УДК 004.056 **DOI:10.30837/rt.2023.3.214.04**

В.І. ЄСІН, д-р техн. наук, В.В. ВІЛІГУРА, І.І. СВАТОВСЬКИЙ, канд. техн. наук

ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ У РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ: ОСНОВНІ АСПЕКТИ

Вступ

Забезпечення безпеки розподілених інформаційних систем (РІС) є критично важливим завданням, оскільки ці системи використовуються переважно для обробки та зберігання великого обсягу чутливої/конфіденційної інформації, такої як фінансові дані, медичні записи, персональні дані тощо. Інформація у світі є одним із найважливіших ресурсів суспільства, без необхідного захисту якої нові інформаційні технології здатні порушити приватне життя людей та діяльність різних організацій. В епоху Великих Даних проблема захищеності чутливих даних ще більше загострюється. Хоча глобальні витрати на безпеку, як очікується, перевищать 195 мільярдів доларів у 2025 р., зломи стають все масштабнішими, зухвалими, зачіпаючи все: від баз даних клієнтів і громадян до даних про вакцини і маршрутизатори Wi-Fi [1]. Згідно зі статистикою, останніми роками у світі неухильно зростає кількість витоків та обсяг скомпрометованих даних. Так, за даними досліджень Dell Technologies [2], у 2022 р. підприємства зіткнулися з великою кількістю інцидентів безпеки, ніж у попередні роки. Це призвело до збільшення часу простою, збільшення втрат даних та збільшення витрат на відновлення. Понад 1 мільйон доларів – це середня ціна втрати даних підприємства у 2022 р. На додаток до фінансових втрат і втрат репутації, що виникають внаслідок витоку даних, слід також враховувати, що сьогодні організації працюють в умовах дедалі більш суворого та швидко змінного нормативно-правового поля, в документах якого передбачено обов'язкове виконання відповідних приписів. Лише у Сполучених Штатах Америки діє понад 20 національних законів про конфіденційність та безпеку даних, а також додаткові закони, прийняті на рівні штатів [1]. Загальний регламент захисту персональних даних (General Data Protection Regulation – GDPR) [3] Європейського Союзу (ЄС) діє у державах-членах ЄС. Подібні правила та закони діють в Україні, Японії, Австралії, Новій Зеландії, Індії, Південній Кореї, Чилі, Бразилії. Таким чином, сьогодні можна однозначно говорити про те, що має місце проблема забезпечення безпеки РІС. Ось деякі її ключові аспекти:

– *безпека мережі*: РІС значною мірою покладаються на мережі для зв’язку між вузлами, тому безпека мережі є життєво важливою для захисту від таких загроз, як прослуховування, перехоплення даних і атаки типу «людина посередині»;

– *контроль доступу*: реалізація належного контролю доступу гарантує, що лише авторизовані користувачі або вузли можуть отримати доступ до певних ресурсів або виконувати певні дії в розподіленій системі;

– *цілісність даних*: мають бути вжиті заходи для виявлення та запобігання підробці даних під час передачі або зберігання;

– *шифрування даних*: чутливі конфіденційні дані, що передаються між вузлами, або зберігаються в розподіленій системі, повинні бути зашифровані, тобто перетворені на нечитану форму з використанням криптографічних примітивів;

– *пом'якшення наслідків розподіленої відмови в обслуговуванні* (distributed denial of service – DDoS): РІС чутливі до DDoS-атак, які можуть порушити їхню роботу, тому необхідні ефективні стратегії запобігання таким атакам;

– *безпечне кодування*: розробники повинні дотримуватися методів безпечного кодування, щоб звести до мінімуму вразливості програмного забезпечення, що працює в розподілених системах;

– *моніторинг та аудит*: безперервний моніторинг дій і подій безпеки в розподіленій системі та ведення журналів аудиту мають вирішальне значення для виявлення інцидентів безпеки;

– *реагування на інциденти*: наявність чітко визначеного плану реагування на інциденти має вирішальне значення для швидкого виявлення інцидентів безпеки, реагування на них і відновлення після них;

– *хмарна безпека*: якщо розподілена система використовує хмарні служби, забезпечення безпеки даних і програм у хмарі має важливе значення;

– *безпека з нульовою довірою* (zero trust security): впровадження моделі безпеки з нульовою довірою, де довіра ніколи не передбачається, може підвищити безпеку в розподілених системах;

– *відповідність нормативним та законодавчим актам*: у багатьох галузях існують спеціальні нормативні вимоги щодо безпеки та конфіденційності даних; дотримання відповідних законів і нормативних актів має вирішальне значення;

– *навчання з питань безпеки*: навчання користувачів і адміністраторів найкращим практикам безпеки та ризикам, пов’язаним із розподіленими системами, може допомогти запобігти порушенням безпеки.

Як видно з викладеного вище, без комплексного підходу до інформаційної безпеки, що поєднує в собі використання законодавчих, організаційних заходів, програмно-технічних засобів, політики та обізнаність користувачів, не обійтися. Регулярні оцінки безпеки, сканування вразливостей та тестування на проникнення можуть допомогти виявити та усунути слабкі місця у розподілених системах. Крім того, організаціям слід бути в курсі загроз, що виникають, і кращих практик забезпечення безпеки, щоб відповідним чином адаптувати свої заходи. У ситуації, що склалася, беручи до уваги сучасний стан розвитку технологій розподілених інформаційних систем, його швидкоплинний характер, науково-практичні досягнення в галузі інформаційної безпеки, кваліфікацію зловмисників, які постійно вдосконалюють можливості шкідливого впливу, положення та рекомендації різних нормативно-правових актів у багатьох випадках фахівцям з інформаційних систем, щоб забезпечити надійне безпечне функціонування останніх, потрібні відповідні знання з питань забезпечення безпеки. Тобто знання актуальних сучасних методів, прийомів та засобів забезпечення безпеки. Ця стаття якраз і націлена на надання таких знань. У ній у стислому викладі представлено достатньо широке коло питань, пов'язаних із безпекою розподілених інформаційних систем.

1. Ключові поняття інформаційної безпеки

Термін «*безпека*» є широко поширеним. Він використовується у політиці, військовій сфері, науці, техніці, освіті тощо. При цьому, як показує аналіз [4], його трактування, а отже і розуміння, буває різним. Причому розбіжності у трактуванні можуть бути дуже значними. Щоб виключити неоднозначність, представимо кілька визначень, даних у різних авторитетних джерелах, що дозволяють зрозуміти його суть з погляду аспектів, що розглядаються далі. В онлайн-словнику [5] дається достатньо загальне визначення безпеки (security) як якості або стану безпечного буття, такого як свобода від небезпеки, страху або тривоги, з іншого боку це щось, що захищає. Подібно цей термін визначають і автори [6]: «безпека – це стан безпечного буття та відсутності небезпеки чи шкоди. Крім того, це дії, вжиті для забезпечення безпеки когось чи чогось». У роботі [6] також формулюється, що «*безпека* – це захист», зазначаючи при цьому, що «захист від зловмисників – тих, хто навмисно чи іншим чином може завдати шкоди, є кінцевою метою безпеки». У роботі [7] безпека визначається з погляду системного підходу, тобто як системна властивість. При цьому констатується, що безпека – це набагато більше, ніж набір функцій та механізмів. А *безпека інформаційних технологій* – це характеристика системи, а також набір механізмів, які логічно і фізично охоплюють систему. Найбільш близькими до проблематики, що розглядається далі, є визначення *безпеки* (*security*), наведені в документах NIST [8, 9], що фактично повторюють суть визначення

інформаційної безпеки (*information security*), наведеного в стандарті ISO/IEC 27000 [10] та в іншому пізнішому документі NIST [11]. Так, відповідно до ISO/IEC 27000 [10] *інформаційна безпека* визначається як збереження конфіденційності, цілісності та доступності інформації. При цьому в стандарті ISO/IEC 27000 звертається увага на те, що крім зазначених вище можуть мати значення й інші властивості інформації, такі як: *автентичність* (*authenticity*); *підзвітність* (*accountability*); *невідмовність* (*non-repudiation*) та *надійність* (*reliability*).

Забезпечення безпеки розподілених інформаційних систем є надзвичайно важливим і водночас складним завданням. Так як навіть єдине слабке місце в системі може призвести до порушення безпеки всієї системи, і зробити заходи захисту її активів (актив – сутність, що імовірно представляє цінність для власника об'єкта оцінки [12]), що використовуються, марними. Основними активами будь-якої інформаційної системи є її обладнання, програмне забезпечення та дані. Безпеку у розподілених системах можна грубо розділити на дві частини [13, 14]. Одна частина стосується зв'язку між користувачами або процесами, які, можливо, знаходяться на різних машинах. Основним механізмом захисту передачі між вузлами мережі є організація безпечного каналу, що забезпечує автентифікацію взаємодіючих сторін, конфіденційність і цілісність даних, переданих повідомлень. Інша частина стосується авторизації, яка полягає у забезпеченні отримання процесом лише тих прав доступу до ресурсів у розподіленій системі, на яку він має право. Хоча для коректності, доцільно зазначити, що є ще деякі інші складові заходів безпеки, наприклад, резервне копіювання, моніторинг системи та інші. Говорячи про безпеку в комп'ютерній системі загалом, можна помітити, що вона тісно пов'язана з поняттям надійності. Неформально надійна комп'ютерна система – це система, якій ми по праву довіряємо у наданні послуг [14 – 16]. Надійність – це властивість системи, яка поєднує такі атрибути, як [15, 16]: доступність (availability), достовірність / безвідмовність (reliability), функціональну безпеку (safety) та ремонтопридатність (maintainability). Однак, якщо ми хочемо повною мірою довіряти комп'ютерній системі, слід враховувати й такі атрибути надійності, як конфіденційність і цілісність.

Конфіденційність (*confidentiality*) – це концепція заходів, що використовуються для забезпечення захисту секретності даних, об'єктів чи ресурсів (або інакше – властивість, що полягає в тому, що інформація не надається або не розкривається неавторизованим особам, організаціям або процесам). Метою захисту конфіденційності є запобігання або мінімізація несанкціонованого доступу до даних [17]. Конфіденційність відноситься до якості комп'ютерної системи, згідно з якою її інформація не надається або не розкривається неавторизованим особам, організаціям або процесам, вона розкривається лише уповноваженим сторонам. Широкий спектр заходів безпеки, що забезпечує захист конфіденційності, включає насамперед контроль доступу, шифрування та стеганографію.

Говорячи про конфіденційність, слід також відзначити деякі, пов'язані з нею, поняття та аспекти, а саме [17, 18]:

– *чутливість* (*sensitivity*) – відноситься до якості інформації, розкриття якої може завдати шкоди (harm) або збитків (damage). Збереження / підтримка (*maintaining*) конфіденційності чутливої інформації допомагає запобігти шкоді чи збитку;

– *обачність* / *обережність* (*discretion*) – це акт рішення, при якому оператор може впливати на розкриття інформації чи контролювати її, щоб мінімізувати шкоду чи збиток;

– *критичність* (*criticality*). Рівень, до якого інформація є критично важливою, є мірою її критичності. Чим вищий рівень критичності, тим більша ймовірність збереження конфіденційності інформації;

– *приховування* (*concealment*) – це дія з *приховування* / *хования* (*hiding*) або запобігання розкриття інформації. Часто приховування розглядається як *засіб укриття* (*cover*), *заплутування / обфускації даних* (*obfuscation*) чи *відволікання* (*distraction*). Поняття приховування пов'язане з безпекою через *безвісність* (*obscurity*), тобто зі спробою отримати захист за допомогою приховування (*hiding*), *мовчання* (відсутність відомостей – *silence*) чи *секретності* / *таємності* (*secrecy*). Хоча безпека через безвісність (неясність, невідомість) зазвичай

не вважається дійсною (valid) мірою безпеки, у деяких випадках вона все ж таки може мати значення;

– *приватність* (*privacy*) – означає збереження конфіденційності інформації, яка дозволяє встановити особистість, або яка може заподіяти будь-кому *шкоду (збиток, лихо, зло – harm*), *збентеження* / *незручності* (*embarrassment*) або *немилість* / *ганьбу* (*disgrace*) у разі її розкриття;

– *усамітнення* (*seclusion*) – має на увазі зберігання чогось у віддаленому місці. Це місце також може забезпечувати суворий контроль доступу;

– *ізоляція* (*isolation*) – це дія щодо збереження чогось окремо від інших. Ізоляція може використовуватися для запобігання змішування інформації або її розкриття.

Цілісність (*integrity*) – це концепція захисту *достовірності* / *надійності* (*reliability*) та *правильності* (*correctness*) даних. Захист цілісності запобігає несанкціонованій зміні даних. Це гарантує, що дані залишаються *правильними* (*correct*), *незмінними* (*unaltered*) та *збереженими* (*preserved*). Правильно реалізований захист цілісності надає засоби для авторизованих змін, одночасно захищаючи від намічених та зловмисних несанкціонованих дій (таких як віруси та вторгнення), а також від помилок, допущених авторизованими користувачами (таких як *помилки* (*mistakes*) чи *недогляди* / *упущення* (*oversights*)) [17]. Іншими словами, неправильні зміни в захищеній комп'ютерній системі повинні виявлятися та виправлятися. Цілісність можна розглядати з трьох точок зору: 1) запобігання (*preventing*) внесенню змін неавторизованими суб'єктами; 2) запобігання внесенню авторизованими суб'єктами несанкціонованих змін, наприклад, помилок; 3) підтримка (*maintaining*) внутрішньої та зовнішньої узгодженості об'єктів, щоб їх дані були правильним і істинним відображенням реального світу, а будь-які відносини / зв'язки (*relationship*) з будь-яким дочірнім, рівним (*peer*) або батьківським об'єктом були дійсними (valid), узгодженими (consistent) та такими, що піддаються перевірці (*verifiable*).

Для забезпечення цілісності в системі повинні бути передбачені елементи керування для обмеження доступу до даних, об'єктів та ресурсів. При цьому слід зазначити, що є інші поняття, аспекти, пов'язані з цілісністю, зокрема: *точність* (*accuracy*) – бути правильним / коректним (*correct*) та чітким (*precise*); *правдивість* (*truthfulness*) – бути справжнім відображенням дійсності; *справжність* (*автентичність – authenticity*) – бути справжнім (*authentic*) або непідробним (*genuine*); *дійсність* (*валідність – validity*) – бути фактично або логічно обґрунтованим; *невідмовність* / *неспростовність* (*non-repudiation*) – неможливість відмовитися від авторства, нездатність заперечувати здійснення дії; *підзвітність* (*accountability*) – бути відповідальним або тим, що має зобов'язання за дії та результати; *відповідальність* (*responsibility*) – бути відповідальним (мати обов'язки) або мати контроль над чимось чи кимось; *комплектність* (*completeness*) – наявність всіх необхідних компонентів або частин; *повнота* (всеосяжність – *comprehensiveness*) – бути повним за обсягом; повне включення всіх потрібних елементів.

Конфіденційність та цілісність залежать один від одного. Без цілісності об'єкта (тобто неможливості зміни об'єкта без дозволу) конфіденційність не може бути підтримана.

Третій принцип Тріади CIA (*Confidentiality, Integrity, Availability*) – це доступність, що означає, що авторизованим суб'єктам надається своєчасний та безперервний доступ до об'єктів. Часто засоби управління захистом доступності підтримують достатню пропускну здатність та своєчасність обробки, якщо це необхідно організації або спричинено ситуацією. Якщо механізм безпеки забезпечує доступність, він забезпечує високий рівень гарантії того, що дані, об'єкти та ресурси доступні авторизованим суб'єктам. Доступність передбачає ефективний безперервний доступ до об'єктів, у тому числі в умовах DoS атак. Доступність також передбачає, що підтримуюча інфраструктура, включаючи мережеві служби, засоби зв'язку та механізми контролю доступу, функціонує і дозволяє авторизованим користувачам отримувати авторизований доступ. Для підтримки доступності в системі повинні бути передбачені елементи керування для забезпечення авторизованого доступу та прийнятного рівня продук-

тивності, забезпечення необхідної надмірності, підтримання надійних резервних копій та запобігання втраті або знищенню даних. При цьому доступність залежить як від цілісності, так і конфіденційності. Без цілісності та конфіденційності доступність не може бути підтримана.

З іншого боку, безпеку інформаційних систем слід розглядати з точки зору необхідності захисту від різних загроз безпеці наданих ними служб (сервісів, послуг) та даних. Насамперед слід враховувати такі існуючі типи загроз безпеки (security threats), як [19, 20]: перехоплення (*interception*); переривання (*interruption*); модифікація (*modification*); фабрикація (*fabrication*).

