Keywords: masking, privacy, mathematical models, Shadowsocks.

Клімович Сергій Олегович, кандидат технічних наук, начальник кафедри захисту інформації в телекомунікаційних системах та мережах Військового інституту телекомунікацій та інформатизації імені Героїв Крут.

Serhii Klimovych, candidate of technical sciences, Head of Department of Information Protection in Telecommunication Systems and Networks, Military Institute of Telecommunication and Information technologies named after the Heroes of Kruty.
E-mail: robota_ks@ukr.net.
Orcid ID: 0000-0001-7209-2176.

DOI: 10.18372/2410-7840.25.17936
УДК 004.49

ДОСЛІДЖЕННЯ ПРОБЛЕМАТИКИ БЕЗПЕКИ В ХМАРНИХ СЕРЕДОВИЩАХ ТА ВИРІШЕННЯ З ЗАСТОСУВАННЯМ ПІДХОДУ "БЕЗПЕКА ЯК КОД"

Олександр Вахула, Іван Опірський

"Безпека як код"- це підхід організації безпеки в хмарних середовищах, який полягає на методі інтеграції контролів безпеки, політик та кращих практик безпосередньо в процеси розробки та розгортання програмного забезпечення. Процес інтеграції включає трансформацію вимог безпеки та конфігурацій в програмний код, який в свою чергу вважається невід'ємною частиною повного життєвого циклу розробки програмного забезпечення. Вбудовуванням мір безпеки в код, скріпти, шаблони та автоматизовані робочі процеси, організація забезпечує, що є чітко визначені контролі безпеки, які консистентно та примусово будуть застосовані на всіх операційних фазах створення програмного забезпечення (розробка, тестування, впровадження, підтримка). В даній статті розглянуто основні проблеми побудови безпеки в хмарних середовищах та їх причини, також розглядає складові та принципи підходу «Безпека як код", приклад реалізації з поясненням, переваги даного підходу, а також роль DevSecOps. Ця стаття має на меті допомогти читачам зрозуміти важливість підходу "Безпека як код", як одного з найефективніших методів організації безпеки в хмарних середовищах. Так, як хмарні середовища продовжують розвиватися та поширюватися, а загрози стають все більш складними, підхід "Безпека як код" являє собою основну стратегію для про-активного захисту цифрових активів. Ця публікація слугує посібником для розуміння, впровадження та отримання переваг від підходу "Безпеки як код", надаючи уявлення про майбутній ландшафт безпеки хмарних середовищ та важливу роль автоматизації та інтеграції у вирішенні сучасних викликів безпеки. Для підтримки дослідження було проведена широкий аналіз літератури та статей, які надають інформацію про підхід "Безпека як код" та його застосування.

Ключові слова: *Безпека як код, Інфраструктура як код, DevSecOps, DevOps, хмарні середовища, цикл розробки програмного забезпечення, загрози безпеки.*

ВСТУП

У хмарних обчисленнях, які постійно розвиваються, де гнучкість та інновації поєднуються, неможливо переоцінити важливість надійних заходів безпеки. Оскільки організації продовжують використовувати трансформаційний потенціал хмарних технологій, необхідність захисту цифрових активів від постійно зростаючого спектру загроз стає не просто пріоритетом, а стратегічним імперативом.

«Безпека як код», народжена на стику кібербезпеки та розробки програмного забезпечення, являє собою зміну парадигми того, як організації концептуалізують, впроваджують і підтримують свої стратегії безпеки в хмарних середовищах. Цей підхід інкапсулює злиття принципів і методів безпеки в код, створюючи про-активну, автоматизовану та інтегровану екосистему безпеки, яка бездоганно узгоджується з сучасними методологіями розробки.

В даній статті автори розглядають основні проблеми безпеки, з якими стикаються споживачі хмарних сервісів. Кореневими причинами яких можна вважати нерозуміння моделі спільної відповідальності, яка є фундаментальною; динамічність і масштабованість середовища на відміну від наземної інфраструктури до якої звикли; недостатня видимість ресурсів або "тіньове" IT; недооцінка ризиків пов'язаних з API; складність навігації даних в розподіленому середовищі, в тому числі

чутливих; ручне налаштування конфігурацій, велика ймовірність помилок у зв'язку з людським фактором; складність сервісу управління ідентифікаційними даними та доступами (IaM); мультихмарні та гібридні середовища; брак відповідних спеціалістів з хмарної безпеки, – попит перевищує пропозицію.

Ціль публікації наголосити на вище перечислених проблемах, їх причинах та розглянути підхід "Безпека як код", що допоможе вирішити, якщо не всі, то більшість з них при правильному застосуванні. Авторами досліджено основні принципи, методи, переваги та спосіб реалізації "Безпеки як код" та встановлено, що даний підхід є ефективним та має ряд суттєвих переваг в даній сфері застосування.

