https://orcid.org/0000-0002-5080-1856

УДК 004.056.5

Ю.М. ЛИСЕЦЬКИЙ*

КЕРУВАННЯ ВРАЗЛИВОСТЯМИ ЯК ОРГАНІЗАЦІЙНО-ТЕХНІЧНИЙ ПРОЦЕС

*ТОВ «ЕС ЕН ТІ УКРАЇНА», м. Київ, Україна

Анотація. *Керування вразливостями — це організаційно-технічний процес і набір практик для виявлення, оцінки, пріоритезації, усунення і валідації вразливостей у програмному і апаратному забезпеченні в IT-інфраструктурі та системах промислової автоматизації, які використовують організації і установи. Поява та розвиток процесу керування вразливостями нерозривно пов'язані із самою історією інформаційно-телекомунікаційних технологій. Епохальною подією у цій галузі була поява у 1999 році Common Vulnerabilities and Exposures (CVE) — загальнодоступного реєстру ідентифікованих вразливостей, що стало фундаментом для уніфікації назв, обміну інформацією й інструментального системного застосування сканерів, бази даних, експлойт-фідерів. Наступним суттєвим кроком вперед стала розробка Common Vulnerability Scoring System — системи порівняння вразливостей на основі загальних метрик впливу та ймовірності експлуатації. Ці здобутки радикально підвищили масштаби й можливості автоматизації для інструментів VM. У загальному процесі VM ключову роль відіграють сканування як мережеве, так і кінцевих точок; збір контексту щодо IT-середовища; математичні/статистичні моделі пріоритезації знахідок; операційні процеси реагування, як-от планування патчів або відкат за потреби. Повторюваний цикл VM можна розглядати як безперервний ланцюжок: виявлення → нормалізація знахідок → пріоритезація знахідок → усунення недоліків → верифікація → звітність. Впровадження CVE було видатним кроком вперед в історії керування вразливостями в IT-системах. Сканери вразливостей стали базовим інструментом не лише VM, а й кібербезпеки взагалі. Вони залишаються необхідним, але не достатнім інструментом кіберзахисту. CVE добре працює у виявленні відомих проблем, однак їхні обмеження вимагають доповнення іншими методами: контекстним аналізом ризиків, інтеграцією з DevSecOps-процесами, використанням прогнозних моделей та аналітики загроз. У майбутньому спостерігатиметься перехід до інтелектуального управління вразливостями, де сканери виконуватимуть роль лише одного з модулів у ширшій екосистемі кіберзахисту.*

Ключові слова: *керування вразливостями, процес, сканери вразливостей, IT-системи, виклики, обмеження.*

Abstract. *Vulnerability management is an organizational and technical process and a set of practices for identifying, assessing, prioritizing, addressing, and validating vulnerabilities in the software and hardware of IT infrastructure and industrial automation systems used by organizations and institutions. The emergence and development of the vulnerability management process are inextricably linked to the history of information and telecommunications technologies. A landmark event in this field was the launch in 1999 of Common Vulnerabilities and Exposures (CVE), a publicly accessible registry of identified vulnerabilities, which laid the foundation for the standardization of names, information exchange, and the systematic use of scanners, databases, and exploit feeders. The next significant step forward was the development of the Common Vulnerability Scoring System, i.e. a system for comparing vulnerabilities based on common metrics of impact and exploitation probability. These achievements drastically expanded the scope and automation capabilities of VM tools. In the overall VM process, scanning — both network and endpoint — plays a key role, as does gathering context about the IT environment, mathematical/statistical models for prioritizing findings, and operational response processes, such as patch planning or rollback when necessary. The iterative VM cycle can be viewed as a continuous chain: detection → normalization of findings → prioritization of findings → remediation → verification → reporting. The introduction of*