Концепція *перехоплення* відноситься до ситуації, коли неавторизована сторона отримала доступ до служби або даних. Типовим прикладом перехоплення є випадок, коли зв'язок між двома сторонами був підслуханий кимось іншим. Перехоплення також відбувається, коли дані незаконно копіюються, наприклад, після злому облікового запису (account) користувача або каталогу суб'єкта у файловій системі. У загальному сенсі *переривання* належить до ситуації, у якій служба або дані стають недоступними, непридатними, знищеними тощо. У цьому сенсі атаки типу «відмова в обслуговуванні» (DoS), за допомогою яких хтось зловмисно намагається зробити службу недоступною для інших сторін, є загрозою безпеці, яка класифікується як переривання. *Модифікації* включають несанкціоновану зміну даних або підробку служби, щоб вона більше не дотримувалась своїх початкових специфікацій. Приклади модифікацій: перехоплення та подальша зміна даних, фальсифікація записів у базі даних, зміна програми тощо. Під *фабрикацією* розуміється ситуація, в якій генеруються додаткові дані або дії, які зазвичай не існують. Наприклад, зловмисник може спробувати додати запис до файлу пароля або бази даних, створити фальшиві профілі та опублікувати хибну інформацію в соціальних мережах, щоб вплинути на громадську думку, створити фальшиві вузли або надсилати фальшиві повідомлення. Слід звернути увагу, що модифікація і фабрикація можуть розглядатися як форма фальсифікації (*falsification*) даних.

2. Основні підходи щодо забезпечення безпеки розподілених систем

На даний момент фахівцями в галузі безпеки та розподілених інформаційних систем напрацьовано певні підходи та концепції у цьому напрямку. Як правило, спочатку в них рекомендується визначити вимоги до безпеки системи, тобто описати політики безпеки. Термін «політика комп'ютерної безпеки» (*computer security policy*) має кілька значень [21]. З одного боку, політика – це директиви вищого керівництва щодо створення програми комп'ютерної безпеки, встановлення її цілей та розподілу обов'язків. З іншого боку, термін політика використовується для позначення певних правил безпеки для певних систем. Крім того, політика може відноситися до зовсім інших питань, таких як конкретні управлінські рішення. Далі більшою мірою використовуватимуться аспекти поняття «політика безпеки» як політики інформаційної безпеки (information security policy), маючи на увазі під цим терміном сукупність законів, правил, методів, рекомендацій, що вказують порядок управління, захисту та розподілу інформації. Тобто в даному контексті політика безпеки (security policy) точно описує, які дії можна виконувати сутностям (entities) у системі, а які заборонено. Під сутностями розуміються користувачі, служби, дані, машини тощо. Після того, як політики безпеки будуть встановлені, стає можливим зосередитися на механізмах безпеки, за допомогою яких можна застосовувати певну політику. Механізм безпеки (security mechanism) – це пристрій або функція, призначена для надання однієї або декількох послуг безпеки, які зазвичай оцінюються з точки зору надійності обслуговування та гарантованості проекту [22]. Важливими механізмами безпеки вважаються [13, 14, 23]:

1. *Шифрування* (encryption). Шифрування має фундаментальне значення для комп'ютерної безпеки. Воно перетворює дані на те, що зловмисник не може зрозуміти. У контексті криптографії шифрування – це механізм, що забезпечує конфіденційність даних. Крім того, використовуючи різні криптографічні примітиви, можна забезпечити перевірку цілісності

даних (чи були дані змінені або ні). Маршрутизатор, сервер, кінцева система або виділений пристрій можуть виступати як пристрій шифрування / розшифрування. Дані, що зашифровуються, називаються зашифрованими даними (ciphered or encrypted data). Не зашифровані дані називаються простим або відкритим текстом (plain text or clear text).

2. *Авторизація* (authorization). Авторизація – це надання прав (привілеїв) конкретному учаснику процесу інформаційного обміну (автентифікованого або анонімного), що дозволяють їх власнику (людині, програмі або процесу) мати законний доступ до системи або до її об'єктів. Експерти з безпеки рекомендують використовувати принцип найменших привілеїв. Цей принцип ґрунтується на ідеї, що кожному користувачеві мають бути надані лише мінімально необхідні права для виконання певного завдання. Засоби авторизації користувачів можуть бути реалізовані за допомогою програмного коду та керувати не лише наданими користувачам правами доступу до системи чи об'єктів, але й набором операцій, які користувачі можуть виконувати з кожним об'єктом, який йому доступний.

3. *Автентифікація* (authentication). Термін «автентифікація» зазвичай стосується автентифікації користувачів, але також може стосуватися автентифікації пристроїв або програмних процесів. Тобто автентифікація може використовуватись для перевірки заявленої особи користувача, клієнта, сервера, хоста / вузла чи іншого об'єкта / сутності. Автентифікація згідно [24, 25] – це перевірка особи користувача, процесу або пристрою, часто як необхідна умова для дозволу доступу до ресурсів в інформаційній системі. У випадку з клієнтами основна передумова полягає в тому, що перед тим, як служба починає виконувати будь-яку роботу від імені клієнта, служба повинна впевнитись в особистості клієнта (якщо служба недоступна для всіх). Перевірка справжності (автентифікація) може проводитись різними методами та засобами, що використовують однофакторну та багатофакторну автентифікацію. Сьогодні широке застосування знайшли три типи / категорії факторів, що дозволяють пов'язати людину із встановленими повноваженнями [26 – 29]:

– фактори, що ґрунтуються на знанні (knowledge factors) – інформації, яка повинна зберігатися в секреті і яку може знати лише певний клієнт / користувач, наприклад, пароль, графічний пароль, парольна фраза (користувач «знає»);

– фактори, засновані на володінні (ownership / possession factors) – щось, що є у користувача, наприклад смарт-карти, смартфони, токени безпеки (користувач «має»);

– фактори невід'ємності / властивості (inherence factors) або біометричні фактори (biometric factor) – фізіологічні ознаки, властиві конкретним особам – біометричні дані або зразок поведінки, наприклад швидкість набору тексту, динаміка натискання клавіш, руху миші, сенсорні жести на сенсорних екранах і т. д. (користувач «є» або хто ви).

Хоча в літературі пропонуються й інші фактори (такі як автентифікація з використанням облікових даних людини в соціальних мережах та автентифікація на основі розташування), три перелічені вище категорії факторів є найбільш використовуваними [27]. Методи автентифікації, що стосуються різних факторів, можна комбінувати для підвищення безпеки, така автентифікація відома як багатофакторна [28]. Деякими прикладами багатофакторної автентифікації є комбінація факторів знання та володіння, комбінація факторів знання та приналежності, комбінація факторів володіння та приналежності, а також поєднання всіх трьох відомих факторів [27].

4. *Аудит* (auditing, audit). «Аудит» (audit) та «аудит» (auditing; коректніше, напевно, «аудиторська діяльність») – це досить близькі терміни, пов'язані з процесом перевірки та оцінки систем, процедур, даних та інших аспектів для забезпечення їх точності, надійності та дотримання стандартів. Вони можуть використовуватись у різних контекстах. Під терміном «audit», як правило, мають на увазі процес перевірки та оцінки чогось з метою визначення його правильності, відповідності стандартам або дотримання вимог, а під терміном «auditing» – дія або процес проведення аудиту («audit»). Ці терміни часто використовуються як взаємозамінні, але важливо розуміти різницю між ними. Процес аудиту може бути застосований у різних галузях та мати різні цілі. Аудит (audit) у сфері безпеки (security) –

це незалежний аналіз та перевірка записів та дій для оцінки адекватності системного контролю, забезпечення відповідності встановленим політикам та операційним процедурам, а також рекомендації необхідних змін у засобах контролю, політиках чи процедурах [22]. У контексті інформаційних технологій аудит (auditing) має вирішальне значення для оцінки безпеки систем і даних. IT-аудитори оцінюють засоби контролю / керування, вразливості та потенційні загрози для захисту від витоку даних та кібератак. У контексті розподілених інформаційних систем терміни «audit» та «auditing» мають аналогічне значення, що і в інших областях, але застосовуються спеціально до аудиту інформаційних систем. Аудит (audit) у розподілених інформаційних системах – це процес систематичної перевірки та оцінки компонентів та процесів розподіленої інформаційної системи з метою визначення точності, безпеки, надійності та дотримання стандартів та політик безпеки (приклади: аудит системи керування доступом, аудит мережевих протоколів, аудит захисту від вторгнення, аудит безпеки застосунків і т. д.). Аудит (auditing) у розподілених інформаційних системах – це виконання аудиту (audit). Auditing включає проведення перевірок, аналізу даних, реєстрації подій та виявлення аномалій у розподіленому інформаційному середовищі (приклади: діяльність аудиторів інформаційної безпеки, які аналізують журнали подій та проводять перевірки на рівні мережі та застосунків). Auditing сприяє забезпеченню підзвітності користувачів, запобіганню неналежних дій користувачів та розслідуванню підозрілої активності [30]. Аудит (auditing) розподілених інформаційних систем допомагає організаціям підтримувати надійність, безпеку та відповідність вимогам своєї IT-інфраструктури. Це критично важлива практика для захисту конфіденційних даних, запобігання вразливості та забезпечення правильного функціонування складних взаємопов'язаних систем. Інструменти аудиту (auditing) використовуються для відстеження того, які клієнти отримали доступ до чого і яким чином. Хоча аудит (auditing) насправді не забезпечує жодного захисту від загроз безпеки, журнали аудиту можуть бути надзвичайно корисними для аналізу злому системи безпеки та подальшого вжиття заходів проти зловмисників. З цієї причини зловмисники, як правило, прагнуть не залишати жодних слідів, які могли б зрештою призвести до розкриття їхньої особистості. У цьому сенсі реєстрація доступу в журналах робить атаку більш ризикованою.

Таким чином, очевидно, що безпека розподілених систем багато в чому залежатиме від застосовуваних у ній механізмів, що реалізують відповідні різні правила захисту (політики безпеки). При цьому, реалізуючи відповідні служби захисту, слід враховувати низку важливих аспектів [14, 31]: а) на чому, на кому необхідно сконцентруватися при розробці механізмів захисту: на даних, операціях або користувачах (загалом це часто називають фокусом контролю (focus of control) або об'єктом контролю / керування); б) на якому рівні комп'ютерної системи слід розміщувати механізми безпеки (як правило, комп'ютерну систему можна представити у вигляді деякої багаторівневої моделі, а отже, і організація механізмів безпеки також має бути багаторівневою); в) чому віддається перевага простоті (simplicity) та високому ступеню впевненості (higher assurance) або багатофункціональному середовищу безпеки. Для досягнення високого ступеня впевненості система безпеки має бути досліджена в деталях і якомога вичерпніше. Отже, існує компроміс між складністю та впевненістю. Чим вище рівень впевненості, до якого ви прагнете, тим простіше має бути ваша система. Як наслідок, можна помітити, що багатофункціональні системи безпеки та високий рівень упевненості не легко поєднуються один з одним; г) на кого доцільно покласти завдання щодо визначення та забезпечення безпеки: на центральний об'єкт або на окремі компоненти системи.

Щоб відповісти на ці питання, необхідно розібратися в ряді ключових концепцій та підходів, які допомагають захистити дані та ресурси у розподілених системах. Ось деякі основні з них, що є основою для розробки стратегій та заходів безпеки в розподілених системах і допомагають зменшити ризики та запобігти загрозам безпеці:

– *Об'єкт контролю*. Передбачає використання залежно від специфіки системи та вимог до неї одного з підходів, пов'язаних із концентрацією (фокусуванням) на таких аспектах як: захист безпосередньо асоційованих із застосунком даних; контроль доступу (точна вказівка

того, хто і як може використовувати операції доступу до даних або ресурсів); користувач (вжити заходів, щоб доступ до застосунку мали лише певні користувачі, незалежно від операцій, які вони планують виконувати).

– *Багаторівнева організація механізмів безпеки*. Цей підхід передбачає створення декількох рівнів захисту в системі, кожен з яких виконує конкретні функції, і на кожному з них можуть бути реалізовані відповідні механізми, щоб забезпечити комплексний захист даних та мережевих ресурсів.

– *Простота механізмів захисту.* Використовувати кілька простих механізмів, які легко зрозуміти та яким можна довіряти, завжди є найкращим вибором.

– *Використання криптографічних методів.* Використання криптографічних методів для захисту конфіденційності та цілісності даних під час передачі та зберігання. Наприклад, на рівні передачі зазвичай використовуються протоколи безпечного зв'язку, такі як SSL/TLS, на рівні зберігання – шифрування (у тому числі так зване прозоре шифрування – Transparent Data Encryption – TDE [32]).

– *Організація безпечних каналів* (*secure channels*). Безпечний канал у розподілених інформаційних системах є захищеним засобом зв'язку, який забезпечує конфіденційність і цілісність інформації, що передається між різними компонентами або вузлами РІС, а також здійснює перевірку справжності учасників інформаційного обміну та їх прав доступу до певних ресурсів системи. Тобто безпечний канал забезпечує захист відправників та одержувачів повідомлень від перехоплення (повідомлення не можуть бути підслухані зловмисниками), модифікації та фальсифікації / підробки (здійснюється за допомогою протоколів взаємної автентифікації та цілісності повідомлень). Безпечний канал повинен надавати захист від різних видів атак, таких як: людина посередині (man-in-the-middle), повторне відтворення (replay attack), відмова в обслуговуванні (DoS), перехоплення сеансу (session hijacking), фішингові атаки та інші. Зазвичай не потрібно вводити захист від переривання зв'язку. У РІС використовуються різні технології та протоколи для створення безпечних каналів. Важливо вибирати відповідні з них залежно від конкретних потреб і вимог системи. Нижче наведено деякі загальні технології та протоколи, які використовуються для створення безпечних каналів у розподілених системах: 1) TLS (Transport Layer Security) / SSL (Secure Sockets Layer) протоколи – забезпечують безпечний зв'язок через Інтернет та інші мережі; зазвичай використовуються для захисту веб-трафіку (HTTPS) та електронної пошти (SMTP з TLS/SSL); 2) віртуальні приватні мережі (VPN – Virtual Private Network) – створюють безпечне зашифроване з'єднання через загальнодоступну мережу, забезпечуючи конфіденційність та безпеку даних; 3) IPsec (Internet Protocol Security) – набір протоколів, що використовуються захисту зв'язку лише на рівні IP (Internet Protocol), часто як і в VPN; 4) SSH (Secure Shell) – криптографічний мережевий протокол для безпечного віддаленого доступу до систем та передачі даних; 5) Kerberos – мережевий протокол автентифікації, який використовує квитки (tickets) та криптографію з симетричним ключем (вимагає наявності довіреної третьої сторони – центру сертифікації / розподілу ключів) для забезпечення безпечної автентифікації в незахищеній мережі.

– *Контроль доступу* (access control). Контроль доступу відповідно до визначень [22, 24] – процес задоволення чи відхилення конкретних запитів на: 1) отримання та використання інформації та пов'язаних з нею послуг з обробки інформації; 2) вхід на певні фізичні об'єкти (наприклад, федеральні будівлі, військові об'єкти, прикордонні переходи). У контексті статті нас насамперед цікавитиме перша частина цього визначення, тобто, рішення про дозвіл або заборону суб'єкту доступу до об'єктів системи (мережі, даним, застосунку, сервісу тощо) [22].

– *Безпечне іменування* (secure naming). Основна ідея технології безпечного іменування полягає в тому, щоб вбудувати в самі імена ресурсів (наприклад, доменні імена, імена файлів, URL тощо) інформацію про безпеку та справжність цих ресурсів. Це робиться з метою покращення безпеки та забезпечення автентифікації ресурсів без необхідності покладатися

на зовнішні джерела або центральні установи. Ключовими концепціями та особливостями технології безпечного іменування є: а) самодостатність (ім'я ресурсу містить у собі інформацію про свою справжність або цифровий підпис; немає необхідності звертатися до центральних установ або сертифікаційних органів для перевірки справжності ресурсу); б) захист від підробки (від фальсифікації та атак, пов'язаних із зміною імен ресурсів); в) криптографічна безпека (дані, включені в ім'я ресурсу (наприклад, цифровий підпис), забезпечують криптографічний захист, який може бути перевірений клієнтським пристроєм або користувачем); г) складність організації атак (технологія ускладнює завдання зловмисникам, які намагаються атакувати ресурси, оскільки вони повинні підробити або обійти криптографічний захист, вбудований в імена ресурсів).