Враховуючи зазначене, напрямком подальших досліджень можна вважати розробку методів підвищення ефективності даного методу, а також розширення практики його застосування з ціллю пониження ризиків в хмарних середовищах.

"Управління безпекою як код дозволяє компаніям створювати цінність у хмарі безпечно"[1]. У цьому випадку цінність означає прискорення бізнесу. Існує такий стереотип, що безпека являє собою певну перешкоду для гнучкості і швидкості прийняття та реалізації рішень в бізнесі. Час подолати ці перешкоди і сфокусуватися на цінності та швидкості, яку безпека може принести організації.

Останні дослідження в цій галузі показують, "Безпека як код" - це рушійна сила яка стоїть за майбутнім безпеки аплікацій. Згідно з O'Reilly, так як розробники можуть визначити інфраструктуру за допомогою коду (Infrastructure as Code - IaC), тому це ж саме потрібно зробити, щоб надати безпеці швидкість DevOps [2].

Важливою тенденцією останніх досліджень є також, що "Безпека як код" робить реалізацію безпеки в більш прозорий спосіб та дозволяє професіоналам із безпеки та розробникам говорити однією мовою. Іншими словами, команда безпеки повинна розуміти, як працюють розробники, і використовувати це розуміння, щоб допомогти розробникам створювати необхідні елементи керування безпекою в SDLC (Життєвий цикл розробки програмного забезпечення). Розробники можуть правильно реагувати, використовуючи нові інструменти та методи для покращення безпеки під час процесу розробки [3].

У більшості випадків, щоб забезпечити швидкість у гнучкості бізнесу, безпеку вважають останньою справою, яку потрібно перевірити, коли програмне забезпечення вже розроблено. Крім того, на практиці з кожним новим релізом програмного забезпечення, реалізувати захисні механізми стає важче і важче, як з точки зору часу, так і грошей, якщо тільки це не сплановано та розроблено на ранньому та кожному етапі процесу DevOps. Це призвело до залучення експертів із безпеки до команд DevOps, розробників та IT для співпраці та розробки засобів контролю безпеки в практиках DevOps, що призвело до появи DevSecOps. Простіше кажучи, DevSecOps – це DevOps, які мають у елементи керування безпекою, що забезпечує постійну гарантію захищеності середовища та процесу розробки програмного забезпечення. Здається, Ніл Макдональд з Gartner (MacDonald, 2012) вперше використав термін DevOpsSec (більш популярний як DevSecOps), щоб забезпечити безпеку в практиках DevOps, щоб збалансувати її із швидкістю та гнучкістю. DevSecOps є природним розширенням DevOps, яке реалізовує безпеку за принципом "Зсуву вліво"(Shift-left), що означає на ранніх стадіях і на всіх циклах, шляхом створення автоматизованих засобів контролю безпеки в процесі DevOps [4].

Постановка завдання полягає в аналізі проблематики безпеки в хмарних середовищах та сучасних підходів та методологій, які направлені на їх вирішення. Основною метою є визначення переваг та недоліків цих методів, їх ефективності у боротьбі з кіберзагрозами в хмарних середовищах. Для досягнення поставленої мети потрібно вивчити основні компоненти та принципи підходу, визначити основні засоби, що використовуються для впровадження, а також провести аналіз останніх досліджень та публікацій на цю тему. Отримані результати мають бути використані для розробки пропозицій щодо вдосконалення підходів для безпеки хмарних середовищ.

ОСНОВНА ЧАСТИНА

Проблематика організації безпеки в хмарних середовищах

Хмарні провайдери дотримуються моделі спільної відповідальності щодо безпеки, що призво-

дить до плутанини стосовно того, хто за що відповідає (рис. 1).

Рис. 1. Приклад моделі спільної відповідальності

Недостатня видимість

Хмарне середовище за своєю суттю є комплексним і складається з безлічі сервісів, компонентів, контейнерів і мікросервісів, розподілених по різних регіонах. Ця розподілена багатокомпонентість створює величезну поверхню для атаки, тому вкрай важливо підтримувати повну видимість усіх активів хмарної екосистеми.

Традиційним інструментам безпеки, створеним для наземних середовищ, важко адаптуватися до динамічного хмарного ландшафту. Традиційна концепція «периметра» який не має чітких рамок, що ускладнює моніторинг і захист взаємодії між різними компонентами.

Відсутність видимості призводить до "сліпих зон", де команди відповідальні за безпеку не можуть ефективно контролювати та виявляти події в хмарних ресурсах. Помилки конфігурації, аномальна поведінка, неавторизований доступ і потенційні витоки можуть стати непоміченими, піддаючи ризику конфіденційні дані та критичні для бізнесу сервіси.

Детектування інцидентів, індикаторів компрометації, виявлення першопричини інциденту, так званого "нульового пацієнта", відстеження розповсюдження, стримування стає складним без повного моніторингу цілої хмарної екосистеми.