ISSN 1028-9763. Математичні машини і системи. 2026. № 2

CVE marked a significant step forward in the history of vulnerability management in IT systems. Vulnerability scanners have become a fundamental tool not only for VM but for cybersecurity in general. They remain an essential, yet insufficient, component for cyber defense. CVE works well for identifying known issues but its limitations require supplementation with other methods: contextual risk analysis, integration with DevSecOps processes, and the use of predictive models and threat analytics. In the future, the trend is moving toward intelligent vulnerability management where scanners become just one module in a broader cybersecurity ecosystem.
Keywords: *vulnerability management, process, vulnerability scanners, IT systems, challenges, limitations.*

DOI: 10.34121/1028-9763-2026-2-38-45

1. Вступ

Поява та розвиток процесу керування вразливостями в IT-системах нерозривно пов'язані із самою історією інформаційно-телекомунікаційних технологій. З 1970-х рр. та до кінця 1990-х питання безпеки у персональних і корпоративних мережах вирішувались здебільшого безсистемно: окремі адміністратори патчили системи, виправлення розповсюджувалися вручну або через вузькі корпоративні канали. Інструменти автоматизованого сканування та централізованого управління були експериментальними, а стандарти, по суті, відсутні.

Епохальною подією у цій галузі була поява CVE (Common Vulnerabilities and Exposures) у 1999 році [1]. Це загальнодоступний реєстр ідентифікованих вразливостей, що став фундаментом для уніфікації назв, обміну інформацією й інструментального системного застосування (сканери, бази даних, експлойт-фідери). Завдяки кодифікації вразливостей CVE надала «єдину мову», необхідну для кооперації в рамках загального процесу Vulnerability Management (VM).

Поява реєстру CVE відкрила шлях для стрімкого розвитку напряму IT-послуг, оскільки з'явилась організаційна можливість забезпечити автоматизоване виявлення вразливостей для широкого кола державних та приватних організацій. На зламі тисячоліть почали з'являтись різноманітні комерційні продукти, зокрема, як-от Nessus, який наразі став де-факто галузевим стандартом [2], і сервіс-моделі, а також формуватися перші практики централізованого управління базами вразливостей в окремих організаціях.

Наступним суттєвим кроком вперед стала розробка CVSS (Common Vulnerability Scoring System), або система порівняння вразливостей на основі загальних метрик впливу та ймовірності експлуатації. Паралельно такі агрегатори даних щодо вразливостей, як-от NVD (National Vulnerability Database), стали ключовими репозитаріями, що забезпечували метадані для VM-інструментів, які можуть зчитуватись машинами. Ці здобутки радикально підвищили масштаби й можливості автоматизації для інструментів VM [3].

У минулому десятиріччі, у зв'язку зі стрімким розвитком цифрової інфраструктури веб- та мобільних додатків, на фоні з розвитком хмарних моделей організації IT-систем взагалі та особливо процесів розробки програмного забезпечення (ПЗ), зокрема контейнеризації, на перший план вийшла інтеграція практики VM у процеси розробки. Сканування коду під час розробки та релізів ПЗ призвели до подальшого розвитку методології керування вразливостями, яка тепер вміщувала не тільки базове сканування та ранжування на основі CVEE, але й пріоритезацію на основі експлуатаційних спостережень. На новий рівень вийшла також автоматизація VM. Нормальною практикою стало автоматичне створення завдань для IT-фахівців за результатами чергового автоматичного сканування тієї чи іншої IT-системи.

Починаючи з 20-х рр., акцент у розвитку практик VM змістився на подальше підвищення практичності оцінювання вразливостей. Через стрімке зростання масштабу та складності IT-ландшафту звичайної системи CVE стало недостатньо для ефективного керування вразливостями, та у відповідь на це почали виникати так звані «ймовірнісні моделі» (*probabilistic models),* наприклад, KVE (*Known Exploited Vulnerabilities)*, яка дозволила реа-

лізувати на практиці автоматизацію оцінювання знайдених вразливостей щодо ймовірності їх експлуатації у реальних сценаріях, тобто створення переліків вразливостей на першочергове виправлення [4]. Це, безумовно, поліпшило ситуацію, але не вирішило усіх існуючих проблем.