– *Управління безпекою.* За останні кілька десятиліть дисципліна управління IT-безпекою значно змінилася. Це сталося у відповідь на швидке зростання мережевих комп'ютерних систем та залежність від них, а також пов'язане з цим зростання ризиків для цих систем. Останнім часом було опубліковано низку національних та міжнародних стандартів (серія стандартів ISO 27000, NIST, у тому числі NIST SP 800-18 Rev.1, 2006 р., NIST SP 800-30 Rev.1, 2023 р., NIST SP 800-53, Rev. 5, 2020 р.). Вони є консенсусом щодо передової практики в цій галузі [33]. Управління безпекою загалом – це широка сфера управління, пов'язана з управлінням активами, фізичною безпекою та функціями безпеки людських ресурсів; це процес планування, організації, впровадження, контролю та безперервного поліпшення системи безпеки в організації. Управління безпекою означає відповідальність та дії, необхідні для управління середовищем безпеки, включаючи політики та механізми безпеки [34]. Управління безпекою є ключовим елементом будь-якої організації, особливо у контексті сучасних загроз, пов'язаних із кібербезпекою та тероризмом. Воно дозволяє мінімізувати ризики та забезпечити захист важливих ресурсів та інтересів організації. Управління інформаційною безпекою є важливим елементом у забезпеченні безпеки даних в організаціях, компаніях та установах, тому що воно допомагає: 1) захистити конфіденційність даних (таких як персональні дані клієнтів, банківські дані, інтелектуальну власність тощо); 2) запобігти кібератакам, зламам та іншим загрозам, які можуть призвести до витоку конфіденційних даних, порушення цілісності, доступності даних, завдаючи серйозної шкоди організації; 3) організаціям забезпечувати безпеку даних своїх клієнтів відповідно до вимог законів, нормативних документів та різних правил; 4) зберегти репутацію організації, шляхом запобігання порушенням безпеки даних, які можуть завдати їй серйозної шкоди; 5) підвищити ефективність функціонування організації, зменшити витрати на відновлення після інцидентів безпеки; 6) запобігти витоку інформації, незаконному використанню даних, шкідливим діям з боку співробітників тощо. Якщо говорити про *управління безпекою в розподілених системах*, то це комплекс заходів та процесів, спрямованих на безпеку розподілених систем, включаючи захист від несанкціонованого доступу, шкідливих програм, витоку інформації та інших загроз. Такий комплекс заходів та процесів включає різні аспекти, такі як: автентифікацію та авторизацію користувачів і пристроїв в системі; керування доступом до ресурсів та даних у розподіленій системі; моніторинг подій та виявлення загроз безпеці; управління ризиками; розроблення політик та процедур, що визначають правила та процеси, необхідні для забезпечення безпеки; реагування на загрози, що виникли, і їх запобігання; управління механізмами, що забезпечують конфіденційність, цілісність та доступність даних, що зберігаються в системі та передаються між пристроями в розподіленій системі; забезпечення відповідності розподіленої системи відповідним стандартам безпеки, нормам та галузевим вимогам, таким як GDPR, HIPAA або PCI DSS; навчання користувачів та адміністраторів системи передовим методам забезпечення безпеки та підвищення поінформованості про потенційні загрози та атаки соціальної інженерії та деякі інші.

– *Безпека з нульовою довірою* (*zero trust security*). *Нульова довіра* (zero trust – ZT) є набором концепцій та ідей, призначених для мінімізації невизначеності при забезпеченні правильних рішень про доступ з найменшими привілеями для кожного запиту до інформа-

ційних систем та служб в умовах, коли мережа вважається скомпрометованою [35]. *Архітектура нульової довіри* (zero trust architecture – ZTA) – це план кібербезпеки підприємства, який використовує концепції нульової довіри та включає взаємовідносини / взаємозв'язки компонентів, планування робочих процесів та політики доступу. Таким чином, підприємство з нульовою довірою – це мережева інфраструктура (фізична та віртуальна) та оперативні політики, які діють для підприємства як продукт плану архітектури з нульовою довірою. Принципи моделі «нульової довіри» є сучасним підходом до забезпечення інформаційної безпеки, який передбачає, що не можна довіряти жодному користувачеві, пристрою чи компоненту всередині або поза корпоративної мережі. Ця модель закликає до безперервної перевірки та автентифікації всіх суб'єктів та ресурсів, навіть тих, що знаходяться всередині мережі. Gartner [36] рекомендує організаціям впроваджувати концепцію нульової довіри, щоб насамперед покращити зниження ризиків для найважливіших активів, оскільки саме тут буде отримано найбільшу віддачу від зниження ризиків (при цьому фахівцями Gartner уточнюється, що цей підхід не вирішує всіх завдань безпеки). До 2026 р. Gartner прогнозує, що 10 % великих підприємств матимуть зрілу та вимірну програму нульової довіри [37]. У 2022 р. прибутки ринку ZTNA (zero trust network access) зросли на 80,6 % [38].

Далі розглянемо детальніше деякі з перерахованих вище ключових концепцій та підходів, які допомагають захистити дані та ресурси у розподілених системах.

2.1. Багаторівнева організація механізмів безпеки

Важливим моментом розробки безпечних систем є вирішення, скільки рівнів повинні мати механізми безпеки. Рівень у цьому контексті пов'язаний із логічною організацією системи, що складається з кількох шарів / рівнів. Якщо підійти до розгляду організації системи забезпечення безпеки в контексті рівнів еталонної моделі OSI (Open Systems Interconnection) як концептуальної основи, усвідомлюючи при цьому, що структура розподіленої системи включає окремі рівні для застосунків, проміжного програмного забезпечення, служб та ядра операційної системи, то в у цьому випадку механізми безпеки можуть бути розподілені за |рівнями таким чином:

1. Фізичний рівень. На цьому рівні можна впровадити заходи безпеки для захисту фізичної інфраструктури, такі як контроль доступу до центрів обробки даних, систем відеоспостереження та систем виявлення вторгнень у серверні приміщення.

2. Канальний рівень. На цьому рівні можуть застосовуватися механізми шифрування та автентифікації для захисту передачі даних мережевими каналами. Як приклади заходів безпеки можна навести VPN та протоколи шифрування на канальному рівні.

3. Мережевий рівень. На цьому рівні можна застосувати такі рішення, як використання міжмережевого екрану, сегментацію мережі, системи виявлення та запобігання вторгненням для захисту від мережевих атак.

4. Транспортний рівень. На цьому рівні можна застосувати криптографічні протоколи SSL/TLS для шифрування даних під час передачі.

5. Сеансовий рівень. На цьому рівні можуть застосовуватися механізми керування сеансами та автентифікації на основі сеансів для забезпечення безпеки сеансів зв'язку між об'єктами мережі.

6. Рівень представлення. Механізми безпеки на рівні представлення можуть включати перетворення формату, шифрування даних для їх захисту під час представлення.

7. Прикладний рівень. На цьому рівні заходи безпеки можуть включати автентифікацію користувачів, контроль доступу та захист від шкідливих програм.

Важливо відзначити, що вибір механізмів безпеки та їх реалізація мають бути адаптовані до конкретних вимог безпеки організації та ландшафту загроз (threat landscape). Крім того, чітко визначена політика безпеки та регулярні оцінки безпеки є найважливішими компонентами комплексної стратегії безпеки. Але, з іншого боку, як визначити, чи правильно політика визначає необхідний рівень та тип безпеки вузла розподіленої системи? Як відомо [39],

безпека ґрунтується на припущеннях, специфічних для необхідного типу безпеки та середовища, в якому вона має застосовуватися. Коли хтось зрозуміє припущення, на яких засновані його політики, механізми та процедури безпеки, він дуже добре розумітиме, наскільки ефективні ці політики, механізми та процедури. І в цьому випадку важливу роль грає поняття «довіри». Суб'єкт / сутність (entity) заслуговує на довіру, якщо є достатньо достовірних доказів, що дозволяють вважати, що система відповідатиме набору заданих вимог. Довіра – це міра надійності / достовірності (trustworthiness), заснована на наданих доказах [39]. Різниця між довірою та безпекою важлива. Система або безпечна, або ні (з урахуванням різних випадковостей), але питання про те, чи вважає користувач / клієнт систему безпечною, є питанням довіри [20].

На якому рівні розміщуються або повинні розміщуватись механізми безпеки, залежить від довіри користувача до того, наскільки безпечні служби на конкретному рівні. Безпека може бути забезпечена шляхом розміщення пристроїв шифрування на кожному магістральному комутаторі. Ці пристрої автоматично шифрують і розшифровують пакети, що відправляються між вузлами, але не забезпечують безпечного обміну даними між вузлами на одному й тому самому вузлі, тобто в межах однієї локальної мережі. Якщо Користувач 1 на вузлі A надсилає повідомлення Користувачеві 2 на вузол B і переймається тим, що його повідомлення буде перехоплено, він має бути впевненим у тому, що шифрування між вузлами працює коректно. Це означає, наприклад, що він повинен бути впевнений, що системні адміністратори на обох вузлах вжили належних заходів проти несанкціонованого доступу до пристроїв (втручання у роботу пристроїв, що шифрують).

Якщо тепер припустити, що Користувач 1 не довіряє захисту трафіку між вузлами, тоді він може прийняти рішення про необхідність використання власних заходів захисту. Наприклад, використовувати TLS (Transport Layer Security) для безпечного надсилання повідомлень через TCP-з'єднання. У цьому випадку Користувач 1, довіряючи TLS, вважає, що TLS є безпечним. У розподілених системах механізми безпеки часто розміщуються лише на рівні проміжного програмного забезпечення (ПЗ). Якщо Користувач 1 не довіряє TLS, він може використовувати локальну безпечну службу виклику віддалених процедур (Remote Procedure Call, RPC). Але йому, знову ж таки, доведеться довіряти службі RPC, у тому, що ця служба обіцяє, наприклад, запобігання витоку інформації або належну автентифікацію клієнтів та серверів. Хоча слід пам'ятати, що службам безпеки, розміщеним на рівні проміжного ПЗ розподіленої системи, можна довіряти тільки якщо служби, на які вони покладаються, є дійсно безпечними. Наприклад, якщо захищена служба RPC частково реалізована за допомогою TLS, то довіра до служби RPC залежить від того, наскільки довіряють TLS. Якщо TLS не є довіреною службою, то не може бути довіри і до безпеки служби RPC. Залежність між службами щодо довіри призводять до поняття довірена обчислювальна база (Trusted Computing Base, TCB). TCB – це набір всіх механізмів безпеки комп'ютерної системи (у тому числі й розподіленої), які необхідні для дотримання політики безпеки і яким слід довіряти. Тому, чим менше TCB (менше механізмів, які є критичними при їх компрометації, яка з великою ймовірністю поставить під загрозу безпеку системи в цілому), тим краще для безпеки (менше можливостей для атак). Якщо розподілена система побудована як проміжне ПЗ у існуючій мережній операційній системі, її безпека може залежати від безпеки базових локальних операційних систем. Тобто TCB в розподіленій системі може включати локальні операційні системи на різних вузлах.

2.2. Простота як переважний принцип проектування механізмів захисту

Ще одна важлива проблема проектування, пов'язана з ухваленням рішенням про те, на якому рівні розміщувати механізми безпеки, полягає у простоті. Проектування захищеної комп'ютерної системи вважається складним завданням, але якщо розробник системи зможе використовувати кілька простих механізмів, які легко зрозуміти і яким довіряють, це буде ефективніше. Однак слід враховувати, що для реалізації політик безпеки не завжди можливе

використання лише простих механізмів. Звернемося ще раз до випадку, коли Користувач 1 хоче надіслати повідомлення Користувачеві 2. Шифрування на канальному рівні – це простий та зрозумілий механізм захисту від перехоплення трафіку повідомлень між вузлами. Однак потрібно набагато більше, якщо Користувач 1 хоче бути впевненим, що тільки Користувач 2 отримає повідомлення. А саме потрібні послуги автентифікації на рівні користувача. При цьому Користувачеві 1, можливо, знадобляться знання про принципи роботи цих сервісів, щоб довіряти їм. Тому для автентифікації на рівні користувача може знадобитися хоча б уявлення про криптографічні ключі, електроні цифрові сертифікати, незважаючи на той факт, що багато служб безпеки автоматизовані та приховані від користувачів. В інших випадках сам застосунок є за своєю суттю складним, а впровадження безпеки додатково ускладнює його. Прикладом таких застосунків є застосунки, що включають складні протоколи захисту, зокрема, цифрові платіжні системи. Ідеальний список вимог для онлайн-платежів в електронній комерції може виглядати приблизно так [40]:

1. Конфіденційність. Механізм платежів повинен забезпечувати додаткові рівні конфіденційності, дозволяючи розкривати деталі транзакції лише сторонам, кого визначили покупець або продавець.

2. Цілісність. Повинна підтримуватись цілісність транзакції (фальсифікація або зміна деталей транзакції має бути практично неможливою).

3. Автентифікація. Повинні надаватися методи автентифікації взаємодіючих сторін та/або автентифікації повідомлень, які використовуються для авторизації платежів, щоб запобігти шахрайству.

4. Невідмовність. Повинна забезпечуватися така властивість інформаційної безпеки як невідмовність (щоб захистити як продавця, так і покупця від неправдивих заяв).

5. Доступність. Механізм платежів повинен дозволяти покупцям та продавцям брати участь у платіжних транзакціях, коли це необхідно.

6. Реалізація. Деталі реалізації мають бути абстраговані / приховані (з метою спрощення складності), а також забезпечувати інтерфейси (які повинні розроблятися на основі передових практик) із торговими системами.

7. Інтероперабельність. Механізм платежів має бути інтероперабельним, забезпечуючи максимально широкий доступ продавцям та покупцям.

8. Простота використання. Механізм платежів має бути простим для розуміння та використання покупцем.

9. Захист. Правила та політика механізму платежів повинні забезпечувати захист покупців від несумлінних продавців чи шахраїв.

Складність цифрових платежів часто пов'язана з тим, що для здійснення платежу потрібна взаємодія кількох дійових осіб. У цих випадках важливо, щоб базові механізми, що використовуються для реалізації протоколів (наприклад, таких як SSL/TLS – використовуються під час онлайн платежів через інтернет (електронна комерція через веб-браузер та SSL/TSL) та 3D Secure – ці протоколи додають додатковий рівень автентифікації під час онлайн платежів, зазвичай через введення пароля або одноразового коду), були відносно простими та зрозумілими. Простота сприятиме довірі користувачів, які працюють із застосунком, і що більш важливо, зможе переконати розробників у відсутності «дірок» у системі захисту.

2.3. Використання для захисту криптографічних методів

У захисті розподілених систем особливо важливу роль відіграє криптографія (галузь знань, що втілює в собі принципи, засоби та методи перетворення даних з метою приховування їх семантичного змісту, запобігання їх несанкціонованому використанню або запобігання їх невиявленій модифікації [22, 41]). Справді, створення захищеної системи неможливе без застосування криптографічних методів, що надають у розпорядження розробника засоби, що забезпечують певні гарантії ступеня захисту. Основна ідея застосування цих методів є

простою. Розглянемо відправника B, який хоче передати повідомлення m одержувачу O. Щоб захистити повідомлення від загроз безпеки, відправник спочатку зашифровує його в повідомлення m', а потім надсилає повідомлення m' одержувачу O. O, у свою чергу, повинен розшифрувати отримане повідомлення та отримати оригінал m. Шифрування та розшифрування здійснюються шляхом застосування криптографічних методів з використанням ключів. Вихідне повідомлення називається відкритим текстом (plaintext – P). Зашифроване повідомлення називається зашифрованим текстом (ciphertext – C), яке формально можна подати у такому вигляді: $C = E_K(P)$, де K – ключ для шифрування/розшифрування. Аналогічно можна представити операцію розшифрування: $P = D_K(C)$. При цьому слід пам'ятати, що при передачі повідомлення у вигляді зашифрованого тексту C можливі три різні атаки, від яких необхідно захищатись. По-перше, зловмисник може перехопити повідомлення, причому про це можуть не бути обізнані ні відправник B, ні одержувач O. Зрозуміло, якщо надіслане повідомлення було зашифровано таким чином, що його неможливо розшифрувати, не маючи відповідного ключа, перехоплення буде марним: зловмисник побачить лише незрозумілі дані. Хоча в деяких випадках самого факту передачі повідомлень буває достатньо, щоб зловмисник міг зробити відповідні висновки (наприклад, у період економічних криз, воєнних дій тощо). По-друге, можлива модифікація повідомлення. Змінити відкритий текст легко, але модифікувати зашифрований текст, який був належним чином зашифрований, набагато складніше, тому що зловмиснику спочатку доведеться розшифрувати повідомлення, перш ніж він зможе суттєво змінити його. Крім того, зловмисник також повинен правильно зашифрувати його, інакше одержувач може помітити, що повідомлення було підроблено. Третій тип атаки – це коли зловмисник вставляє зашифровані повідомлення у систему комунікації, намагаючись переконати одержувача O в тому, що це повідомлення отримано від відправника B. І знову, шифрування може допомогти захиститися від подібних атак. При цьому, слід зазначити, що якщо порушник може змінювати повідомлення, він може вставляти повідомлення. Тому сьогодні існують різні криптографічні примітиви та системи (у тому числі симетричні, асиметричні (з відкритим ключем) системи, геш-функції та деякі інші), що дозволяють успішно боротися з переліченими типами атак. Це окрема тема для обговорення, подробиці якої у цій роботі не розглядатимуться.