Дотримання вимог щодо відповідності стандартам є важливою потребою для організацій. Відсутність видимості ускладнює можливість продемонструвати відповідність стандартам аудиторам і регулюючим органам, що потенційно може призвести до штрафів та репутаційних втрат.

Складність сервісу Ідентифікації та Управління Доступом (IAM)

Сучасні хмарні середовища охоплюють широкий спектр сервісів, кожен з яких має власний набір облікових записів користувачів, засобів контролю доступу та авторизації. Ці сервіси можуть охоплювати інфраструктурні ресурси, програми, бази даних тощо, і часто надаються різними хмарними постачальниками.

Кожен хмарний провайдер зазвичай підтримує власне сховище ідентифікаційних даних, у якому зберігається інформація про користувача, облікові дані та політики доступу. Ця різноманітність сховищ ідентифікаційних даних створює так званий силос облікових даних користувача та ускладнює завдання узгодженого керування ідентифікацією в усіх провайдерах та сервісах.

Користувачам і програмам у мультихмарному середовищі часто потрібна взаємодія між різними сервісами. Управління доступом і дозволами, необхідними для цих взаємодій, може швидко стати складним, що призведе до помилок, неправильних налаштувань і прогалин у безпеці.

Величезний обсяг дозволів і ролей, які необхідно визначати, керувати та перевіряти, збільшує ймовірність помилок і недоглядів.

Складні сценарії управління доступом збільшують ризик безпеки. Користувачам можуть бути надані надмірні дозволи або неправильні конфігурації можуть випадково надати неавторизований доступ до конфіденційні даних. Ці прогалини можуть і використовуються зловмисниками для отримання несанкціонованого доступу.

Безпека API

Хмарні сервіси значною мірою базуються на Прикладних програмних інтерфейсах (Application Programming interface – API), які можуть бути вразливими до атак. Дуже часто цей вектор недооцінюється інженерами з безпеки.

Прикладні програмні інтерфейси служать сполучною ланкою, яка забезпечує взаємодію між хмарними сервісами. Технологія дозволяє розробникам отримувати доступ до хмарних ресурсів, маніпулювати даними та виконувати функції віддалено. Ця спрощена взаємодія підвищує ефекти-

вність роботи, але також наражає API на потенційні ризики для безпеки.

Оскільки API забезпечують зв'язок між різними компонентами, вони можуть стати точкою входу для зловмисників. Слабкі місця в дизайні, реалізації або автентифікації API можуть бути використані для отримання неавторизованого доступу, виконання ін'єкцій або несанкціонованого витоку даних.

Поширені вразливості що можуть зашкодити API у хмарних середовищах:

- ін'єкційні атаки (Injection Attacks): брак або недостатня валідація вхідних даних можуть призвести до ін'єкцій, коли зловмисники вставляють шкідливий код або команди у ввід;
- порушена автентифікація (Broken Authentication): слабкі механізми автентифікації або неправильне керування сеансами можуть дозволити неавторизований доступ до API;
- незахищена десеріалізація (Insecure Deserialization): неправильна обробка серіалізованих даних може призвести до віддаленого виконання коду;
- невідповідна авторизація (Inadequate Authorization): недоліки в механізмах контролю доступу можуть дозволити користувачам виконувати дії, на які вони не мають авторизації;
- розкриття конфіденційних даних (Exposure of Sensitive Data): незахищене поводження з даними або неправильне шифрування може призвести до витоку конфіденційної інформації.

У мультихмарних і гібридних хмарних середовищах API сторонніх розробників ще більше ускладнюють ландшафт безпеки. Організації часто покладаються на зовнішні API для спеціалізованих сервісів, збільшуючи поверхню атаки.

Захист даних і відповідність стандартам

Хмарні середовища пропонують гнучкість, дозволяючи організаціям розподіляти дані між різними сервісами, регіонами та навіть декількома хмарними провайдерами. Дані можна зберігати в базах даних, файлових системах, об'єктних сховищах тощо, що охоплює широкий спектр хмарних ресурсів.

Для ефективного захисту даних потрібне шифрування як у стані збереження, так і під час передачі та обробки. Однак різні хмарні сервіси можуть використовувати різні методи шифрування, методи керування ключами та рівні безпеки. Керування шифруванням у цих службах може бути складним.

Управління контролем доступу та дозволами для розосереджених даних є складним завданням. Неправильно налаштований контроль доступу може призвести до неавторизованого доступу, витоку даних і порушення відповідності.

У мультихмарних середовищах, де дані можуть зберігатися на різних хмарних платформах, дотримання нормативних стандартів стає ще складнішим.