Незважаючи на достатньо велику кількість публікацій, присвячену питанням керування вразливостями та інструментальним засобам його забезпечення, дослідженню керування вразливостями IT-систем як організаційно-технічного процесу та його складових приділено недостатньо уваги.

Мета статті — дослідити керування вразливостями як організаційно-технічний процес та його складові.

2. Керування вразливостями

Керування вразливостями або VM — це організаційно-технічний процес і набір практик для виявлення, оцінки, пріоритезації, усунення і валідації вразливостей у програмному й апаратному забезпеченні в IT-інфраструктурі та системах промислової автоматизації, які використовують державні та комерційні установи. Під вразливістю ми розуміємо помилку або слабкість у дизайні, реалізації чи управлінні інформаційної системи**,** яку може експлуатувати зловмисник і що може призвести до порушення конфіденційності, цілісності або доступності системи [5].

У загальному процесі VM ключову роль відіграє сканування (Vulnerability Scanning, VS) як мережеве, так і кінцевих точок; збір контексту щодо IT-середовища; математичні/статистичні моделі пріоритезації знахідок; операційні процеси реагування, якот планування патчів або відкат за потреби. Повторюваний цикл VM можна розглядати як безперервний ланцюжок (рис. 1).

Рисунок 1 — Повторюваний цикл VM

Цей підхід відповідає сучасним настановам з патч-менеджменту і технічних оцінок [6, 7].

Багато нефахівців, та навіть деякі фахівці, часто утотожнюють процеси керування вразливостями та сканування на вразливості, але це не є коректним. Якщо перше — це загальний процес, то сканування — лише техніка з першої стадії «виявлення». Сканер сам по собі видає сигнали (alerts), які треба нормалізувати, верифікувати та прив’язати до контексту, інакше численні помилково-позитивні або неактуальні знахідки перевантажують операції і знижують ефективність. Сканування — необхідне, але недостатнє; VM — це система процесів, людей і даних, що перетворює результати сканерів на відтворювані бізнес-рішення [8]. Отже, сканування є ядром процесу VM: воно забезпечує систематичний пошук відомих слабких місць у програмному забезпеченні, конфігураціях і мережевих сервісах. На рівні практики існують три основні підходи:

- мережеве сканування (remote network-based);
- агентське сканування (host-based);
- аналіз прикладного рівня та вебдодатків (DAST — Dynamic Application Security Testing, SAST — Static Application Security Testing, IAST — Interactive Application Security Testing).

Результати сканування мають пройти верифікацію, фільтрацію та пріоритезацію. Без цього може виникнути парадокс коли організація отримує десятки тисяч знахідок, з яких

значна частина є неактуальною або помилковою. Це веде до таких психологічних факторів, як-от перевантаження інформацією (alert fatigue), втрата довіри до інструментів і зниження швидкості реагування [9].

3. Виклики VM

Розриви покриття

Важливість та складність організації процесу VM у сьогоденних IT-середовищах важко переоцінити, перш за все, через постійно зростаючу складність та гетерогенність IT-систем: у сучасних організаціях вони дедалі рідше є монолітними чи статичними. Натомість вони розвиваються як динамічні екосистеми, що містять локальні дата-центри, хмарні сервіси, гібридні архітектури, контейнеризовані мікросервіси, а також IoT (Internet of Things) або OT (Ooperational Technology). Така гетерогенність створює серйозні виклики для систем VM, оскільки вони стикаються з так званими «розривами покриття» (coverage gaps), коли сканери вразливостей або процеси управління не виявляють, не враховують чи не встигають обробити частину потенційних точок ризику.

Серед основних причин існування «розривів покриття» слід згадати такі:

1. Різнорідність технологій: у складних IT-середовищах поєднуються застарілі системи (від англ.: legacy), сучасні сервери, мобільні пристрої, віртуальні машини, контейнери та IoT. Багато вразливостей є специфічними для окремих платформ, і традиційні сканери не завжди мають необхідні плагіни чи сигнатури [10].