Як відомо, дані можуть перебувати в трьох станах (рис. 1) [42]: *у стані спокою* (at rest), *в дорозі* або *в русі* (in transit або in motion) та *у використанні* (in use). Дані в дорозі або дані в русі – це дані, що активно переміщаються з одного місця в інше, наприклад через Інтернет або через приватну мережу (надсилання електронної пошти, здійснення онлайн-купівлі, доступ до веб-сторінки, передача файлів по мережі). Дані в стані спокою – це неактивні дані, які фізично розміщуються у комп'ютерних сховищах даних у будь-якій цифровій формі (наприклад, файли на жорсткому диску комп'ютера, дані в базі даних, у хмарному сховищі, архіви, дані на USB-накопичувачі, до якого не здійснюється активний доступ, резервні копії за межами офісу (на дисках, стрічках) або в хмарі тощо), і які в даний час не використовуються. Дані у використанні – це дані, які активно обробляються або використовуються комп'ютером або застосунком (наприклад, перегляд інформації на екрані комп'ютера, обробка даних за допомогою програмного забезпечення, виконання запитів до бази даних («активні дані» в контексті знаходження в базі даних або маніпулювання ними з боку застосунку), потокове відео в реальному часі тощо) в даний момент. Кожен із цих трьох станів даних обробляється за допомогою певного набору технологій, що надаються рішеннями щодо запобігання втраті, витоку, спотворенню даних. Безпека даних у дорозі забезпечується за рахунок шифрування даних перед передачею, реалізації різних протоколів автентифікації, перевірки цілісності даних та

Рис. 1. Три стани даних

деяких інших механізмів, що захищають дані при їх передачі по мережах і запобігають їх підслуховування або перехоплення. Ідея захисту даних у дорозі була розглянута вище, нижче також будуть розглянуті деякі заходи їх захисту.

Захист даних у стані використання зазвичай включає контроль доступу, автентифікацію, шифрування даних під час обробки (захищає дані в пам'яті від злому або крадіжки [42]) і забезпечення безпеки середовища обчислень. Технології, такі як апаратні модулі безпеки (hardware security modules – HSM) [43], можуть використовуватися для захисту криптографічних ключів, коли вони знаходяться у використанні. Захист даних у стані спокою включає заходи, такі як шифрування, контроль доступу, автентифікація та фізична безпека (наприклад, закриті шафи, безпечні сховища). Наприклад, комплексна стратегія безпеки Google включає шифрування в стані спокою, яке допомагає захистити вміст клієнтів від зловмисників. Зашифровується весь контент клієнтів Google, що зберігається, без будь-яких дій з боку останніх. У Google Cloud Spanner є три рівні шифрування. Дані в стані спокою розбиваються на фрагменти підфайлів для зберігання і кожен фрагмент шифрується на рівні сховища за допомогою окремого ключа шифрування [44]. Розмір кожного фрагмента може досягати кількох гігабайт. Ключ, який використовується для шифрування даних у блоці, називається ключем шифрування даних (DEK – data encryption key). Два фрагменти не будуть мати однаковий DEK, навіть якщо вони належать одному і тому ж клієнту або зберігаються на одному комп'ютері. Якщо фрагмент даних оновлюється, він шифрується за допомогою нового ключа, а не повторним використанням існуючого ключа. Такий поділ даних, у кожному з яких використовується свій ключ, обмежує ризик потенційної компрометації ключа шифрування даних лише цим блоком. Через великий обсяг ключів у Google та необхідності малої затримки та високої доступності ці ключі зберігаються поруч із даними, які вони шифрують. DEK зашифровуються («обгортаються») за допомогою ключа шифрування ключів (KEK – key encryption key). Нарешті, кожен KEK шифрується ключем шифрування, яким керує клієнт (customer-managed encryption key). Google за допомогою алгоритму AES шифрує дані перед записом їх у систему зберігання БД або на апаратний диск. Шифрування вбудовано у всі системи зберігання. Кожен блок даних має унікальний ідентифікатор. Списки контролю доступу (ACL – access control lists) допомагають гарантувати, що кожен фрагмент може бути розшифрований лише службами Google, які працюють з авторизованими ролями, яким надається доступ лише в даний момент часу. Це обмеження доступу допомагає запобігти доступу до даних без авторизації, зміцнюючи безпеку та конфіденційність даних [45].

Широке поширення для забезпечення безпеки баз даних набула так звана технологія «прозорого шифрування даних» (TDE) [32]. Завдання TDE полягає у забезпеченні захисту даних, що зберігаються на таких носіях, як диски та магнітні стрічки, який необхідний відповідно до багатьох національних та/або міжнародних нормативних документів та правил, таких як Закон Сарбейнза – Окслі (Sarbanes-Oxley) [46], який значно посилює вимоги до фінансової звітності та процесу її підготовки, Закон про переносимість та відповідальність медичного страхування (HIPAA) [47], стандарт безпеки даних індустрії платіжних карток (PCI DSS) [48] тощо. TDE – це технологія шифрування баз даних на жорсткому диску та на будь-якому носії резервного копіювання на рівні файлів. Вона може використовуватися для забезпечення високого рівня безпеки стовпців, таблиць та табличних просторів. Прозоре шифрування даних використовується для шифрування та розшифровування даних та файлів журналів, відповідно, шифруючи дані перед їх записом на диск і розшифровує дані перед їх поверненням у застосунок. Цей процес виконується лише на рівні SQL, він повністю прозорий для застосунків і користувачів. При цьому TDE не захищає ні дані під час передачі, ні дані під час використання. Техніка TDE властива різним системам керування базами даних (СКБД). Прозоре шифрування даних використовується у продуктах компаній Microsoft, IBM, Oracle та деяких інших для шифрування файлів бази даних. Приклад прозорого шифрування даних для таблиць БД Oracle наведено на рис. 2.

Рис. 2. Приклад прозорого шифрування даних в БД Oracle

Суть прозорого шифрування полягає в тому, що використовується поєднання двох ключів: ключа для кожної таблиці бази даних, який є унікальним та майстер-ключа, що зберігається поза базою даних у гаманці [49]. Технологія прозорого шифрування передбачає, що підмножина стовпців для шифрування відома. Наприклад, якщо в табл. 4 стовпця, як показано на рис. 2, і шифруються стовпці 2 і 3, то Oracle згенерує ключ і використовує його для шифрування даних стовпців. На диску стовпці 1 і 4 будуть збережені у відкритому вигляді, а два інших стовпці – у зашифрованому. При виборі користувачем зашифрованих стовпців Oracle непомітно витягує ключ із «гаманця», розшифровує стовпці та показує їх користувачеві. Якщо диск із даними викрадено, їх неможливо витягти без ключів, які зберігаються в «гаманці», зашифрованому майстер-ключом, який сам по собі теж не зберігається у вигляді відкритого тексту. Внаслідок цього зловмисник не зможе розшифрувати дані, навіть якщо викраде диски або скопіює файли. Також за допомогою прозорого шифрування даних можна зашифрувати табличний простір (в якому зберігаються спільно такі об'єкти бази даних, як індекси, таблиці та інші). Усі об'єкти, створені в зашифрованому табличному просторі, шифруються автоматично, тобто всі дані в зашифрованому табличному просторі зберігатимуться на диску в зашифрованому вигляді. Шифрування табличного простору за допомогою прозорого шифрування є корисним, якщо ви хочете захистити всю таблицю, а не тільки окремі стовпці [50].

У NewSQL БД NuoDB підтримується прозоре шифрування даних, аналогічно використовуваному в Oracle Database, Microsoft SQL Server. TDE гарантує, що дані користувача, що зберігаються в архіві, журналі, резервних копіях, spill-файлах (файли для збереження проміжних даних, коли в пам'яті недостатньо пам'яті для виконання запиту) будуть зашифровані перед записом на диск. Інформація NewSQL БД SingleStore, включаючи файли даних, резервні копії та журнали, також захищається за допомогою прозорого шифрування CipherTrust Transparent Encryption від Thales [45].

Однак слід знати і пам'ятати, що TDE не є повномасштабною системою шифрування і не повинна використовуватись у такій якості. Для отримання комплексного рішення слід створити власний інструмент, зокрема, використовуючи можливості конкретної СКБД.

2.4. Безпечні канали

При розгляді питань безпеки у розподілених системах доречно звернутися до базової моделі їх організації – моделі клієнт-сервер. Це пов'язано, в першу чергу, з тим, що забезпечення безпеки розподіленої системи зводиться до двох основних аспектів [14]. Перший з них полягає в тому, як забезпечити безпеку зв'язку між клієнтами та серверами. Безпечний зв'язок у загальному випадку вимагає автентифікації взаємодіючих сторін, забезпечення цілісності повідомлень, а також конфіденційності. При цьому є особливості у принципах захисту зв'язку між клієнтом та групою реплікованих серверів, які також необхідно враховувати. Другий аспект – авторизація, яка пов'язана з проблемою контролю доступу клієнта до ресурсів сервера.

Питання захисту зв'язку між клієнтами та серверами доцільно розглядати з точки зору створення так званого безпечного каналу між сторонами, що взаємодіють. Безпечний канал (secure channel) – це шлях передачі даних між двома об'єктами або компонентами, який забезпечує конфіденційність, цілісність та захист від повторного відтворення, а також взаємну автентифікацію між об'єктами або компонентами. Безпечний канал (іноді його називають довіреним каналом (trusted channel) [51]) може бути забезпечений за рахунок використання прийнятих криптографічних, фізичних чи процедурних методів або їх комбінації. Далі

коротко розглянемо питання, пов'язані з автентифікацією взаємодіючих сторін, конфіденційністю повідомлень, їх цілісністю, а також особливості безпечної групової взаємодії з кількістю учасників більше двох.

Автентифікація як важливий компонент безпеки інформаційних систем.

Автентифікація є одним із важливих компонентів безпеки інформаційних систем. Згідно з нормативними документами [22, 25] *автентифікація* – це перевірка особи користувача, процесу або пристрою, часто як попередня умова для надання доступу до ресурсів в інформаційній системі. Іншими словами, автентифікація полягає у перевірці автентичності користувача, процесу або пристрою за пред'явленим ідентифікатором. Така перевірка повинна унеможливлювати фальсифікацію сутностей (користувачів, процесів, пристроїв) у системі та їх компрометацію. Без автентифікації зловмисник може отримати доступ до конфіденційної інформації або виконати небажані дії у системі від імені іншого користувача. При цьому слід зазначити, що автентифікація та цілісність повідомлень пов'язані один з одним та їх доцільно розглядати в сукупності.

Автентифікація на основі загального секретного ключа.

Розглянемо спочатку протокол автентифікації на основі спільного секретного ключа, який вже використовується Користувачем 1 та Користувачем 2. І поки неважливо, яким безпечним способом вони отримали цей спільний секретний ключ. Для опису протоколу введемо деякі позначення.

Для стислості позначимо Користувачів 1 і 2 як $U1$ і $U2$ відповідно. Їхній спільний ключ позначимо як $K_{U1,U2}$. Протокол використовує загальний підхід, при якому одна сторона запитує в іншої відповідь, яка може бути правильною, тільки якщо інша знає спільний секретний ключ. Такі рішення відомі як протоколи «виклик-відповідь».

У випадку автентифікації на основі спільного секретного ключа протокол виконується, як показано на рис. 3. Спочатку Користувач 1 надсилає свій ідентифікатор Користувачеві 2 (повідомлення 1), вказуючи, що він хоче встановити канал зв'язку між ними. Користувач 2 відповідно надсилає виклик Користувачеві 1 (повідомлення 2). Такий виклик може набувати форми випадкового числа. Користувач 1 повинен зашифрувати запит за допомогою секретного ключа $K_{U1,U2}$, яким він ділиться з Користувачем 2, та повернути зашифрований виклик Користувачеві 2 (повідомлення 3). Коли Користувач 2 отримує відповідь $K_{U1,U2}(R_{U2})$ на свій виклик R_{U2}, він може знову розшифрувати повідомлення, використовуючи спільний ключ, щоб переглянути, чи воно містить R_{U2}. Якщо це так, він знає, що Користувач 1 знаходиться на іншій стороні, тому що ніхто більше не міг зашифрувати R_{U2} за допомогою $K_{U1,U2}$.

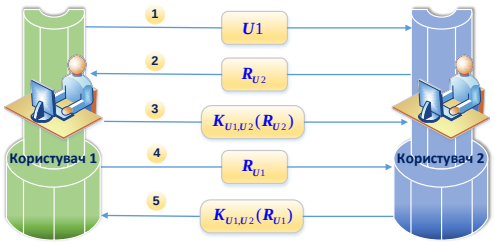

Рис. 3. Автентифікація на основі загального секретного ключа

Іншими словами, Користувач 2 тепер переконався, що він дійсно контактує з Користувачем 1. Однак, зверніть увагу, що Користувач 1 ще не підтвердив, що це дійсно Користувач 2 з іншого боку каналу. Тому він надсилає виклик R_{U1} (повідомлення 4), на який Користувач 2 відповідає, повертаючи $K_{U1,U2}(R_{U1})$, (повідомлення 5). Коли Користувач 1 розшифрує його

за допомогою $K_{U1,U2}$ та побачить свій R_{U1}, він буде впевнений, що контактує з Користувачем 2. Слід зазначити, що, налаштовуючи цей протокол для покращення його продуктивності, можна порушити його коректність, що позначиться на безпеці. Про це свідчать дослідження, які проводять розробники криптографічних протоколів протягом багатьох років [14].

Автентифікація з використанням центру розподілу ключів.

Однією із проблем використання спільного секретного ключа для автентифікації є масштабованість. Якщо розподілена система містить N хостів і кожному хосту потрібно спільно використовувати секретний ключ з кожним з решти $(N-1)$ хостів, системі в цілому необхідно керувати $N(N-1)/2$ ключами, і кожен хост може керувати $(N-1)$ ключами. Для великих N це стає проблемою. Виходом із цієї ситуації може бути рішення використати Центр розповсюдження ключів (ЦРК). ЦРК розділяє секретний ключ з кожним з хостів, при цьому жодній парі хостів спеціальний спільний секретний ключ не потрібний. Іншими словами, використання ЦРК вимагає управління всього N ключами замість $N(N-1)/2$, що явно є прогресом. Тобто, якщо Користувач 1 хоче встановити безпечний канал з Користувачем 2, він може це зробити за допомогою довіреного ЦРК. Ідея в цілому полягає в тому, що ЦРК роздає ключ Користувачам 1 та 2, який вони можуть використовувати для спілкування один з одним (рис. 4).

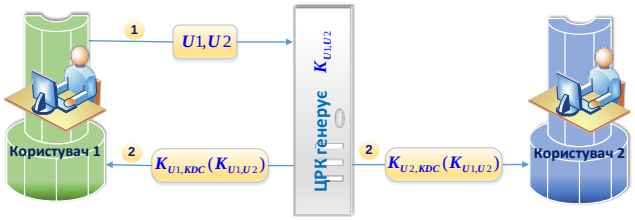

Рис. 4. Принцип використання Центру розповсюдження ключів

Користувач 1 спочатку надсилає повідомлення до ЦРК, вказуючи, що він $(U1)$ хоче спілкуватися з Користувачем 2 $(U2)$. ЦРК повертає повідомлення, що містить спільний секретний ключ $K_{U1,U2}$, який він може використовувати. Повідомлення шифрується секретним ключем $K_{U1,KDC}$, який є спільним для Користувача 1 та ЦРК. Крім того, ЦРК відправляє $K_{U1,U2}$ також Користувачеві 2, але тепер він зашифровується секретним ключем $K_{U2,KDC}$, який є спільним для Користувача 2 та ЦРК. Слід зазначити, що основним недоліком такого підходу є те, що Користувач 1 може захотіти розпочати налаштування безпечного каналу з Користувачем 2 ще до того, як Користувач 2 отримає спільний ключ від ЦРК. Крім того, ЦРК потрібно знайти Користувача 2, щоб передати йому відповідний ключ у цьому циклі налаштування. Ці проблеми можна обійти, якщо ЦРК просто передає повідомлення $K_{U2,KDC}(K_{U1,U2})$ назад Користувачеві 1 і дозволяє йому потурбуватися про з'єднання з Користувачем 2. Це призводить до протоколу, показаному на рис. 5. Повідомлення $K_{U2,KDC}(K_{U1,U2})$ також відоме як квіток. Задача Користувача 1 – передати цей квіток Користувачеві 2.

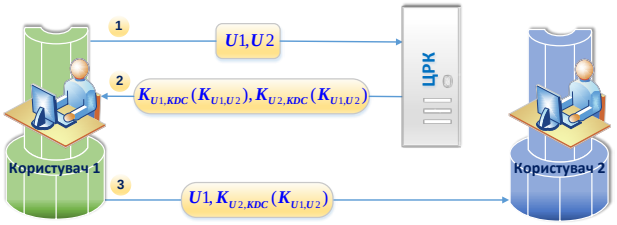

Рис. 5. Протокол встановлення з'єднання між користувачами з використанням талону

Зверніть увагу, що Користувач 2, як і раніше, єдиний, хто може осмислено використовувати квіток, оскільки він єдиний, крім ЦРК, який знає, як розшифрувати інформацію, що міститься в ньому. Протокол, показаний на рис. 5 є варіантом добре відомого протоколу автентифікації з використанням ЦРК – протоколу автентифікації Нідхема – Шредера (Needham-Schroeder) [52].