Дотримання правил щодо резидентства та юрисдикції даних є складним викликом. Забезпечити збереження та обробку даних у межах правових норм відповідних нормативних актів може бути проблемою, коли дані розосереджені між хмарними сервісам з різними географічними розташуваннями.

Керування конфігураціями безпеки

Невідповідність до вимог та рекомендацій конфігурації є найпоширенішим ризиком, що становить небезпеку для хмарних сервісів і даних.

Можливість гнучкого і швидкого масштабування хмарних ресурсів є причиною відхилення від стандартних конфігурації безпеки згідно організаційних політик, так званого дрейфу. Особливо мануальні зміни конфігурації, де присутній людський фактор і є висока ймовірність помилок.

Дотримання нормативних вимог і галузевих стандартів вимагає консистентних конфігурацій безпеки. Недотримання цих конфігурацій може призвести до порушення комплаєнсу відповідності та юридичних наслідків.

Мультихмарні і гібридні середовища, де використовується кілька хмарних провайдерів, кожен з яких має різні сервіси, інтерфейси та парадигми захисту, відповідно мультиплікується складність менеджменту безпекою.

Кожна хмарна платформа може стати силосом практик безпеки, що ускладнює підтримку узгодженості політик безпеки, засобів контролю доступу та механізмів виявлення загроз.

Для ефективного керування безпекою різні хмарні платформи часто потребують спеціальних знань по кожному з провайдерів. Команди по-

винні розуміти нюанси функцій безпеки та конфігурацій кожної з платформ.

Консистентні процеси щодо виявлення та реагування на загрози в мультихмарному середовищі є викликом для команд безпеки. Різні інструменти та механізми моніторингу ускладнюють уніфікацію процедур виявлення загроз та реагування на інциденти.

У гібридних середовищах, де дані переміщуються між локальною інфраструктурою та кількома хмарними платформами, захист даних і безпечна передача даних стають надскладними, так як не має повної видимості.

Брак спеціалістів з відповідним досвідом безпеки в хмарних технологіях

Попит на експертів з хмарної безпеки перевищує кількість наявних кадрів. Організаціям важко знайти й утримати фахівців із необхідними навичками для розробки, впровадження та керування надійними заходами безпеки в хмарі.

За відсутності досвіду є ризики налаштування вразливих конфігурацій, що стає найпоширенішою причиною витоку даних в хмарних середовищах.

Також, для ефективного виявлення загроз і реагування на інциденти в хмарних середовищах потрібні спеціальні знання та досвід. Різні хмарні постачальники пропонують різні рішення з безпеки, інструменти та практики. Експерти повинні орієнтуватися в усіх цих нюансах, щоб запровадити консистентні заходи безпеки на різних платформах.

Підхід "Безпека як код" в хмарних середовищах

З огляду на всі вище перераховані проблеми, які часом можуть бути стримуючи фактором міграції організацій в хмару, "Безпека як код" є чи не найефективнішим підходом для забезпечення безпеки хмарних середовищ, яка одночасно надає гнучкість та швидкість. Для того щоб успішно реалізувати підхід "Безпека як код", ми потребуємо цілої хмарної стратегії, яка працює, як код також. Основна думка, що ми не можемо реалізувати захист підходом "Безпека як код" для чогось, що не є реалізоване як код.

Більшість споживачів хмарних послуг одностайні з тим, що "Інфраструктура як код" (Infrastructure as a Code-IaC) дозволяє автоматизувати швидке розгортання сервісів у хмарі виключаючи будь, яке ручне налаштування, відповідно і помилки. «Безпека як код» розвиває цей підхід ще далі, визначаючи політики безпеки, стандарти та кращі практики програмно, щоб їх можна було по замовчуванні використовувати в конфігураційних скриптах, які використовуються для налаштування хмарних сервісів і систем. IT департаменти можуть перейти від вічного балансування між гнучкістю бізнесу та безпекою, до усвідомлення, що можна поєднати ці елементи та забезпечити відповідний рівень одного і другого не приносячи в жертву жоден.

Розглянемо спрощено на прикладі (рис. 2), організаційні політики містять список контролів безпеки, які вимагаються, контролі розбиваються на правила, які трансформуються в код, який є зрозумілий для Централізованого сервісу перевірки відповідності до політик, пізніше правила групуються в політики, які організовані ієрархічно та визначаються структурою з наслідуванням. Централізований сервіс перевірки відповідності до політик, являю собою умовно пропускний пункт, де код інфраструктури перевіряється на відповідність ресурсів котрі мають розгорнутися до заданих політик.

Рис. 2. Спрощений опис процесних складових реалізації "Безпека як код"

Якщо, наприклад, організація встановить політику, згідно з якою, персональні дані чи дані платіжних карток в сховищах повинні бути зашифровані під час зберігання, то ця політика буде заявлена в процесі одним з правил, яке автоматично запускається, коли DevSecOps надсилає код на розгортання хмарних ресурсів. Код, який порушує політику, автоматично відхиляється.