2. Динаміка хмарних середовищ та контейнерів: широко вживані сьогодні контейнери та оркестратори (Docker, Kubernetes) створюють IT-середовища з високою швидкістю змін. Активи можуть існувати лічені хвилини, перш ніж зникнути або бути замінені. Це ускладнює повноцінне покриття та вчасне виявлення вразливостей [11], які можуть бути використані зловмисниками.

3. Сліпі зони у мережі: в умовах розповсюдження сегментації мережі та багаторівневої віртуалізації сканери можуть не бачити певних підмереж або закритих середовищ, особливо в хмарах чи середовищах з обмеженим доступом [12].

4. Людський фактор і процесні прогалини: керування вразливостями передбачає інтеграцію технічних і організаційних процесів, але на практиці ці процеси навіть сьогодні постійно стикаються з неповними або неточними інвентаризаціями активів, відсутністю оновлених політик, а також із таким психологічним чинником, як-то перевантаження інформацією. У комплексі це постійно призводить до того, що навіть відомі вразливості залишаються невиправленими [13].

Наслідками цих факторів стає те, що у IT-керівництва організацій може з’являтись хибне відчуття безпеки, коли є впевненість, що покриття повне, хоча насправді критичні активи залишаються поза увагою. Також існує велика небезпека асиметричних ризиків, коли розриви покриття з’являються саме у найбільш динамічних зонах – хмарних середовищах, мобільних додатках, IoT, які одночасно є й найпривабливішими для атак. У комплексі все це призводить то системної вразливості, коли навіть один «невидимий» актив може стати точкою входу для атаки з ефектом доміно.

Інакше кажучи, існують рішення для всіх цих проблем, зокрема, інвентаризація IT-активів у режимі реального часу, агентське та безагентське сканування в комбінації, що дозволяє покривати як постійні активи, так і швидкоплинні середовища; адаптивне управління ризиками, де пріоритети визначаються на основі контексту бізнес-активів, а не лише CVSS-рейтингу; ширше використання людиноцентричних методів: психоедукація фахівців із кібербезпеки, управління когнітивним навантаженням, автоматизація первинного аналізу результатів сканувань.

Хибнопозитивні і хибнонегативні результати

Хибнопозитивні (false positives або FP) результати сканування виникають, коли сканер повідомляє про вразливість там, де її немає. Серед найпоширеніших причин появи хибнопозитивних результатів слід згадати евристичний характер перевірок, коли, наприклад, результат сканування схожий на вразливу версію, але патч уже встановлено; обмежена видимість, наприклад, через блокування фаєрволом або нестандартні конфігурації IT-середовища, або недостатня актуальність сигнатур. Знахідки типу FP підривають довіру до самого процесу керування вразливостями [14]. Хибнонегативні (false negatives або FN) результати ще небезпечніші, бо IT-персонал організації може бути впевненим, що її IT-системи є безпечними, хоча насправді вони під ризиком. Серед основних причин виникнення FN можна назвати використання нових або 0-day вразливостей, відсутніх у базах CVE; наявність складних IT-середовищ, наприклад, із контейнерами або мікросервісами; просунуті обхідні техніки зловмисників, які змінюють сигнатури або структуру атак.

Основними підходами до зниження кількості FN/FN є:

• комбіноване використання інструментів, насамперед сканерів, коли кожен із них дає різне покриття, та агрегація результатів знижує кількість FN;

• кореляція з іншими джерелами, як-от дані KEV/EPSS, бази даних відомих експлойтів вразливостей, що дозволяє відсіяти FP та підсвітити найбільш релевантні CVE;

• валідація знахідок через практичні тести, насамперед через тести на проникнення, які пропонують найвищу достовірність щодо знахідок, але вимагають суттєвих витрат;

• автоматизація контекстуалізації, коли використання різних технік та інструментів інвентаризації програмних та апаратних IT-активів дозволяє відфільтровувати нерелевантні CVE.