Автентифікація з використанням криптографії з відкритим ключем.

Автентифікацію користувачів можна здійснювати і без ЦРК, використовуючи можливості криптосистеми з відкритим ключем.

Цілісність та конфіденційність повідомлень.

Як зазначалося вище, крім автентифікації, безпечний канал також повинен забезпечувати цілісність і конфіденційність повідомлень. Конфіденційність легко встановлюється шляхом простого шифрування повідомлення перед його надсиланням. Шифрування може здійснюватися через секретний ключ, наданий одержувачу, або, альтернативно, через відкритий ключ одержувача. Однак забезпечити цілісність повідомлення дещо складніше. А саме, крім автентифікації, є принаймні дві проблеми, пов'язані із забезпеченням цілісності повідомлення, які слід вирішувати. Перша проблема пов'язана з тим, щоб одержувач не міг зловмисно змінювати на свою користь отримане повідомлення і стверджувати, що воно було таким, яким було представлено ним. Друга проблема пов'язана з відправником – щоб він не міг заперечувати, що повідомлення, яке він надіслав, було зовсім іншим, а не таким, яким його представив відправнику одержувач (тобто відправник фактично відмовляється від того, що сам написав). Ці дві проблеми можуть бути вирішені, якщо відправник (Користувач 1) підписує повідомлення у цифровій формі таким чином, що його підпис однозначно пов'язаний з його змістом. Унікальний взаємозв'язок між повідомленням та його підписом запобігає тому, що модифікації повідомлення залишаться непоміченими. Крім того, якщо підпис відправника може бути перевірений на автентичність, він не зможе згодом заперечувати той факт, що він підписав повідомлення. Існує кілька проблем із цією схемою, хоча сам по собі протокол правильний. По-перше, дійсність підпису Користувача 1 (відправника) зберігається лише доти, доки закритий / секретний (private) ключ Користувача 1 залишається секретом. Якщо Користувач 1 хоче відмовитись від повідомлення навіть після відправки Користувачеві 2 свого підтвердження, він може заявити, що його особистий ключ був викрадений до того, як повідомлення було надіслано. Інша проблема виникає, коли Користувач 1 вирішує змінити свій закритий ключ (це сприяє підвищенню безпеки). Але як тільки Користувач 1 змінив ключ, його повідомлення, надіслане Користувачеві 2, стає марним. У таких випадках може знадобитися центральний орган, який відстежує зміну ключів на додаток до використання міток часу під час підписання повідомлень.

Ще однією проблемою подібної схеми є те, що Користувач 1 шифрує все повідомлення своїм закритим ключем. Таке шифрування може бути дорогим з точки зору вимог до обробки (або навіть математично нездійсненним, оскільки передбачається, що повідомлення, яке інтерпретується як двійкове число, обмежене заздалегідь певним максимумом), і при цьому насправді в ньому немає необхідності. А в чому є необхідність, то це в тому, щоб унікально пов'язати підпис з єдиним конкретним повідомленням. Дешевшою та практичною схемою є використання дайджесту повідомлення (message digest). Дайджест повідомлення – це результат застосування геш-функції до повідомлення (також відомий як «геш-значення») [53]; геш-значення – бітовий рядок фіксованої довжини, створений геш-функцією [54]; тобто під дайджестом повідомлення можна розуміти бітовий рядок фіксованої довжини h, який був обчислений з повідомлення m довільної довжини за допомогою криптографічної геш-функції H. Якщо m змінити на m', його геш $H(m')$ відрізнятиметься від $h = H(m)$, щоб можна було легко виявити, що відбулася модифікація.

Для цифрового підпису повідомлення Користувач 1 може спочатку обчислити дайджест повідомлення, а потім зашифрувати дайджест своїм закритим ключем, як показано на рис. 6.

Зашифрований дайджест надсилається разом із повідомленням Користувачеві 2. Слід зауважити, що саме повідомлення надсилається у вигляді відкритого тексту (кожен може його прочитати). Якщо потрібна конфіденційність, повідомлення має бути зашифроване відкритим ключем Користувача 2.

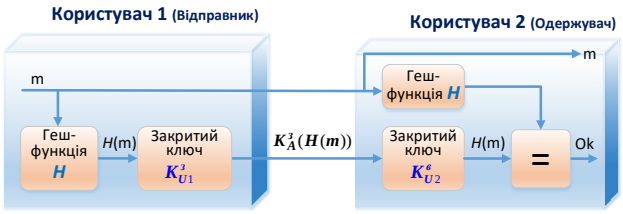

Рис. 6. Цифровий підпис повідомлення за допомогою дайджеста повідомлення

Коли Користувач 2 отримує повідомлення та його зашифрований дайджест, йому потрібно просто розшифрувати дайджест за допомогою відкритого ключа Користувача 1 та окремо розрахувати дайджест повідомлення. Якщо дайджест, отриманий із повідомлення, та розшифрований дайджест збігаються, Користувач 2 знає, що повідомлення було підписано Користувачем 1.

2.5. Контроль доступу

Для кращого розуміння проблем, пов'язаних з контролем доступу (керуванням доступу), доцільно звернутися до простої моделі, наведеної на рис. 7.

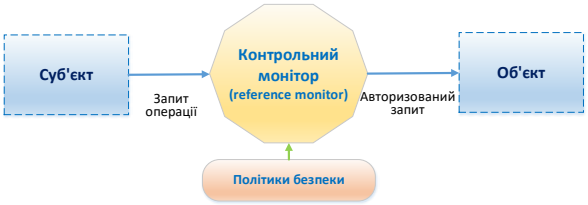

Рис. 7. Загальна модель керування доступом до об'єктів

Модель складається із суб'єктів, які видають запит на доступ до об'єкта. Об'єкт дуже схожий на об'єкти, які розглядалися досі (сервери, пристрої, програми, процеси, дані тощо). Об'єкт можна уявити як інкапсуляцію власного стану та реалізацію операцій над цим станом. Операції об'єкта, які можуть запросити суб'єкти, виконуються через інтерфейси. Суб'єкти найкраще розглядати як процеси, що діють від імені користувачів, але вони також можуть бути об'єктами, які потребують послуг інших об'єктів для виконання своєї роботи. Керування доступом до об'єкта полягає у захисті об'єкта від викликів суб'єктів, яким не дозволено виконувати певні (або ніякі) методи. Крім того, захист може включати питання управління об'єктами, такі як створення, перейменування або видалення об'єктів. Захист забезпечується так званим контрольним монітором / монітором звернень (reference monitor).

На підставі встановлених політик безпеки контрольний монітор визначає, чи може суб'єкт виконувати конкретну операцію. Монітор викликається (наприклад, базовою операційною системою) щоразу, коли викликається об'єкт. Отже, дуже важливо, щоб контрольний монітор сам по собі був захищений від злому, тобто зловмисник не повинен мати доступу до нього. Забезпечити безпеку легше, якщо є чітка модель того, що потрібно захищати та кому і що дозволено робити [14]. Тому невід'ємною частиною будь-якого проекту створення безпеки комп'ютерних систем є наявність моделі безпеки (security model), під якою розуміється формальне уявлення політики безпеки [55]. Сьогодні в комп'ютерних системах знайшли широке поширення моделі безпеки на основі дискреційної, мандатної, рольової політики, а також політики атрибутів.

2.5.1. Дискреційний контроль доступу

Роботи з моделей дискреційного (вибіркового) керування доступом (discretionary access control – DAC) до даних інформаційних систем (ІС) з'явилися ще в 60 – 70-х роках минулого століття. Вони досить широко висвітлені у науковій літературі. Одна з перших таких моделей була розроблена Лемпсоном (Lampson) [56, 57], а потім удосконалена Грехемом (Graham) і Деннінгом (Denning) [58]. Модель Грехема – Деннінга сформувала основу для наступних систем безпеки, наприклад для широко поширеної моделі Харрісона – Руццо – Ульмана – HRU (Harrison – Ruzzo – Ullman) [59]. Крім того, також відомі такі моделі як модель ADEPT-50 [60], п'ятивимірний простір Хартсона [61], модель Take-Grant [62] та деякі інші. Авторами цих моделей було внесено значний внесок у теорію безпеки комп'ютерних систем. Їхні роботи заклали основу для подальшого створення та розвитку захищених ІС.

У теоретичному та практичному плані найбільшого розвитку та застосування отримали дискреційні моделі, засновані на матриці доступу (матриці контролю доступу) – M, яка описує права доступу суб'єктів (S) до об'єктів (O), рядки якої відповідають суб'єктам доступу $s_1, s_2, ..., s_m$, стовпці об'єктам доступу $o_1, o_2, ..., o_n$, а в елементах матриці $M[s_i, o_j]$ записуються дозволені операції (види доступу) $op_1, op_2, ..., op_L$ (наприклад, читання (r_d), запис з модифікацією (w), запис без модифікації (тільки з новим записом або дописуванням у файл) (a), запуск об'єкта на виконання (e)) відповідного суб'єкта над відповідним об'єктом. Як зазначається в монографії [63], за потреби елементи матриці можуть містити вказівники на процедури. Ці процедури виконуються під час кожної спроби доступу до заданого об'єкта. Тим самим рішення про доступ може прийматися на підставі складніших залежностей, не настільки очевидних, як у простій матриці доступу. Ця модель передбачає, що усі спроби доступу до об'єктів перехоплюються і перевіряються спеціальним керуючим процесом. Таким чином, суб'єкт s_i отримає ініційований ним доступ op_l до об'єкта o_j лише у випадку, якщо елемент матриці $M[s_i, o_j]$ має значення op_l. Однак, враховуючи, що системі може знадобитися підтримка тисяч суб'єктів (користувачів) та мільйонів об'єктів, які потребують захисту, реалізація матриці контролю доступу в якості істинної матриці не є підходящим способом. Багато записів у матриці будуть порожніми: один суб'єкт, як правило, матиме доступ до відносно невеликої кількості об'єктів. У цьому випадку доцільно використовувати більш ефективний спосіб. Один із широко застосовуваних підходів полягає в тому, що кожен об'єкт підтримує список прав доступу суб'єктів, які хочуть отримати доступ до об'єкта. По суті це означає, що матриця розподілена по стовпцях по всіх об'єктах, а порожні записи пропущені (рис. 8). Цей тип реалізації призводить до того, що називається списком контролю доступу (ACL). Передбачається, що кожен об'єкт має свій власний ACL, тобто для кожного об'єкта ACL перераховані суб'єкти та їх дозволені права доступу. Інший підхід полягає у розподілі матриці по рядках шляхом надання кожному суб'єкту списку можливостей (capability list), які він має для кожного об'єкта (рис. 9). Іншими словами, можливість відповідає запису в матриці контролю доступу. Відсутність можливості (capability) для конкретного об'єкта означає, що суб'єкт немає прав доступу до цього об'єкту.

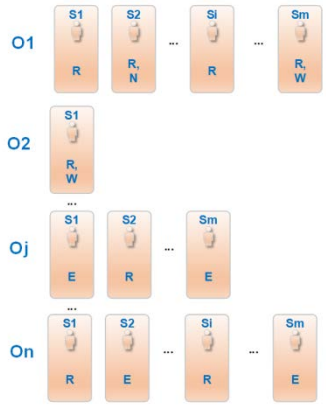

Рис. 8. Список контролю доступу

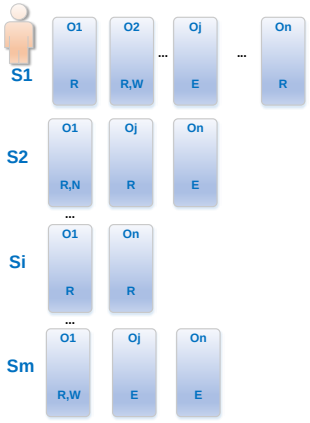

Рис. 9. Список можливостей

Можливість можна порівняти з квитком: її власнику надаються певні права, пов'язані з цим квитком. При цьому квиток має бути захищений від змін його власником. Один підхід, який особливо застосовується у розподілених системах, полягає у захисті списку можливостей за допомогою підпису. Різниця між тим, як контроль доступу ACL та список можливостей використовуються для захисту доступу до об'єкта, показано на рис. 10.

Коли клієнт надсилає запит на сервер, серверний контрольний монітор, використовуючи ACL, перевіряє, чи він знає клієнта і чи дозволено виконувати запитану операцію, як показано на рис. 10, *a*. У разі використання списку можливостей клієнт просто передає свій запит разом зі списком можливостей на сервер. Сервер не зберігає відомостей про клієнта, оскільки вся потрібна йому інформація для відповідних дій міститься у переданій можливості. Отже, серверу потрібно лише перевірити, чи дійсна ця можливість і чи вказана запитана операція у списку можливостей. Цей підхід до захисту об'єктів (на підставі можливостей) показано на рис. 10, *b*.

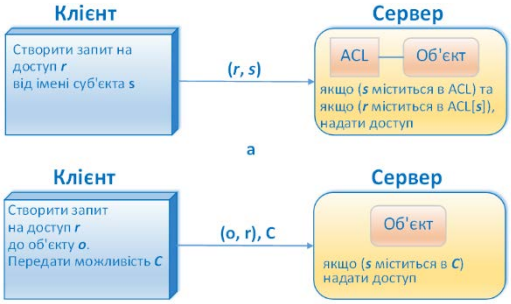

Рис. 10. Порівняння варіантів захисту за допомогою ACL та списку можливостей:
a) використання ACL;
b) використання можливостей

ACL та можливості допомагають ефективно реалізувати матрицю контролю доступу, ігноруючи всі порожні записи. Проте ACL або список можливостей також може стати досить великим, якщо не буде вжито додаткових заходів. Одним із загальних способів скорочення списків контролю доступу є використання доменів захисту (protection domain – набір пар <*об'єкт, права доступу*>). Кожна пара визначає для даного об'єкта, які саме операції дозволено виконувати. Отже, кожного разу, коли суб'єкт запитує виконання операції над об'єктом, контрольний монітор спочатку шукає домен захисту, пов'язаний із цим запитом. Потім, з урахуванням домену, монітор може перевірити, чи дозволено виконання запиту. Існують різні варіанти використання доменів захисту, наприклад, створення груп користувачів (у тому числі ієрархічних), ролей. В цілому ж, існують різні підходи, а також їх комбінації, які використовуються для керування доступом.

2.5.2. Контроль доступу на основі мандатної політики

Якщо в дискреційних моделях керування доступом відбувається шляхом надання суб'єктам повноважень для здійснення певних операцій над конкретними об'єктами, то мандатні моделі керують доступом неявним чином – за допомогою призначення всім сутностям системи (суб'єктам, об'єктам) рівнів безпеки, які визначають всі допустимі взаємодії між ними. Отже, мандатна модель керування доступом (mandatory access control – MAC) не розрізняє сутностей, яким присвоєно однаковий рівень безпеки, і на їх взаємодії обмеження відсутні. Тобто мандатний підхід до розмежування доступу, ґрунтуючись лише на парадигмі ранжо-

ваної довіри, без урахування специфіки інших характеристик суб'єктів і об'єктів, призводить в більшості випадків до надмірності прав доступу для конкретних суб'єктів в межах відповідних класів безпеки, що суперечить самому поняттю розмежування доступу. Тому в тих ситуаціях, коли керування доступом вимагає більш гнучкого підходу, мандатний принцип розмежування доступу доповнюється дискреційним всередині відповідних класів безпеки. У теоретичних моделях для цього вводять матрицю доступу, що розмежовує дозволений за мандатним принципом доступ до об'єктів одного рівня безпеки. Наприклад, у СКБД Oracle реалізація технології мандатного доступу накладається на реалізацію дискреційної моделі. Так, реалізація технології мандатного доступу, що ґрунтується на механізмі OLS (Oracle Label Security), спирається не лише на дискреційну модель доступу (спочатку перевіряються права суб'єкта на виконання відповідної операції над таблицею), а й на механізм VPD (якщо суб'єкт має відповідні привілеї, перевіряється, чи не прикріплені до таблиці будь-які політики VPD – Virtual Private Database). Після всього цього перевіряється наявність політик OLS, призначених таблиці, що захищається: порівнюються мітки, присвоєні окремим рядкам, з авторизацією міток користувачів, дозволяючи або забороняючи доступ [64, 65].