Прикладами політики також можуть бути, що образи для розгортання контейнерів чи віртуальних машин мають бути з довірених реєстрів, обов'язкова наявність бекапу бази даних, реплікація ресурсу в дві зони доступності, обов'язкове шифрування диску віртуальної машини, вимоги до тегування та імені ресурсів і тд. Джерелами політик можуть бути стандарти, регуляції, кращі практики та рекомендації, в тому числі зовнішніх інституцій, такі як Cloud Security Alliance (CSA), Center for Internet Security (CIS), NIST (National Insitute of Standard and Technology), GDPR, HIPAA, PCI DSS, SOC2 та відповідно внутрішні. В більшості ці вимоги та рекомендації можуть бути описані, як код, який в свою чергу може являти собою, як превентивний, детективний так і реактивний контроль (рис. 3).

Інфраструктура як код (IaC) є реквізитом що передує модулю статичної перевірки на відповідність дотримання політик. IaC може бути реалізований задоволена за допомогою CloudFormation for AWS, Deployment management for GCP чи Resource Manager для Azure, а якщо потрібне універсальне рішення, тоді Terraform чи Pulumi.

Статична перевірка політики повинна бути інтегрована з CI/CD конвеєром інфраструктурного коду та відповідати кращим практикам GitOps, щоб уникнути встановлення помилкових конфігурацій та виправляти невідповідності на ранніх етапах [5].

Рис. 3. Процес перевірки відповідності політикам

Детективний контроль полягає в перевірці невідповідності змін ресурсів, спричинених неконтрольованими факторами, такими як ручні зміни або встановлення процесу, який не відповідає стандартам IaC. Динамічна перевірка політики забезпечує сканування інфраструктури в реальному часі для підтвердження її поточного стану.

Реактивний контроль виконується відповідно до виявлених подій невідповідності та забезпечує автоматичне виправлення за допомогою безсерверних функцій. Компонента Централізованого сервісу перевірки відповідності до політик може бути реалізована за допомогою Open Policy Agent (OPA) або Regula, які є програмним забезпечення з відкритим кодом. В Cloud Native Computing Foundation (CNCF) OPA була прийнята, як інкубаційний проект у квітні 2019 року. OPA може працювати з файлами у форматі JSON та здійснювати статичну перевірку Infrastructure as Code, що відповідає практиці превентивного контролю.

Якщо розглядати інструмент динамічної перевірки актуального стану відповідності щодо політик вже працюючих хмарних ресурсів, то можна використати Cloud Custodian.

Програмний продукт з відкритим кодом, який є детективним, а також може бути і реактивним контролем. Сам продукт створений на Python, безагентний та може бути розгорнутий як безсерверна функція, де правила мають бути описані у форматі YAML [6].

Роль DevSecOps у реалізації підходу безпеки як коду

DevSecOps, це еволюція філософії DevOps, яка інтегрує безпеку в процес розробки та розгортання програмного забезпечення з ранніх етапів циклу початку. Роль DevSecOps у парадигмі «Безпека як код» є ключовою, оскільки вона гарантує, що питання безпеки впроваджуються протягом усього життєвого циклу розробки програмного забезпечення, даючи про-активний і цілісний підхід до хмарної безпеки.

Пропонуємо розглянути основні принципи методології DevSecOps та як вони перетинаються з підходом "Безпека як код". Для допомоги використаємо графічне відображення життєвого циклу розробки програмного коду із зазначеними контролями безпеки (рис. 4).