4. Чому CVE було недостатньо?

Як ми вже згадували вище, хоча впровадження CVE було видатним кроком вперед в історії керування вразливостями в IT-системах, цього виявилось недостатньо. Практика багатьох років продемонструвала, що насправді експлуатується лише невелика частка CVE, через що підхід «патчити все критичне» виявився недостатньо ефективним. Чому так сталося? Насправді питання, «які вразливості виправляти першими», виявилось одним із найкритичніших у системах керування вразливостями. І протягом тривалого часу організації спиралися на систему CVSS як основний інструмент оцінки критичності вразливостей. Простий підхід «виправляємо все, що має CVSS $\geq$ 7.0» або «все критичне», здавався ефективним, але на практиці призводив до перевантаження фахівців, оскільки щороку виявляються десятки тисяч нових CVE. Поглиблений статистичний аналіз із метою вирішення цієї проблеми показав, що CVSS як ізольований показник не відображає реальної експлуатаційності вразливості та контексту бізнес-ризиків. Хоча CVSS надає числове значення критичності на основі базових, часових і контекстних метрик [15], він не відображає динаміку загроз. Це створювало перекіс: багато вразливостей із CVSS 9.0+ ніколи не експлуатувалося, тоді як експлойти «середніх» уразливостей активно використовувалися зловмисниками. Саме це призвело до еволюції від «CVSS-порогів» до ризик-орієнтованої пріоритезації, що враховує ймовірність експлуатації, активність атак, контекст активу та потенційний вплив на бізнес.

Подальшим кроком стало створення EPSS (Exploit Prediction Scoring System) — відкритої моделі, що прогнозує ймовірність експлуатації вразливості на основі статистики CVE, експлойтів і поведінки атакувальників [16].

Це стало поворотним моментом, оскільки організації отримали змогу ранжувати не лише те, «наскільки небезпечна вразливість за CVSS», а й те, «наскільки ймовірно, що вона буде використана найближчим часом».

Завдяки цим змінам сучасна пріоритезація знахідок у VM є ризик-орієнтованою й орієнтується на багатофакторні моделі. Серед них основними є:

• експлуатаційність (EPSS): чи існують публічні експлойти, чи зафіксовано активні атаки;

• контекст IT-активу: критичність для бізнесу, чутливість даних, роль в архітектурі;

• динамічні фактори: наявність патчів, складність виправлення, можливість застосування компенсуючих контролів;

• прогнозні моделі: використання машинного навчання для прогнозування появи експлойтів або масштабності атак [17].

Цей підхід дає організаціям можливість приділити першочергову увагу не всім CVSS $\geq$ 9.0, а тим уразливостям, які мають високу ймовірність експлуатації, стосуються бізнес-критичних активів та вже потрапили до списків активно експлуатованих (наприклад, CISA Known Exploited Vulnerabilities).

5. Сканери вразливостей як ключовий компонент VM

Сканери вразливостей стали базовим інструментом не лише VM, а й кібербезпеки взагалі ще з початку 2000-х років, коли з'явилися продукти, як-от Nessus, OpenVAS чи Qualys [18]. Вони дозволили автоматизувати виявлення відомих вразливостей у мережах, операційних системах та вебдодатках. Проте розвиток технологій, перехід до хмарних і контейнерних середовищ, а також зростання кількості вразливостей висвітлили низку обмежень, які сьогодні вважаються критичними:

1. Хибнопозитивні та хибнонегативні результати: жоден сканер не забезпечує повної точності. Хибнопозитивні призводять до перевантаження аналітиків і зниження довіри до результатів, тоді як хибнонегативні створюють небезпечні «сліпі зони» [19].

2. Залежність від сигнатур і баз CVE: більшість сканерів працюють за принципом пошуку відомих патернів. Це означає, що нові чи zero-day вразливості залишаються поза увагою до моменту оновлення бази [20].