Найбільш широке поширення серед моделей мандатного керування доступу (багаторівневого захисту) отримала модель Белла – ЛаПадули (Bell-LaPadula model) [66]. Графічне представлення моделі Белла – ЛаПадули показано на рис. 11. На рис. 11 суцільна стрілка від об'єкта до суб'єкта показує, що суб'єкт здійснює читання об'єкта (інформаційний потік йде від об'єкта до суб'єкта). Пунктирна стрілка від суб'єкта до об'єкта показує, що суб'єкт здійснює запис в об'єкт (інформаційний потік йде від суб'єкта до об'єкта). Таким чином, направлення інформаційних потоків вказуються стрілками (наприклад, суб'єкт B може читати дані з об'єкта 1, але не може зчитувати дані з об'єкта 3).

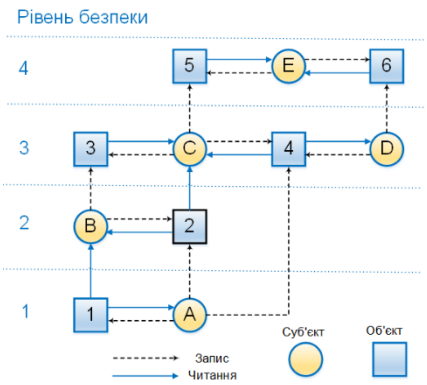

Рис. 11. Багаторівнева модель безпеки Белла-ЛаПадули

Модель Белла – ЛаПадули зіграла величезну роль у розвитку теорії комп'ютерної безпеки, і її положення були введені в якості обов'язкових вимог до систем, що обробляють інформацію, що містить державну таємницю, в стандартах захищених інформаційних систем. Однак при практичній реалізації моделі Белла – ЛаПадула виникає ряд проблем, наприклад, таких як [64]: завищення рівня безпеки; запис наосліп; привілейовані суб'єкти. Розширення моделі Белла – ЛаПадули, пов'язані з пошуком умов і обмежень, що підвищують її безпеку, також не знімають всіх недоліків мандатної доступу. Зокрема, мандатний доступ знімає проблему «троянських програм», але тільки з точки зору небезпечних потоків «зверху вниз».

В цілому ж основним недоліком багаторівневих моделей є неможливість керування доступом до конкретних об'єктів на основі врахування індивідуальних особливостей кожного з суб'єктів. Таким чином, обидва розглянуті вище підходи не в повній мірі можуть ефективно і гнучко управляти безпечним доступом до даних. Отже, обидва підходи як би припускають пошук різних компромісів між ефективністю, гнучкістю і безпекою. Очевидно, що оптимальне вирішення питань безпеки має здійснюватися із застосуванням обох видів моделей.

2.5.3. Керування доступом на основі ролей

Неважко бачити, що розглянуті підходи передбачають пошук різних компромісів між ефективністю, гнучкістю та безпекою. Очевидно, що для забезпечення безпеки систем необхідно використовувати можливості обох розглянутих вище моделей. Такі можливості можна реалізувати, використовуючи модель керування доступом на основі ролей (role-based access control – RBAC) [67]. Основою моделі RBAC є додатково введена в суб'єктно-об'єктну модель системи категорія активних сутностей – роль. Модель керування доступом на основі ролей визначає особливий тип політики, заснований на компромісі між гнучкістю керування

доступом, характерною для дискреційних моделей, і жорсткістю правил контролю доступу, властивою мандатним моделям. У RBAC моделі класичне поняття суб'єкт розділяється на дві складові: користувач і роль. Користувач – це людина, що працює з системою і виконує певні службові обов'язки. Роль – це активно діюча в системі абстрактна сутність, з якою пов'язується певний набір повноважень (привілеїв), необхідних для здійснення певної діяльності. Керування доступом при використанні рольової політики здійснюється в два етапи: 1) створення ролей і визначення їх повноважень (прав доступу до об'єктів); 2) призначення ролей користувачам системи. Слід зазначити, що користувач може бути асоційований з декількома ролями. Дана можливість значно спрощує адміністрування складних систем. Керування доступом на основі ролей вимагає розбиття процесу функціонування системи і роботи користувача на сеанси. RBAC модель описує систему у вигляді наступних множин [67]: U – множина користувачів; R – множина ролей; P – сукупність повноважень на доступ до об'єктів (реалізована, наприклад, у вигляді матриці доступу); C – множина сеансів роботи користувачів з системою. Взаємозв'язок користувачів, ролей, повноважень (привілеїв) і сеансів показаний на рис. 12.

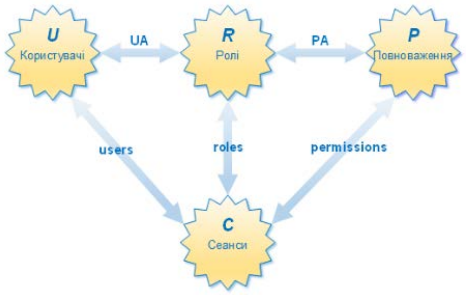

Рис. 12. Взаємозв'язок ролей, повноважень, користувачів і сеансів

Основне правило (критерій безпеки) рольового доступу визначається наступним чином: система вважається безпечною, якщо і тільки якщо будь-який користувач $u \in U$ в системі, що працює в сеансі $c \in C$, може здійснювати дії, що вимагають повноважень $p \in P$, тільки в тому випадку, якщо $p \in permissions(c)$, де $permissions(c)$ – набір доступних у сеансі c повноважень. До переваг моделі керування доступом на основі ролей можна віднести гнучкі, що динамічно змінюються в процесі функціонування систем правила розмежування доступу до ресурсів PIC з великою кількістю користувачів та об'єктів, у тому числі завдяки можливості побудови ієрархій ролей. Оперувати ролями набагато зручніше ніж суб'єктами, оскільки це більш відповідає поширеним технологіям обробки інформації, що передбачають розділення обов'язків і сфер відповідальності між користувачами. Однак для систем керування, доступ в яких ґрунтується на механізмі ролей, немає строгих доказів безпеки відповідно до визначених формалізованих критеріїв. Такий підхід дозволяє отримувати прості і зрозумілі правила контролю доступу, які легко можуть бути застосовані на практиці, але позбавляє систему теоретичної доказової бази. Тому безпека RBAC моделі ґрунтується на контрольних механізмах дискреційних або мандатних моделей, засобами яких регулюється доступ рольових суб'єктів (суб'єктів-ролей) до об'єктів системи.

2.5.4. Керування доступом на основі атрибутів

Керування доступом на основі атрибутів (attribute-based access control – ABAC) – це модель, яка еволюціонує з моделі RBAC. В останні роки цей підхід до керування доступом привернув значну увагу з боку бізнесу, наукових кіл та органів стандартизації. Системи, що підтримують механізм керування доступом на основі атрибутів, здатні реалізовувати концепції як дискреційного контролю доступу (DAC), так і мандатного контролю доступу (MAC). Більш того, системи ABAC можуть забезпечувати рішення керування доступом з адаптацією до ризиків (risk-adaptable access control – RAdAC – контроль доступу, що адаптується до ризику), при цьому значення ризику виражаються у вигляді змінних атрибутів [68]. ACL і RBAC у певному сенсі є особливими випадками ABAC з погляду використовуваних атрибутів [69]. Але при цьому, на відміну від керування доступом на основі ролей, керування доступом на основі атрибутів може виражати складні набори правил (у цьому підході немає обмежень на складність бізнес-правил), що включають можливість оцінки безлічі різних атрибутів. Кожна ситуація оцінюється не з точки зору ролі користувача та дії, яку він хоче вчинити, а з погляду атрибутів, які до них відносяться. Шляхом визначення узгоджених

атрибутів суб'єкта та об'єкта у політиках безпеки ABAC усуває необхідність у явних авторизаціях окремих суб'єктів, необхідних у методі доступу, відмінному від ABAC, що знижує складність керування списками та групами доступу. ABAC визначає доступ (допустимі операції/дії над об'єктами) шляхом зіставлення поточних значень атрибутів сутностей (суб'єкта та об'єкта) та умов середовища з вимогами, зазначеними у правилах керування доступом [69]. Атрибути можна розглядати як характеристики всього, що може бути визначено і чому може бути присвоєно значення. Правила або політики, які можуть бути реалізовані в моделі ABAC, обмежені лише мовою обчислень та безліччю доступних атрибутів. Ця гнучкість дозволяє найбільшій кількості суб'єктів отримати доступ до найбільшої кількості об'єктів без вказівки індивідуальних відносин між кожним суб'єктом і кожним об'єктом, що робить цей підхід керування доступом, ідеальним для багатьох розподілених або швидко мінливих середовищ [70]. ABAC забезпечує підвищену точність визначення політики порівняно з попередніми моделями, дозволяючи використовувати більше вхідних даних для ухвалення рішення щодо контролю доступу, надаючи більший набір можливих комбінацій цих факторів, щоб відобразити ширший та більш визначений набір можливих правил для вираження політики. В результаті можна легко налаштовувати та змінювати політики ABAC у міру зміни потреб.

Хоча в даний час не існує єдиного узгодженого визначення ABAC, існують загальноприйняті визначення з авторитетних джерел та опис його функцій. Одне з таких визначень наведено в роботі [69]: «Керування доступом на основі атрибутів (ABAC) – метод керування доступом, при якому запити суб'єкта на виконання операцій з об'єктами дозволяються або відхиляються на основі призначених атрибутів суб'єкта, призначених атрибутів об'єкта, умов середовища та набору політик, визначених з погляду цих атрибутів та умов». Крім того, у цій же роботі визначаються основні поняття, що містяться у наведеному визначенні та мають безпосереднє відношення до суті даного підходу (моделі) до керування доступом. А саме: атрибути – це характеристики суб'єкта, об'єкта або умов середовища; атрибути містять інформацію, задану парою «ім'я-значення». Суб'єкт – це людина-користувач або не фізична особа (non-person entity – NPE, наприклад, автономна служба або програма, яка видає запити доступу для виконання операцій з об'єктами). Суб'єктам надається один або кілька атрибутів. Передбачається, що суб'єкт та користувач є синонімами. Об'єкт – це системний ресурс, доступ до якого керується системою ABAC, наприклад, пристрої, файли, записи, таблиці, процеси, програми, мережі або домени, які містять або отримують інформацію. Це може бути ресурс або запитаний об'єкт, а також усе, над чим може бути виконана операція суб'єктом, включаючи дані, застосунки, служби, пристрої та мережі. Операція (дія) – це виконання функції на запит суб'єкта над об'єктом. Операції включають читання, запис, редагування, видалення, копіювання, виконання та зміну. Політика – це подання правил або відносин, які дозволяють визначити, чи слід дозволити запитаний доступ, враховуючи значення атрибутів суб'єкта, об'єкта і, можливо, умов середовища. Умови середовища (environment conditions) – операційний чи ситуативний контекст, у якому виникають запити доступу. Умови середовища – це зумовлені показники середовища. Характеристики середовища не залежать від суб'єкта або об'єкта і можуть включати поточний час, день тижня, місцезнаходження користувача або поточний рівень загрози.

Сьогодні існує кілька моделей (базова ABAC [69], HGABAC (Hierarchical Attribute-Based Access Control) [71], LaBAC (Label-Based Access Control) [72] та деякі інші [73]), стандартів (стандарт розширюваної мови розмітки керування доступом (Extensible Access Control Markup Language – XACML) [74], стандарт контролю доступу наступного покоління – NGAC (Next Generation Access Control) [75 – 77], дослідницьких прототипів та продуктів, що втілюють концепції ABAC. У сукупності ці концепції визначають ABAC як систему контролю доступу, яка включає: дані контролю доступу для вираження атрибутів і політик; набір адміністративних операцій (мовою політики) для налаштування даних контролю доступу; набір функцій для реалізації політики щодо запитів на виконання операцій над об'єктами та

для вироблення рішень про доступ для задоволення або відхилення цих запитів на основі поточного стану даних керування доступом. Ця система охоплює чотири рівні функціональної декомпозиції: виконання, ухвалення рішення, адміністрування, дані керування доступом, а також кілька компонентів, що працюють разом. Спільними для всіх моделей ABAC є два типи атрибутів [78]: 1) *атрибут суб'єкта*; кожному суб'єкту призначається набір атрибутів, які можуть представляти особистість суб'єкта, вік, ролі, належність або інші загальні характеристики політики, наприклад рівень допуску; 2) *атрибути об'єкта*; кожному об'єкту призначається набір атрибутів. Атрибути об'єкта (їх іноді називають атрибутами ресурсу [69]) характеризують дані та інші ресурси шляхом ідентифікації колекцій об'єктів, наприклад пов'язаних з певними проектами, застосунками або класифікаціями безпеки.

Атрибути суб'єкта та об'єкта можуть бути призначені їх сутностям або за допомогою адміністративних дій, або за допомогою властивостей або метаданих, які підтримуються системою. На додаток до атрибутів суб'єкта та об'єкта існують атрибути середовища (також відомі як умови середовища), які є загальними для кількох, але не для всіх моделей. Атрибути середовища / навколишнього середовища (environment/environmental attributes) – це атрибути, які залежать від наявності системних датчиків, які можуть виявляти та повідомляти значення. Вони дещо відрізняються від атрибутів суб'єкта та об'єкта (ресурсу), оскільки не є властивостями останніх, а є вимірними характеристиками, що належать до оперативного чи ситуаційного контексту, у якому виникають запити на доступ.

Керування доступом на основі атрибутів показано на рис. 13, де:

1) суб'єкт (користувач) запитує доступ до об'єкта (деякого ресурсу);

2) механізм контролю доступу ABAC ACM (access control mechanism), отримавши запит від суб'єкта, оцінює: а) правила (на основі яких приймається рішення на доступ до об'єкту); б) атрибути суб'єкта (атрибути суб'єкта у профілі користувача можуть включати, наприклад, ідентифікатор, посаду, членство у групах, членство у підрозділах та організаціях, рівень керівництва, рівень допуску та інші критерії ідентифікації; ці дані часто беруться із системи, відділу кадрів чи іншим чином); в) атрибути об'єкта (наприклад, дата створення файлу, його власник, ім'я та тип файлу, конфіденційність даних); г) умови середовища для прийняття рішення (всі атрибути навколишнього середовища пов'язані з контекстуальними факторами, такими, як час і місце спроби доступу, місцезнаходження та пристрій суб'єкта, протокол зв'язку і надійність шифрування; контекстна інформація також може включати сигнали ризику, встановлені організацією, такі як ступінь надійності автентифікації, звичайні моделі поведінки суб'єкта тощо). Потім ABAC ACM визначає, які операції / дії суб'єкт може виконувати з об'єктом;

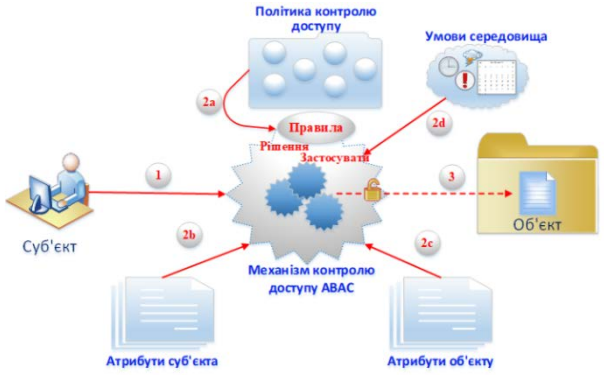

Рис. 13. Керування доступом на основі атрибутів

3. Суб'єкту надається доступ до об'єкта, якщо його авторизовано.

ABAC добре адаптований для керування доступом до розподілених систем, оскільки ABAC надає докладні визначення та мета-атрибути, які підтримують призначення привілеїв на основі структури PIC, яка потребує управління федерацією та автономією між скоординованими системами в PIC [78].

Загалом до переваг ABAC можна віднести: а) гнучкість (дозволяє реалізувати різні політики контролю доступу, обмежені лише широтою спектру доступних атрибутів та можливостями, які може виразити комп'ютерна мова); б) точність контролю доступу (дозволяє більш точно, на відміну від розглянутих вище моделей (MAC, DAC, RBAC), визначити, хто має доступ до яких ресурсів залежно від конкретних атрибутів); в) зниження рівня ризику (завдяки точнішому контролю доступу); г) легкість керування (атрибути та політики можуть

бути легше змінені відповідно до потреб організації без необхідності зміни всієї системи та перевірені); д) можливість задоволення складних вимог безпеки (ABAC може допомогти організаціям дотримуватись нормативних вимог, таких як GDPR або HIPAA, шляхом більш детального контролю доступу до конфіденційних даних).

До недоліків ABAC можна віднести: а) складність реалізації. Реалізація ABAC може бути складною та вимагати значних зусиль на етапі проектування та впровадження. Особливо це актуально для великих та складних систем (адміністраторам необхідно вручну визначати атрибути, призначати їх кожному компоненту та створювати політики на основі різних умов); б) складність адміністрування. Керування політиками ABAC та атрибутами може вимагати високої кваліфікації адміністраторів та спеціалістів з безпеки; в) продуктивність. Використання багатьох атрибутів для прийняття рішень про доступ може вплинути на продуктивність системи, особливо при великій кількості запитів на доступ.