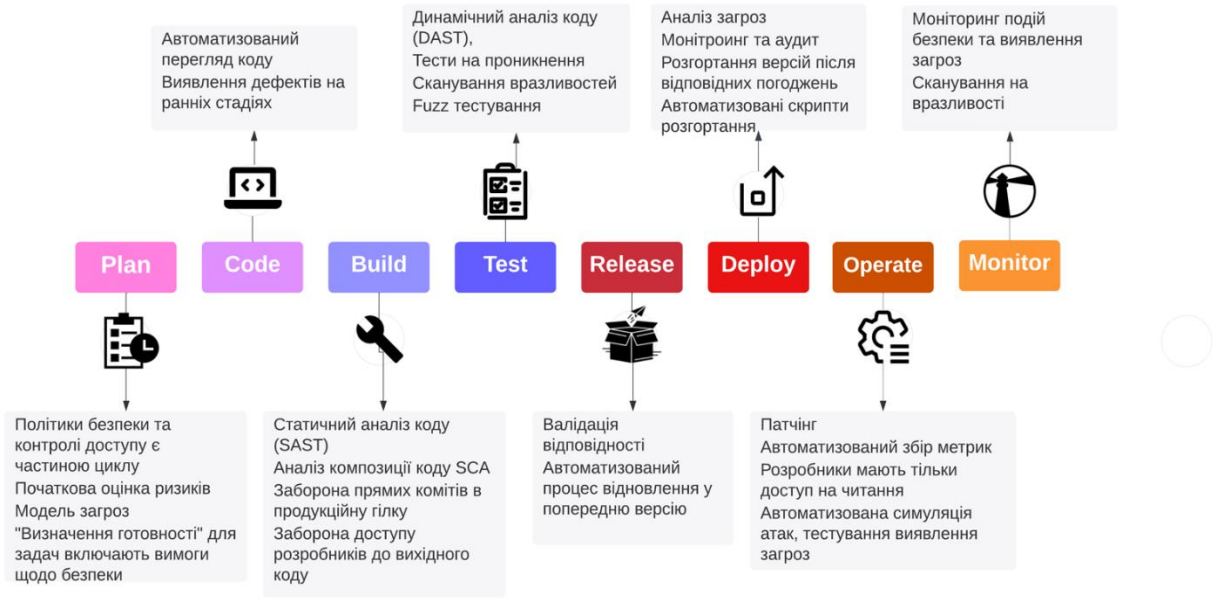

Рис. 4. Цикл розробки програмного забезпечення з контролями безпеки, частина з яких може бути реалізована методом "Безпека як код"

Принцип Shift-Left в DevSecOps практиках означає», що інтеграція безпеки має відбуватися на ранніх етапах розробки. Власне «Безпека як код» забезпечує таке включення контролів, зменшуючи ризик розгортання незахищених конфігурацій [10].

Автоматизовані перевірки відповідності в DevSecOps означає максимальну автоматизацію і виключення мануальної складової в налаштуваннях, яка добре узгоджується з ціллю підходу "Безпеки як код. Автоматизовані перевірки безпеки та сканування можна легко інтегрувати в конвеєри безперервної інтеграції та безперервного розгортання (CI/CD). Це гарантує, що код та інфраструктура оцінюються на відповідність вимогам безпеки на кожному етапі розробки.

Спільний підхід в DevSecOps являє собою співпрацю між командами розробки, операцій та безпеки. У контексті "Безпеки як коду" ця співпраця гарантує, що всі команди розуміють і виконують вимоги безпеки. Експерти з безпеки надають вказівки щодо визначення політик, а розробники впроваджують ці політики у вигляді коду.

Перегляд і аналіз коду: DevSecOps передбачає постійний процес. У парадигмі «Безпека як код» цей процес виходить за межі функціонального коду й охоплює код, пов'язаний із безпекою. Автоматизовані інструменти аналізу коду можуть

допомогти виявити недоліки безпеки та порушення відповідності.

Безперервний моніторинг: DevSecOps означає постійний моніторинг програм та інфраструктури. За допомогою підходу "Безпеки як коду" можна здійснювати моніторинг хмарного середовища на наявність відхилень від політик безпеки та конфігурацій. Автоматизовані інструменти моніторингу можуть швидко ідентифікувати про будь-які відхилення від встановлених стандартів безпеки та здійснювати виправлення до відповідного рівня.

DevSecOps мають мати інструменти швидкого реагування на інциденти безпеки. Реалізація підходу Безпеки як код дозволяє автоматизувати реакцію на інциденти, що стосуються відхилення від усталених практик та політик. Здатність швидко реагувати є критично важливою.

Симбіоз методологій DevSecOps та підходу "Безпека як код" створює синергію та надійний фундамент безпеки для хмарних середовищ. Він узгоджує безпеку з принципами автоматизації, співпраці та постійного вдосконалення, що дозволяє організаціям активно вирішувати проблеми безпеки в динамічному хмарному ландшафті.

Фундаментальні принципи підходу "Безпеки як код"

Виділяють чотири основних фундаментальних принципи підходу "Безпеки як код":

1. Автоматизація – "Безпека як код" має покладатися на автоматизацію для послідовного та масштабованого впровадження політик безпеки. Це включає автоматизацію розгортання засобів контролю безпеки, виявлення вразливостей і усунення проблем;

2. Контроль версій – "Безпеку як код" слід розглядати як програмний код і керувати ним у системі контролю версій. Це забезпечує чітку історію змін, співпрацю між командами та перевірку змін у тестовому середовищі перед продуктовим середовищем;

3. Повторне використання - "Безпека як код" має бути модульним та мати можливість багаторазового використання. Це дозволяє різним командам використовувати та ділитися стандартними елементами керування та конфігураціями безпеки, скорочуючи час і зусилля, необхідні для впровадження безпеки;

4. Відкриті стандарти – "Безпека як код" має будуватися на відкритих стандартах. Це забезпечує більш гнучкий і агностичний підхід, зменшуючи прив'язаність до постачальників і дозволяючи командам використовувати найкращі рішення для різних випадків застосування.

