникнення одних класів нападу й поява інших. Клас, що становив у наборі помітну частку, практично зникає, і для моделі це не шкідливо: правило, що його виявляло, лишається правильним, воно просто не спрацьовує. Значно гірша протилежна ситуація — поява нового класу, ознак якого в наборі немає взагалі: модель не помиляється на ньому, вона його не бачить. Частку нового класу слід рахувати не від усіх записів, а від записів цільового класу, тобто від нападів, інакше його значущість систематично занижується.

Зростання частки шифрованого трафіку знецінює не окрему ознаку, а цілий спосіб спостереження: коли більшість сеансів захищена, усі ознаки, що спиралися на вміст, втрачають силу, і натомість з'являється новий набір ознак — метадані захищеного сеансу, відбитки клієнта, тайминги й розміри повідомлень, — якого в старих наборах немає. **Зміна масштабу й швидкостей** знецінює всі абсолютні пороги за обсягом і кількістю: поріг, установлений для мережі на десятки мегабітів за секунду, не має сенсу для мережі на десятки гігабітів.

Усе це можна не припускати, а виміряти. Для кожної ознаки порівнюють її розподіл у старому й новому наборах за індексом стабільності популяції $PSI = \Sigma(a_i - b_i)\cdot\ln(a_i/b_i)$, де a_i і b_i — частки i-ї категорії у двох наборах. Тлумачення показника усталене: значення нижче 0,10 означають, що розподіл практично не змінився; від 0,10 до 0,25 — помітний зсув; понад 0,25 — ознака змінилася настільки, що навчену на ній модель переносити не можна (рисунок 14.1).



Рисунок 14.1 – Старіння набору: зсув розподілів за портом призначення, версією протоколу та класами нападу

Числа навчальної моделі показують масштаб явища: частка сеансів на одному порту впала з 0,54 до 0,08, а на іншому зросла з 0,30 до 0,79, що дає значення показника в кілька одиниць — на порядок вище порогової величини. Протилежний бік тієї самої картини не менш важливий: сім ознак із дванадцяти мають показник нижче 0,10, тобто лишилися стабільними, і саме вони становлять ту частину старого набору, яка досі придатна. Про межі самого показника теж слід сказати: він вимірює величину зсуву й не вказує його напрямку, тому однакове значення вийде і для ознаки, що зникла, і для ознаки, що стала переважною.

4. ОБМІН ДАНИМИ ТА ПАСПОРТ НАБОРУ

Головна перешкода обміну полягає в тому, що корисні дані майже завжди містять відомості про людей і про внутрішній устрій мережі. **Знеособлення** замінює прямі ідентифікатори, але має межу, яку легко недооцінити: воно прибирає ідентифікатори і не прибирає структуру. Якщо збережено часові мітки, обсяги й зв'язки, особу або організацію часто можна встановити повторно — класичні приклади такого повторного встановлення походять не з кібербезпеки, а з медичних записів та історій перегляду.

Синтетичні набори формують програмою за описаною моделлю. Переваг три: немає жодних відомостей про людей, розмітка точна за побудовою, а

результат відтворюваний, бо достатньо передати програму формування та початкове значення генератора. Вада одна, але вагома: **у синтетичному наборі є рівно те, що в нього заклали**, тому модель, яка добре на ньому працює, доводить лише, що вона вивчила закладену модель. Звідси обов'язкова перевірка, яку легко пропустити: чи не потрапила в набір сама відповідь — ознака, що виникла як побічний наслідок формування й прямо корелює з позначкою.

Обмін моделями замість обміну даними змінює сам предмет обміну: організації передають не дані, а навчену модель або її оновлення, і дані не залишають мережі власника. Дві вади варто знати: з параметрів навченої моделі за певних умов можна частково відновити відомості про навчальну вибірку — це пряме продовження теми атак виведення з лекції про машинне рознавчання; і отримувач не може перевірити якість чужих даних, тому приймає модель на віру.

Усе сказане зводиться до однієї практичної вимоги: **кожен набір супроводжується паспортом**. Паспорт не є формальністю — він розв'язує три задачі, яких не розв'язує ніщо інше: робить результат відтворюваним, дозволяє оцінити придатність набору до нової задачі й фіксує умови використання. Обов'язкові поля: назва та версія; походження (хто і чим зібрав або сформував дані, із зазначенням програми та початкового значення генератора для синтетичних наборів); період, якому відповідає стан мережі; обсяг у записах і ознаках; склад класів і їх частки; спосіб розмітки; виконане знеособлення; відомі обмеження та вади; умови використання й публікації; контрольні суми файлів.

ВИСНОВКИ

1. Три групи джерел — дані операційної системи, мережевого трафіку та прикладного рівня — відповідають на різні питання, і жодна з них не замінює іншої.

2. Джерело добирають за питанням, на яке треба відповісти, причому визначальним є не те, що джерело показує, а те, чого воно не показує.

3. У даних немає намірів, немає того, чого не спостерігали, і немає контексту організації — ці три прогалини не усуваються збільшенням обсягу.

4. Похідний набір успадковує всі вади первинного й додає власні; результат на застарілому наборі придатний для порівняння методів і непридатний як оцінка придатності до сучасної мережі.

5. Старіння має чотири причини — зміну протоколів, зміну складу класів, зростання частки шифрованого трафіку та зміну масштабу — і вимірюється індексом стабільності популяції.

СПИСОК ЛІТЕРАТУРИ

1. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. 3rd ed. Sebastopol: O'Reilly Media, 2022. 850 p.

2. Maleh Y., Alazab M., Tawalbeh L. (Eds.). Big Data Analytics and Intelligent Systems for Cyber Threat Intelligence. Gistrup: River Publishers, 2023. 312p.

3. McKinney W. Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter. 3rd ed. Sebastopol: O'Reilly Media, 2022. URL: https://wesmckinney.com/book/.

4. Molnar C. Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. 2nd ed. 2022. URL: https://christophm.github.io/interpretable-ml-book/.

5. Reis J., Housley M. Fundamentals of Data Engineering: Plan and Build Robust Data Systems. Sebastopol: O'Reilly Media, 2022. 445 p.

6. Stamp M. Introduction to Machine Learning with Applications in Information Security. 2nd ed. Boca Raton: Chapman and Hall/CRC, 2023. 548 p.

А64 Аналіз великих даний : конспект лекцій для здобувачів першого (бакалаврського) рівня вищої освіти освітньої програми «Кібербезпека» галузі знань 12 (F) Інформаційні технології спеціальності F5 (125) Кібербезпека та захист інформації денної та заочної форм навчання / уклад. Н.М. Сурова. Луцьк: ЛНТУ, 2026. 105 с.

Методичне видання складене відповідно до робочої програми навчальної дисципліни «Аналіз великих даних» з метою використання здобувачами спеціальності F5 (125) Кібербезпека та захист інформації при вивченні даної дисципліни.

Комп’ютерний набір Н.М. Сурова

Редактор Н.М. Сурова

Підп. до друку «___» _____ 2026 р.
Формат 60х84/16. Папір офс. Гарнітура Таймс.
Ум. друк. арк. ____. Тираж 10 прим. Зам. ____

Відділ іміджу та промоцій
Луцького національного технічного університету
43018, м. Луцьк, вул. Львівська, 75