Більш детально з різними модифікаціями моделей ABAC та їх можливостями при розгортанні ABAC у різних архітектурах застосунків (великі дані, веб-сервіси, робочі процеси) та у комерційних продуктах можна ознайомитись у роботі [78].

2.5.5. Використання міжмережевого екрану

Викладені вище моделі контролю доступу можуть чудово застосовуватися при розробці автономної розподіленої системи, яка більш менш ізольована від решти світу. Однак, коли доступ до ресурсів відкривається стороннім користувачам (наприклад, надсилання пошти, доступ до веб-сайтів, надання локальних ресурсів тощо), завдання забезпечення його контролю суттєво ускладняються. Щоб захистити ресурси в цих умовах, потрібен інший підхід. На практиці зовнішній доступ до будь-якої частини розподіленої системи контролюється спеціальним монітором, відомим як міжмережевий екран / брандмауер (firewall). Міжмережевий екран (МЕ) – це частина комп'ютерної системи або мережі, призначена для блокування несанкціонованого доступу та дозволу зовнішнього зв'язку [22]. Використовуючи МЕ для контролю підключення, організація може запобігти несанкціонованому доступу до своїх систем і ресурсів. МЕ – це міжмережевий шлюз (inter-network gateway), який обмежує трафік передачі даних в одну з підключених мереж та з неї (та, яка, як кажуть, знаходиться «всередині» МЕ) і, таким чином, захищає системні ресурси цієї мережі від загроз з іншої мережі (тієї, яка знаходиться, як кажуть, «за межами» брандмауера) [79, 80]. Брандмауери – це пристрої або програми, які контролюють потік мережевого трафіку між мережами або вузлами, що використовують різні заходи безпеки [81]. Брандмауери можна використовувати для логічного застосування користувацьких наборів правил до мережевого трафіку. МЕ, які зазвичай розміщуються на межах мережі, можуть обмежувати як вхідний, так і вихідний трафік на основі різних характеристик даних [80]. Проста маршрутизована мережа з брандмауером представлена на рис. 14.

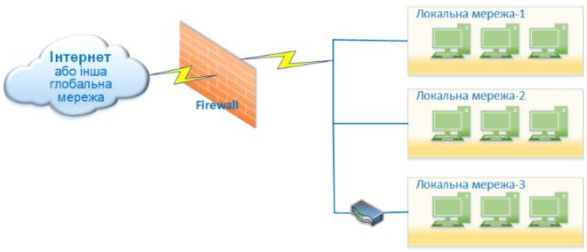

Рис. 14. Проста маршрутизована мережа з брандмауером

Можна виділити такі типи технологій міжмережевих екранів [81 – 84]: *МЕ із фільтрацією пакетів* (packet filtering firewall); *шлюз рівня каналу* (circuit-level gateway); *шлюз рівня програми* (application-level gateway) – також відомий як проксі-брандмауер (proxy firewall); *МЕ з відстеженням / перевіркою стану* (stateful inspection firewall); *МЕ наступного покоління* (next-generation firewall – NGFW).

Міжмережевий екран із фільтрацією пакетів працює як маршрутизатор і приймає рішення щодо того, чи слід передавати мережевий пакет на основі адреси джерела та одержувача, що міститься в заголовку пакета. Він безпосередньо перевіряє поточний мережевий трафік на

рівнях OSI 3 (мережевий рівень) та 4 (транспортний рівень), з метою ухвалення рішення про те, чи слід відкидати або пересилати пакети до місця призначення.

Фільтр має набір налаштованих правил на основі IP-адреси призначення та джерела, номера порту та іншої інформації. Якщо пакет не відповідає жодному з цих правил, він або автоматично відкидається, або генерується повідомлення ICMP (Internet Control Message Protocol), що повідомляє джерело відкинутого пакета. МЕ з фільтрацією пакетів є досить простими і не споживають багато ресурсів. Однак вони не є найефективнішими МЕ, оскільки їх легко оминути. У таких МЕ відсутня можливість аналізу протоколів вищих рівнів мережевої моделі OSI. Крім того, недоліком методу, що використовується в них, є те, що можуть існувати конфліктуючі правила, які необхідно вирішити для деяких пакетів.

Шлюзи рівня каналу працюють лише на рівні TCP. TCP-з'єднання передаються через комп'ютер, що по суті діє як провідник [82]. Для виявлення шкідливого контенту, шлюзи на рівні каналу відстежують TCP-рукостискання та інші повідомлення про ініціювання сеансу мережевого протоколу по всій мережі, коли вони встановлюються між локальним та віддаленим вузлами, щоб визначити, чи є сеанс, що ініціюється, законним (чи вважається віддалена система довіреною). Однак вони не перевіряють самі пакети, тому не є найкращим способом запобігання проникненню шкідливого програмного забезпечення в мережу. Хоча шлюзи на рівні каналів забезпечують більш високий рівень безпеки, ніж МЕ з фільтрацією пакетів, їх слід використовувати спільно з іншими системами. Наприклад, шлюзи рівня каналу зазвичай використовуються разом із шлюзами рівня застосунку.

Проксі-брандмауер (*шлюз рівня застосунку*) є одним із найбезпечніших типів МЕ [84]. Він знаходиться між захищеною мережею та рештою світу, функціонуючи як єдина точка входу в мережу та точка виходу з неї. Проксі-брандмауер блокує запит від системи у внутрішній мережі перед його відправкою за призначенням. Він веде себе як сервер при взаємодії з хостом клієнта і як клієнт – при надсиланні або отриманні даних від хоста сервера [83]. При цьому сервер та клієнт ніколи не мають прямого з'єднання. Шлюзи рівня застосунків мають перевагу, яка в деяких середовищах дуже важлива – весь вхідний та вихідний трафік легко реєструвати та контролювати. Проксі-брандмауер встановлює з'єднання з джерелом, а потім перевіряє вміст пакета (у тому числі наявність шкідливого ПЗ). Пошту можна перевірити на наявність «брудних» слів (dirty words), що вказують на те, що через шлюз проходять конфіденційні або обмежені дані. Веб-запити можна перевірити на відповідність політикам компанії, а небезпечні поштові вкладення можна видалити [82]. Таким чином, тільки якщо дані схвалені, вони надсилаються за призначенням. Таким чином, проксі-сервери додають рівень захисту, забезпечуючи анонімність пристроїв усередині мережі. Однак, хоча шлюзи, що здійснюють фільтрацію на рівні застосунків, забезпечують значну безпеку даних, вони можуть суттєво вплинути на продуктивність мережі (можливе уповільнення швидкості передачі) і ними може бути складно керувати.

Міжмережевий екран з відстеженням стану – це тип МЕ, що поєднує в собі технології, що раніше обговорювалися (фільтрації пакетів і шлюзів рівня каналу), для забезпечення більшої безпеки. Насправді перевірка стану є дуже складною версією пакетного фільтра [84]. Даний тип МЕ може відстежувати стан мережевих підключень, таких як потоки TCP, дейтаграми UDP та повідомлення ICMP, і може зберігати важливі властивості кожного з'єднання в пам'яті. У сукупності ці властивості називаються станами з'єднання і можуть включати такі відомості, як IP-адреси та порти, що беруть участь у з'єднанні, а також порядкові номери пакетів, що проходять через з'єднання. Для відстеження стану з'єднань використовуються спеціальні таблиці сеансів. Точність роботи МЕ цього залежить від обробки відповідних таблиць сеансів, а також від механізму фільтрації пакетів. Брандмауери з відстеженням стану використовують дані як про минулі, так і про поточні мережеві з'єднання при прийнятті рішень щодо фільтрації, тим самим пропонуючи більше можливостей (у тому числі високий рівень контролю над тим, який контент потрапляє до мережі або виходить із неї), але зі збільшенням обчислювальних витрат [81].

Міжмережеві екрани наступного покоління розширюють можливості брандмауерів із перевіркою стану, додаючи такі функції, як фільтрація застосунків, глибока перевірка пакетів (DPI – deep packet inspection), відстеження VPN-трафіку, адаптивні правила та виявлення загроз [80]. Згідно з визначенням аналітиків Gartner [85], NGFW – це міжмережеві екрани з глибокою перевіркою пакетів, які виходять за рамки перевірки та блокування портів / протоколів та додають перевірку на рівні застосунків, запобігання вторгненням та отримання відомостей ззовні брандмауера. NGFW не слід плутати з автономною системою запобігання вторгненням в мережу (IPS – intrusion prevention system), яка включає стандартний або некорпоративний МЕ, або МЕ і IPS в одному пристрої, які не є тісно інтегрованими. Як мінімум, на думку фахівців Gartner, NGFW має забезпечувати [86]: без переривання роботи, вбудовану, bump-in-the-wire (BITW) конфігурацію (BITW – це клас пристроїв зв'язку, які можуть бути вставлені в існуючі (успадковані) системи для підвищення цілісності, конфіденційності або надійності зв'язку існуючим логічним каналом без зміни кінцевих точок зв'язку); стандартні можливості брандмауера першого покоління (наприклад, перетворення мережевих адрес (NAT – network address translation), SPI (stateful packet inspection – перевірка пакетів з відстеженням стану) та віртуальна приватна мережа (VPN) тощо; інтегрований механізм IPS на основі сигнатур; поінформованість про застосунки, повна видимість трафіку і детальний контроль (це означає, що NGFW повинні блокувати або дозволяти пакети в залежності від того, до якого застосунку вони прямують, аналізуючи трафік на рівні 7 еталонної моделі OSI – на рівні застосунків; традиційні МЕ не мають такої можливості, оскільки вони аналізують трафік лише на рівнях 3 та 4 моделі OSI [87]); можливість включати інформацію ззовні брандмауера (наприклад, чорні та білі списки); способи оновлення, щоб мати можливість включати майбутні інформаційні потоки та загрози безпеці; розшифрування SSL для ідентифікації небажаних зашифрованих програм (інспекцію трафіку).

NGFW можуть забезпечувати інтелектуальний аналіз та контроль застосунків, запобігання вторгненням, захист від шкідливих програм та перевірку SSL на мультигігабітних швидкостях, масштабованість для підтримки мереж із найвищою продуктивністю [86]. Порівняно з попередніми поколіннями NGFW мають перевагу в тому, що вони динамічні. Вони можуть використовувати методи машинного навчання для виявлення невідомих раніше загроз шляхом розпізнавання моделей поведінки. Ключовими розробниками цих продуктів експерти називають Fortinet, Palo Alto Networks, Check Point, Cisco та деякі інші [88, 89]. МЕ наступного покоління в даний час відносяться до категорії зрілих рішень. Однак перехід діючих інформаційних систем на хмарні платформи IaaS (Infrastructure-as-a-Service), такі, як Amazon Web Services, Microsoft Azure Google Cloud Platform, змушує задуматися над розширенням можливостей МЕ нового покоління. Таким чином, сьогодні МЕ можуть бути розгорнуті як апаратні пристрої, бути програмними або надаватися як послуга. Апаратний брандмауер – це пристрій, що діє як безпечний шлюз між пристроями всередині периметра мережі та за його межами. Програмний брандмауер або хост-брандмауер працює на сервері або іншому пристрої. Програмне забезпечення хост-брандмауера має бути встановлене на кожному пристрої, що потребує захисту. Хмарний брандмауер – це продукт безпеки, який, як і традиційний брандмауер, фільтрує потенційно шкідливий мережевий трафік. На відміну від традиційних МЕ, міжхмарні хмарні екрани розміщуються в хмарі. Ця хмарна модель брандмауерів також називається брандмауер як послуга (FWaaS – Firewall-as-a-Service) [90]. Хмарні екрани між мережами утворюють віртуальний бар'єр навколо хмарних платформ, інфраструктури та застосунків, так само, як традиційні МЕ утворюють бар'єр навколо внутрішньої мережі організації. Зважаючи на те, що схильність до загроз тієї чи іншої організації різна, для кожного конкретного випадку необхідно шукати відповідне рішення щодо використання МЕ.

Виходячи з сказаного вище, можна зробити висновок, що брандмауери утворюють один із найбільш часто використовуваних механізмів захисту в мережевих системах. При цьому слід пам'ятати, що важливим аспектом також є те, що сам брандмауер має бути надійно захищений від будь-яких загроз безпеці, зокрема він ніколи не повинен виходити з ладу.

Не менш важливо, щоб правила, що наказують, що може проходити, були несуперечливі та встановлювали наміри. Як відомо [91], правильне налаштування МЕ є серйозною проблемою.

Висновки

1. Забезпечення безпеки розподілених інформаційних систем, в яких безліч компонентів може знаходитися в різних місцях і взаємодіяти один з одним за допомогою мережі, що створює вразливості та ризики для безпеки, які не існують у традиційних централізованих системах, є однією з ключових проблем сучасності, що охоплює безліч технологій, методів та процедур, спрямованих на захист систем від різноманітних загроз. Важливість та необхідність її вирішення зростає у зв'язку зі збільшенням обсягу інформації, що зберігається та обробляється в таких системах та передається по мережах зв'язку.

2. Внаслідок шкідливих дій з боку зловмисника, що асоціюються з перехопленням, перериванням, модифікацією та фабрикацією, мають місце такі основні загрози для розподілених систем: несанкціонований доступ до даних, ресурсів і мережі (це може статися, якщо зловмисники отримають доступ до облікових записів або мережевих з'єднань); недостатній захист даних (якщо дані не захищені належним чином, це може призвести до несанкціонованого доступу до них); вразливості у програмному забезпеченні (розподілені системи можуть використовувати велику кількість програмного забезпечення, яке може містити вразливості, які можуть бути використані зловмисниками для атак на систему); неправильна конфігурація системи (може створювати вразливість атак на систему); відмова компонентів системи. Регулярні оцінки безпеки, сканування вразливостей та тестування на проникнення можуть допомогти виявити та усунути слабкі місця у розподілених системах. Особи, які відповідають за безпеку в організаціях, компаніях, повинні завжди бути в курсі актуальних загроз та передових методів забезпечення безпеки, щоб відповідним чином адаптувати існуючі заходи безпеки.

3. Для забезпечення безпеки у розподілених системах доцільно використовувати комплексні заходи, такі як: механізми автентифікації та авторизації (це дозволяє засвідчити легітимність користувачів, процесів та компонентів системи, а також керувати доступом до ресурсів та даних у системі; дані механізми є основою для створення безпечних каналів, що забезпечують безпечну взаємодію користувачів та процесів у системі); механізми контролю доступу та політики безпеки (кожен із ресурсів може підтримувати власний список доступу, в якому перераховуються права доступу всіх користувачів або процесів; крім того, процес може мати сертифікат, який точно встановлює його права на певний набір ресурсів); шифрування даних (як тих, що зберігаються у відповідних сховищах, базах даних системи, так і тих, що передаються через мережу); моніторинг та протоколювання дій / ведення журналу (це дозволяє відстежувати дії користувачів та компонентів системи для виявлення незвичайних або підозрілих активностей); резервне копіювання (регулярне створення резервних копій даних з метою швидкого їх відновлення у разі втрати або ушкодження); регулярне оновлення програмного забезпечення (це допомагає виявляти та виправляти вразливості в системі, які можуть бути використані зловмисниками для атак на систему) та деякі інші. Усі ці заходи повинні бути застосовані відповідно до конкретних вимог та особливостей розподіленої системи, щоб забезпечити надійний захист від різноманітних загроз безпеці, так як навіть єдине слабке місце в системі може призвести до порушення безпеки всієї системи, і зробити заходи захисту її активів, що використовуються, марними. Тому розробка та правильне комплексне застосування цих механізмів для забезпечення ефективного захисту елементів РІС є непростим, нетривіальним завданням, що потребує знання як теоретичних положень, так і кращих практик у галузі інформаційної безпеки та кібербезпеки.