Переваги підходу "Безпеки як код"

Першою перевагою є швидкість. Щоб повною мірою реалізувати бізнес-переваги хмари, команди безпеки повинні рухатися в темпі, до якого вони не звикли в локальних середовищах. Ручні налаштування контролів безпеки створюють тертя, яке сповільнює розвиток і ставить під сумнів загальну цінність хмари для бізнесу.

Друга перевага – зниження ризику. Локальні засоби контролю безпеки просто не враховують нюанси хмари. Хмарна безпека вимагає, щоб її елементи розвивалися протягом усього життєвого циклу розробки. Єдиний спосіб досягти такого рівня інтеграція безпеки через "Безпеку як код".

Даний підхід сприяє розвитку бізнесу. Вимоги щодо безпеки та відповідності стають все більш важливими для продуктів і послуг компаній. У цьому відношенні "Безпека як код" не тільки прискорює час виходу на ринок, але й розширює можливості для інновацій і креативності продукту без шкоди для безпеки.

Покращення співпраці та моралі - оскільки розробницькі команди перейшли швидше до гнучких робочих процесів, то це спровокувало певне відставання команд відповідальних за безпеку, які часто працювали за старими методологіями. При застосуванні підходу команди працюють в одному ритмі та мають спільне розуміння, бо умовно говорять однією мовою коду.

Збільшена видимість та прозорість – із застосуванням підходу "Безпека як код" команди з безпеки чітко розуміють, які політики застосовані і працюють.

ВИСНОВКИ

Формування майбутньої безпеки хмарного середовища за допомогою концепції "Безпека як код" У постійно змінному світі хмарних обчислень, де гнучкість та інновації є важливими, значущість надійних практик безпеки не може бути переоцінена. Як організації приймають цифровий трансформаційний шлях та мігрують свої інф-

раструктури до хмари, важливість динамічного та пристосованого підходу до безпеки стає критичною. Вводиться концепція "Безпека як код", революційна концепція, яка не тільки відповідає вимогам сучасних хмарних середовищ, але також перетворює основи кібербезпеки. Ця публікація показала, що "Безпека як код" - це не просто модний термін, а трансформаційна стратегія, яка поєднує принципи безпеки з практиками розробки програмного забезпечення. Обробляючи політики безпеки, контролі та найкращі практики як код, організації отримують можливість автоматизувати, інтегрувати та впроваджувати заходи безпеки протягом усього життєвого циклу хмарних ресурсів. Одним з висновків з нашого дослідження може бути те , що "Безпека як код" - це більше, ніж технологічна зміна, вона представляє еволюційний зсув. Команди, що складаються з розробників, операторів та експертів з безпеки, об'єднуються навколо спільної мети захисту цифрових активів. Шляхом автоматизованого тестування, постійного моніторингу та ітеративних вдосконалень, ці команди не тільки закривають вразливості, але й просувають культуру прозорої безпеки. Організації різних секторів відчули покращення у сфері безпеки, спрощення забезпечення відповідності вимогам та прискорення часу реагування на інциденти. Концепція виявила свою ефективність у різних хмарних середовищах, від стартапів до підприємств, надаючи стандартизоване середовище, яке відповідає динаміці хмарної інфраструктури. "Безпека як код" є стійкою стратегією, здатною адаптуватися до нових загроз та технологій.

В результаті дослідження можна зробити висновки, що підхід “Безпека як код” при відповідній реалізації може понизити суттєво більшість ризиків які викликані розглянутими проблемами, які становлять найбільшу загрозу для цінних інформаційних активів та ресурсів.

Дана публікація надає нам напрям для подальших досліджень щодо підвищення ефективності даного методу, а також розширення його застосування на більшу кількість сервісів, які представлені хмарними провайдерами, а також дослідження можливості та практичності його застосування в таких типах середовищ, як мультихмарні чи гібридні.

ЛІТЕРАТУРА

[1] Chhavi Adtani, Aaron Bawcom, Jan Shelly Brown, Rich Cracknell, Rich Isenberg, Kaz Kazmier, Pablo Prieto-Munoz, and David Weinstein (2022). Security as code: The best (and maybe only) path to securing cloud applications and systems: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/security - as -code-the-best-and-maybe-only-path-to-securing-cloud-applications-and-systems.

[2] Sarthak Das (2023). Security as Code 1st Edition.

[3] Kim Carter (2017). Francois Raynaud on DevSecOps https://ieeexplore.ieee.org/document/8048652.

[4] Rakesh Kumar, Rinkaj Goyal (2020). Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC): https: // www.sciencedirect.com /science/article/abs/pii/S0167404820302406.

[5] Xuejiao Zhang (2021). Cloud governance and compliance on AWS with policy as code: https://aws.amazon.com/ru/blogs/opensource/cloud-governance-and-compliance-on-aws-with-policy-as-code/.