3. Обмежене охоплення сучасних технологій: контейнеризовані середовища, мікросервіси, безсерверні функції (serverless) часто не мають достатнього рівня інтеграції зі сканерами. Це ускладнює повноцінний моніторинг у DevOps-середовищах.

4. Проблеми масштабування: у великих організаціях із тисячами активів регулярні повні скани створюють значне навантаження на мережу, що може впливати на продуктивність бізнес-систем [22].

5. Відсутність контексту ризику: сканери виявляють вразливості, але не враховують бізнес-критичність активу, на якому знайдено проблему. Це знижує цінність результатів для прийняття рішень на рівні CISO [23].

6. Обмежений аналіз експлуатаційності: хоча деякі сучасні рішення інтегрують дані EPSS чи threat intelligence, більшість класичних сканерів не прогнозують імовірність реальної експлуатації [24].

На сьогодні розроблено цілу низку підходів до зменшення впливу цих обмежень, серед яких можна згадати таке:

• поєднання агентських і безагентських методів: агентське сканування дозволяє отримувати точніші дані без перевантаження мережі, тоді як безагентське забезпечує ширше охоплення. Комбінація двох підходів зменшує прогалини;

• інтеграція з контекстними системами: зв'язок сканерів із SIEM та системами управління IT-активами дозволяє враховувати бізнес-критичність вразливостей;

• DevSecOps-процеси: інтеграція сканування в CI/CD пайплайни та інфраструктуру як код дозволяє виявляти проблеми ще до етапу розгортання [25];

• використання threat intelligence: додавання даних із CISA KEV, експлойт-баз і EPSS допомагає відсіяти «шум» та фокусуватися на реально небезпечних уразливостях;

• людиноцентричні практики: автоматизація сортування знахідок, навчання аналітиків та зменшення інформаційного перевантаження підвищують ефективність роботи з результатами сканування.

6. Висновки

Отже, керування вразливостями — це організаційно-технічний процес і набір практик для виявлення, оцінки, пріоритезації, усунення і валідації вразливостей у програмному і апаратному забезпеченні IT-систем для підвищення рівня кіберзахисту IT-інфраструктури організацій. Сканери вразливостей залишаються необхідним, але не достатнім інструментом кіберзахисту. Вони добре працюють у сфері виявлення відомих проблем, однак їхні обмеження вимагають доповнення іншими методами: контекстним аналізом ризиків, інтеграцією з DevSecOps-процесами, використанням прогнозних моделей та аналітики загроз. У майбутньому тенденція рухається до інтелектуального управління вразливостями, де сканери стають лише одним із модулів у ширшій екосистемі кіберзахисту.