Список літератури:

1. Dhanarani A., Evans R., Loumi H., Lowenthal R., Lopes P., Mesaros M., Schaeumer B., Wahl P., Williams A., Zaidi N. Oracle Database Security a technical primer. Fifth edition. 2023. 160 p.
2. Global Data Protection Index 2022 Key Findings. October 2022. URL: https://www.delltechnologies.com/asset/en-nz/products/data-protection/industry-market/global-data-protection-index-key-findings.pdf.
3. General Data Protection Regulation GDPR. URL: https://gdpr-info.eu/.
4. Заплатинський В. М. Логіко-детермінантні підходи до розуміння поняття «Безпека» // Вісник Кам'янець-Подільського нац. ун-ту ім. Івана Огієнка. Фізичне виховання, спорт і здоров'я людини. Кам'янець-Подільський : Кам'янець-Подільський нац. ун-т ім. Івана Огієнка, 2012. Вип. 5. С. 90–98.
5. Dictionary. URL: https://www.merriam-webster.com/dictionary/security.
6. Whitman M. E., Mattord H. J. Principles of Information Security. 6th ed. Cengage Learning, 2017. 656 p.
7. Stoneburner G. NIST Special Publication 800-33. Underlying Technical Models for Information Technology Security. URL: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-33.pdf.
8. NIST Special Publication 800-66 Revision 1. An Introductory Resource Guide for Implementing the Health Insurance Portability and Accountability Act (HIPAA) Security Rule. October 2008. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-66r1.pdf.
9. NISTIR 8074 Volume 2. Supplemental Information for the Interagency Report on Strategic U.S. Government Engagement in International Standardization to Achieve U.S. Objectives for Cybersecurity. December 2015. URL: http://dx.doi.org/10.6028/NIST.IR.8074v2.
10. ISO/IEC 27000:2018 Information technology. Security techniques. Information security management systems. Overview and vocabulary. URL: https://www.iso.org/standard/73906.html.
11. NIST Special Publication NIST SP 800-66r2 ipd. Implementing the Health Insurance Portability and Accountability Act (HIPAA) Security Rule: A Cybersecurity Resource Guide. July 2022. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-66r2.ipd.pdf.
12. ISO/IEC 15408-1:2022 Information security, cybersecurity and privacy protection – Evaluation criteria for IT security. Part 1: Introduction and general model. URL: https://www.iso.org/obp/ui/ru/#iso:std:iso-iec:15408:1:ed-4:v1:en
13. Tanenbaum A. S., Van Steen M. Distributed systems principles and paradigms. Prentice Hall, 2002. 803 p.
14. Van Steen M., Tanenbaum A. S. Distributed systems. Third edition. Pearson Education, Inc. 2017. 596 p.
15. Avizienis A., Laprie J. C., Randell B. Fundamental concepts of dependability. Department of Computing Science Technical Report Series. University of Newcastle upon Tyne. 2001. 21 p.
16. Laprie J. C. Dependability – Its Attributes, Impairments and Means // Randell B., Laprie J.C., Kopetz H., Littlewood B. (eds) Predictably Dependable Computing Systems. ESPRIT Basic Research Series. Springer, Berlin, Heidelberg. 1995. P. 3-24. https://doi.org/10.1007/978-3-642-79789-7_1
17. Chapple M., Stewart J. M., Gibson D. CISSP Certified Information Systems Security Professional Official Study Guide, 8th ed. Sybex, John Wiley & Sons, Inc.: Indianapolis, Indiana, 2018. 1050 p.
18. Chapple M., Stewart J. M., Gibson D. CISSP: certified information systems security professional official study guide. 9th Edition. Sybex, John Wiley & Sons, Inc.: Indianapolis, Indiana, 2021. 1248 p.
19. Pfleeger C. P., Pfleeger S. L. Security in Computing. 3rd edition. Upper Saddle River, NJ, USA: Prentice Hall, 2002. 746 p.
20. Pfleeger C. P., Pfleeger S. L. Security in Computing. Fifth Edition. Margulies. Prentice Hall. 2015. 944 p.
21. Swanson M., Guttman B. NIST 800-14. Generally Accepted Principles and Practices for Securing Information Technology Systems. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=890092.
22. Committee on National Security Systems (CNSS) Glossary. CNSSI No. 4009. 2022. URL: https://www.niap-ccevs.org/Ref/CNSSI_4009.pdf
23. Priscilla O. Top-Down Network Design. Cisco Press: Indianapolis, IN, USA. 2010. 447 p.
24. NIST Special publication 1800-10. Protecting Information and System Integrity in Industrial Control System Environments: Cybersecurity for the Manufacturing Sector. 2022. https://doi.org/10.6028/NIST.SP.1800-10.
25. FIPS PUB 200. Federal information processing standards publication. Minimum Security Requirements for Federal. Information and Information Systems. 2006. URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.200.pdf.
26. Harini N., Padmanabhan T. R. 2CAuth: A new two factor authentication scheme using QR-code. International Journal of Engineering and Technology. 2013. 5(2). P. 1087–1094.
27. Velásquez I., Caro A., Rodríguez A. Authentication schemes and methods: A systematic literature review // Information and Software Technology. 2018. Vol. 94. P. 30–37. https://doi.org/10.1016/j.infsof.2017.09.012
28. O'Gorman L. Comparing passwords, tokens, and biometrics for user authentication. In Proceedings of the IEEE. 2003. 91(12). P. 2021–2040. https://doi.org/10.1109/JPROC.2003.819611
29. Ometov A., Bezzateev S., Mäkitalo N., Andreev S., Mikkonen T., Koucheryavy Y. Multi-Factor Authentication: A Survey. Cryptography 2018. 2(1). 1. https://doi.org/10.3390/cryptography2010001
30. Auditing Database Activity. URL: https://docs.oracle.com/en/database/oracle/oracle-database/12.2/tdpsg/auditing-database-activity.html#GUID-BF747771-01D1-4BFB-8489-08988E1181F6
31. Gollmann D. Computer Security. 3rd ed. Hoboken, NJ, USA: Wiley, 2011. 436 p.

32. Methods and systems for transparent data encryption and decryption. Richard James McCarty, Austin, TX (US); International Business Machines Corporation, Armonk, NY (US) – N 10/422,667. US Patent 7426,745 B2, 16 September 2008.
33. Stallings W., Brown L. Computer security principles and practice. Fourth Edition. 2018. 778 p.
34. Security management definition. URL: https://www.lawinsider.com/dictionary/security-management.
35. Rose S., Borchert O., Mitchell S., Connelly S. Zero Trust Architecture. Special Publication NIST SP 800-207. 2020. https://doi.org/10.6028/NIST.SP.800-207.
36. Gartner. URL: https://www.gartner.com/en/about.
37. Gartner. Gartner Predicts 10% of Large Enterprises Will Have a Mature and Measurable Zero-Trust Program in Place by 2026. URL: https://www.gartner.com/en/newsroom/press-releases/2023-01-23-gartner-predicts-10-percent-of-large-enterprises-will-have-a-mature-and-measurable-zero-trust-program-in-place-by-2026.
38. Gartner Research. Market Share Analysis: Enterprise Network Equipment, Worldwide, 2022. URL: https://www.gartner.com/en/documents/4412099.
39. Bishop M. Computer Security: Art and Science. Second ed. Addison-Wesley, Reading, MA., 2019. 1383 p.
40. Bouch A. 3-D Secure: A critical review of 3-D Secure and its effectiveness in preventing card not present fraud. University of London. 2011. URL: https://www.58bits.com/thesis/3-D_Secure.pdf.
41. NIST Special Publication 1800-21. Mobile Device Security: Corporate-Owned Personally-Enabled (COPE). URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1800-21.pdf.
42. Google Cloud documentation. Encryption in transit. URL:https://cloud.google.com/docs/security/encryption-in-transit.
43. Sommerhalder M. Hardware Security Module. In: Mulder V., Mermoud A., Lenders V., Tellenbach B. (eds) Trends in Data Protection and Encryption Technologies. Springer, Cham. 2023. P. 83–87.
44. Google Cloud. Default encryption at rest. URL: https://cloud.google.com/docs/security/encryption/default-encryption
45. Єсін В. І., Вілігура В. В. Основні категорії NewSQL баз даних та їх особливості // Радіотехніка. 2022. № 211. С. 37–66. https://doi.org/10.30837/rt.2022.4.211.03.
46. Sarbanes-Oxley Act of 2002. Public Law 107–204, Approved July 30, 2002, 116 Stat. 745. URL: https://www.govinfo.gov/content/pkg/COMPS-1883/pdf/COMPS-1883.pdf
47. Scholl M., Stine K., Hash J., Bowen P., Johnson A., et al. NIST Special Publication 800-66 Revision 1. An Introductory Resource Guide for Implementing the Health Insurance Portability and Accountability Act (HIPAA) Security Rule. 2008. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-66r1.pdf
48. Payment Card Industry (PCI) Data Security Standard. Requirements and Testing Procedures Version 4.0. 2022. URL: https://www.pcisecuritystandards.org/documents/PCI-DSS-v4_0.pdf
49. Nanda A., Feuerstein S. Oracle PL/SQL for DBAs: Security, Scheduling, Performance & More. O'Reilly Media, Inc., 2005. 454 p.
50. Advanced Security Guide. Introduction to Transparent Data Encryption. URL: https://docs.oracle.com/en/database/oracle/oracle-database/23/asoag/introduction-to-transparent-data-encryption.html#GUID-62AA9447-FDCD-4A4C-B563-32DE04D55952
51. Barker E., Kelsey J. NIST Special Publication 800-90A Revision 1. Recommendation for Random Number Generation Using Deterministic Random Bit Generators. 2015. http://dx.doi.org/10.6028/NIST.SP.800-90Ar1.
52. Needham R. M., Schroeder M. D. Using encryption for authentication in large networks of computers. Communications of the ACM. 1978. 21(12). P. 993–999. https://doi.org/10.1145/359657.359659
53. FIPS 186-5. Federal information processing standards publication (Supersedes FIPS 186-4). Digital Signature Standard (DSS). Category: computer security. Subcategory: cryptography. 2023. https://doi.org/10.6028/NIST.FIPS.186-5
54. Barker E., Chen L., Roginsky A., Vassilev A., Davis R., Simon S. NIST Special Publication 800-56B Revision 2. Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography. 2019. https://doi.org/10.6028/NIST.SP.800-56Br2
55. Cruz-Cunha M. M., Oliveira E. F., Tavares A. J., Ferreira, L. G. Handbook of research on social dimensions of semantic technologies and web services. Hershey, PA: IGI Global, 2009. 1180 p.
56. Lampson B. W. Protection. ACM SIGOPS Operating Systems Review. 1974. 8(1). P. 18–24.
57. Lampson B. W. Dynamic protection structures. Proceedings of the November 18-20, 1969, fall joint computer conference. 1969. P. 27–38.
58. Graham G. S., Denning P. J. Protection: principles and practice. Proceedings of the May 16-18, 1972, spring joint computer conference. 1971. P. 417–429.
59. Harrison M. A., Ruzzo W. L., Ullman J. D. Protection in Operating Systems // Communications of the ACM, 1976. 19(8). P. 461–471.
60. Weissman C. Security controls in the ADEPT-50 time-sharing system // Proceedings of the November 18-20, 1969, fall joint computer conference. 1969. P. 119–133.
61. Hartson H. R., Hsiao D. K. A Semantic Model for Database Protection Languages // Proceedings of the second international conference on Systems for Large Data Bases. 1976. P. 27–42.
62. Lipton R. J., Snyder L. A linear time algorithm for deciding subject security // Journal of the ACM (JACM). 1977. 24(3). P. 455–464.

63. Hoffman L.J. Modern Methods for Computer Security and Privacy. Englewood Cliffs, NJ, USA: Prentice-Hall. Inc., 1977. 268 p.
64. Білігура В. В. Аналіз формальних моделей управління доступом і особливості їх застосовності для баз даних // Радіотехніка. 2021. Вип. 205. С. 53–70. https://doi.org/10.30837/rt.2021.2.205.05.
65. Білігура В. В., Горбенко Ю. І., Єсін В. І., Рассомахін С. Г. Використання формальних моделей безпеки в захищених базах даних // Фізико-математичне моделювання та інформаційні технології. 2021. № 32. С. 70–74. https://doi.org/10.15407/fmmit2021.32.070
66. Bell D. E., LaPadula L. J. Secure Computer Systems: Unified Exposition and Multics Interpretation (MTR-2997 Rev. 1). Bedford, Mass.: MITRE Corp., 1976. 129 p.
67. Sandhu R.S., Coyne E. J., Feinstein H. L., Youman C. E. Role-based access control models // IEEE Computer. 1996. № 2. P. 38–47.
68. NIST. Attribute Based Access Control. URL: https://csrc.nist.gov/Projects/Attribute-Based-Access-Control.
69. Hu V. C., Ferraiolo D., Kuhn R. Guide to Attribute Based Access Control (ABAC) Definition and Considerations. NIST Special Publication 800-162. 2014. URL: https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.sp.800-162.pdf.
70. Hu V. C., Kuhn D. R., Ferraiolo D. F., Voas J. Attribute-Based Access Control // Computer. 2015. Vol. 48. No. 2. P. 85–88. https://doi.org/10.1109/MC.2015.33.
71. Servos D., Osborn S.L. HGABAC: Towards a Formal Model of Hierarchical Attribute-Based Access Control // Cuppens, F., Garcia-Alfaro, J., Zincir Heywood, N., Fong, P. (eds) Foundations and Practice of Security. FPS 2014. Lecture Notes in Computer Science. Springer, Cham. 2015. Vol. 8930. P. 187–204.
72. Biswas P., Sandhu R., Krishnan R. Label-based access control: An ABAC model with enumerated authorization policy // Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control (ABAC '16). Association for Computing Machinery, New York, NY, USA. 2016. P. 1–12. https://doi.org/10.1145/2875491.2875498.
73. Servos D., Osborn S. L. Current research and open problems in attribute-based access control // ACM Computing Surveys (CSUR). 2017. 49(4). P. 1–45. https://doi.org/10.1145/3007204.
74. OASIS eXtensible Access Control Markup Language (XACML) TC. URL: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml
75. International Committee for Information Technology Standards, Information technology – Next Generation Access Control – Functional Architecture (NGAC–FA), ANSI/INCITS 499-2018, American National Standards Institute, New York, January 30, 2018. 57 p.
76. INCITS 565-2020. Information technology – Next Generation Access Control (NGAC). American National Standard for Information Technology. April, 2020.
77. Ferraiolo D., Chandramouli R., Hu V., Kuhn R. A Comparison of Attribute Based Access Control (ABAC) Standards for Data Service Applications. NIST Special Publication 800-178. 2016. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-178.pdf. http://dx.doi.org/10.6028/NIST.SP.800-178.
78. Hu V. C., Ferraiolo D. F, Chandramouli R., D. Kuhn D. R. Attribute-Based Access Control. Artech House. 2017. 280 p.
79. Shirey R. Internet Security Glossary. Version 2. 2007. №. rfc4949. URL: https://datatracker.ietf.org/doc/html/rfc4949.
80. Stouffer K., Pease M., Tang C.Y., Zimmerman T., Pillitteri V., Lightman S., Hahn A., Saravia S., Sherule A., Thompson M. NIST Special Publication NIST SP 800-82r3. Guide to Operational Technology (OT) Security. 2023. https://doi.org/10.6028/NIST.SP.800-82r3.
81. Scarfone K., Hoffman P. Special Publication 800-41 Revision 1. Guidelines on Firewalls and Firewall Policy. 2009. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-41r1.pdf.
82. Cheswick W. R. Bellovin S. M., Rubin A. D. Firewalls And Internet Security: Repelling The Wily Hacker. 2nd ed. Addison-Wesley Professional. 2003. 464 p.
83. Mukkamala P. P., Rajendran S. A survey on the different firewall technologies // International Journal of Engineering Applied Sciences and Technology. 2020. 5(1). P. 363–365.
84. Goralski W. The illustrated network: how TCP/IP works in a modern network. Morgan Kaufmann. 2017. 899 p.
85. Next-generation Firewalls (NGFWs). URL: https://www.gartner.com/en/information-technology/glossary/next-generation-firewalls-ngfws.
86. Malecki F. Next-generation firewalls: Security with performance // Network Security. 2012. Vol. 2012. Issue 12. P. 19–20.
87. What is a next-generation firewall (NGFW)? URL: https://www.cloudflare.com/learning/security/what-is-next-generation-firewall-ngfw/.
88. What are Network Firewalls? URL: https://www.gartner.com/reviews/market/network-firewalls.
89. A Leader Positioned Highest in Ability to Execute. URL: https://www.fortinet.com/solutions/gartner-network-firewalls.
90. What is a cloud firewall? URL: https://www.cloudflare.com/learning/cloud/what-is-a-cloud-firewall/.
91. Wool A. Trends in firewall configuration errors: Measuring the holes in Swiss cheese // IEEE Internet Computing. 2010. 14(4). P. 58–65. https://doi.org/10.1109/MIC.2010.2910.1109/MIC.2010.29.

Надійшла до редколегії 05.09.2023

Відомості про авторів:

Есін Віталій Іванович – д-р техн. наук, Харківський національний університет імені В.Н. Каразіна, професор кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук; Україна; e-mail: v.i.yesin@karazin.ua; ORCID: https://orcid.org/0000-0003-1977-7269

Вілігура Владислав Вікторович – Харківський національний університет імені В.Н. Каразіна, аспірант кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук; Україна; e-mail: viligura93@gmail.com; ORCID: https://orcid.org/0000-0002-1137-2382

Сватовський Ігор Іванович – канд. техн. наук, Харківський національний університет імені В. Н. Каразіна, старший науковий співробітник, доцент кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук; Україна; e-mail: i.svatowsky@karazin.ua; ORCID: https://orcid.org/0000-0002-1836-5599