СПИСОК ДЖЕРЕЛ

1. Holm H., Sommestad T., Almroth J., Persson M. A quantitative evaluation of vulnerability scanning. *Information Management & Computer Security.* 2011. Vol. 19 (4). P. 231–247.
2. Scanning Large Networks with Nessus. URL: https://www.tenable.com/blog/scanning-large-networks-with-nessus (дата звернення: 07.01.2026).
3. Karlsson M. The Edit History of the National Vulnerability Database and similar Vulnerability Databases. 2012. 100 p.
4. Лисецький Ю.М., Старовойтенко О.О. Керування вразливостями IT-систем. *Вісник воєнної розвідки.* 2026. № 92. С. 29–33.
5. Brumă O.-V. Vulnerabilities of Information Systems. *International Journal of Information Security and Cybercrime*. 2020. Vol. 9 (1). P. 9–14.
6. Mell P., Scarfone K., Romanosky S. A proposed metric for vulnerability exploitation probability. NIST Computer Security White Paper. National Institute of Standards and Technology. 2025. 34 p.
7. Souppaya M., Scarfone K. *NIST Special Publication 800-40 Revision 4: Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology*. National Institute of Standards and Technology. URL: https://csrc.nist.gov/pubs/sp/800/40/r4/final. NIST Computer Security Resource Center+1 (дата звернення: 10.01.2026).
8. Bennouk K., Ait Aali N., El Bouzekri El Idrissi Y., Sebai B., Faroukhi A.Z., Mahouachi D. A Comprehensive Review and Assessment of Cybersecurity Vulnerability Detection Methodologies. *Journal of Cybersecurity and Privacy*. 2024. Vol. 4 (4). P. 853–908.
9. History. URL: https://www.cve.org/about/history (дата звернення: 12.01.2026).
10. Li Z., Sun L., Xu W., Chi C.H., Xue Y. Vulnerability scanning and analysis in large-scale heterogeneous systems. *Journal of Computer Security.* 2021. Vol. 29 (2). P. 135–162.
11. Shu R., Gu X., Enck W. A study of security vulnerabilities on Docker Hub. *Proc. of the Seventh ACM Conference on Data and Application Security and Privacy.* 2017. P. 269–280.
12. Khan R., McLaughlin K., Laverty D., Sezer S. STRIDE-based threat modeling for cyber-physical systems. *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. 2020. P. 1–6.
13. Fruhlinger J. What is vulnerability management? A process for securing software and networks. *CSO Online*. URL: https://www.csoonline.com/ (дата звернення: 14.01.2026).
14. Doupe A., Cova M., Vigna G. Why Johnny can't pentest: An analysis of black-box web vulnerability scanners. *Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2010)*. 2010. P. 111–131.

15. Mell P., Scarfone K., Romanosky S. A complete guide to the Common Vulnerability Scoring System version 2.0. *Forum of Incident Response and Security Teams (FIRST)*. 2007. 23 p.
16. Jacobs J., Spring J., Stoner E., Sauerwein C. Exploit Prediction Scoring System (EPSS). *Proc. of the 2021 IEEE European Symposium on Security and Privacy Workshops*. 2021. P. 587–596.
17. Sabottke C., Suciu O., Dumitraş T. Vulnerability disclosure in the age of social media: Exploiting Twitter for predicting real-world exploits. *24th USENIX Security Symposium (USENIX Security 15)*. 2015. P. 1041–1056.
18. Mell P., Scarfone K., Romanosky S. A Complete Guide to the Common Vulnerability Scoring System Version 2.0. *FIRST-Forum of Incident Response and Security Teams*. 2007. P. 1–23.
19. Gupta S., Badve O., Vora P. Automated vulnerability assessment and penetration testing using Nessus. *International Journal of Computer Applications.* 2020. Vol. 176 (38). P. 1–7.
20. Fang Z., Zhang L., Chen Y., Chen Z. Research on vulnerability detection technology in software security. *IEEE Access.* 2020. Vol. 8. 2020. P. 586–600.
21. Shu R., Gu X., & Enck, W. A study of security vulnerabilities on Docker Hub. *Proc. of the Seventh ACM Conference on Data and Application Security and Privacy*. 2017. P. 269–280.
22. Li Z., Sun L., Xu W., Chi C.H., Xue Y. Vulnerability scanning and analysis in large-scale heterogeneous systems. *Journal of Computer Security.* 2021. Vol. 29 (2). P. 135–162.
23. Zhang Y., Chen X., Xiang Y. Context-aware risk-based vulnerability management. *Future Generation Computer Systems.* 2019. Vol. 94. P. 444–456.
24. Jacobs J., Spring J., Stoner E., Sauerwein C. Exploit Prediction Scoring System (EPSS). *Proc. of the 2021 IEEE European Symposium on Security and Privacy Workshops*. 2021. P. 587–596.
25. Rahman M.S., Williams L., Bradshaw G. A framework for improving security patch management in DevOps. *Proc. of the 2019 International Conference on Software and System Processes*. 2019. P. 134–143.

Стаття надійшла до редакції 24.02.2026 / прийнята до друку 28.04.2026