[6] Xuejiao Zhang (2020). Compliance as code and auto-remediation with Cloud Custodian.

[7] Fausto Lendeborg (2021). Security as Code is the Future to Governing Risk: https://cloudsecurityalliance.org/blog/2021/10/19/security-as-code-is-the-future-to-governing-risk/.

[8] Becki Lee (2022). Using Open Policy Agent (OPA) to Apply Policy-as-Code to Infrastructure-as-Code https:// cloudsecurityalliance.org / blog/ 2020 /04/02/ using - open - policy-agent-opa-to-apply-policy-as-code-to-infrastructure-as-code/.

[9] Ricardo Ferreira (2022). Policy Design in the Age of Digital Adoption: Explore how PolicyOps can drive Policy as Code adoption in an organization's digital transformation 1st Edition.

[10] Saif Gunja (2023). Shift left vs shift right: A DevOps mystery solved: https://www.dynatrace.com/news/blog/what-is-shift-left-and-what-is-shift-right.

RESEARCH ON SECURITY ISSUES IN CLOUD ENVIRONMENTS AND SOLUTIONS USING THE "SECURITY AS CODE" APPROACH

"Security as code" is an approach to security organization in cloud environments, which is based on the method of integrating security controls, policies and best practices directly into the software development and deployment processes. The integration process includes the transformation

of security requirements and configurations into software code, which in turn is considered an integral part of the full software development life cycle. By embedding security measures into code, scripts, templates, and automated workflows, an organization ensures that there are well-defined security controls that will be consistently and enforced across all operational phases of software creation (development, testing, implementation, support). This article examines the main problems of building security in cloud environments and their causes, also considers the components and principles of the "Security as code" approach, an implementation example with an explanation, the advantages of this approach, as well as the role of DevSecOps. This article aims to help readers understand the importance of the Security as Code approach as one of the most effective methods for managing security in cloud environments. As cloud environments continue to evolve and proliferate, and threats become more sophisticated, the Security as Code approach represents a core strategy for proactively protecting digital assets. This publication serves as a guide to understanding, implementing, and benefiting from a Security as Code approach, providing insight into the future cloud security landscape and the critical role of automation and integration in addressing today's security challenges. To support the research, an extensive review of literature and articles providing information on the Security as Code approach and its application was conducted.

Keywords: Security as code, Infrastructure as code, DevSecOps, DevOps, Cloud environments, software development cycle, security threats.

Вахула Олександр Петрович, асистент кафедри захисту інформації Національного університету «Львівська політехніка».

Oleksandr Vakhula, assistant at the Department of Information Security, National University "Lviv Polytechnic".
Email: oleksandr.p.vakhula@lpnu.ua.
Orcid ID: 0009-0008-5367-3344.

Опірський Іван Романович, д.т.н., професор, завідувач кафедри захисту інформації Національного університету «Львівська політехніка».

Ivan Opirskyy, Doctor of Technical Sciences, Professor, Head of the Department of Information Security, National University "Lviv Polytechnic".
E-mail: ivan.r.opirskyi@lpnu.ua.
Orcid ID: 0000-0002-8461-8996.

DOI: 10.18372/2410-7840.25.17937
УДК 004.056.53

ВПРОВАДЖЕННЯ НОВИХ ЗАСОБІВ І МЕТОДІВ ПІДВИЩЕННЯ РІВНЯ КІБЕРБЕЗПЕКИ ОБ'ЄКТІВ КРИТИЧНОЇ ІНФРАСТРУКТУРИ

Андрій Давидюк

Наявні методи та засоби забезпечення кібербезпеки об'єктів критичної інформаційної інфраструктури, розроблені на основі міжнародних стандартів та кращих практик, є досить ефективними в умовах мирного часу, проте не враховують гібридний характер війни, за якого з'являються нові загрози, зокрема такі як фізичне знищення, захоплення противником, відсутність можливості постійного моніторингу та контролю, обмеження в ресурсах захисту та наявному персоналі, проблеми в поставках обладнання для відновлення, перебої в процесах обміну інформацією, потреба у частій зміні умов функціонування, динамічне зростання кількості та якості кібератак тощо, через що їх ефективність значно спадає. З огляду на це виникає потреба у розробці нових та удосконалення існуючих методів та засобів кіберзахисту з метою підвищення рівня кібербезпеки критичної інфраструктури. Від забезпечення кібербезпеки об'єктів критичної інформаційної інфраструктури як невід'ємної частини об'єктів критичної інфраструктури залежить безпека населення, виконання бойових завдань військами.

***Ключові слова**: управління вразливостями, SCAP, опис кібератаки на основі шаблону з траєкторією поведінки керованої системи, оцінка ризиків, візуальна аналітика, антифішингова інфраструктура, система обміну знаннями та досвідом.*

ВСТУП

Забезпечення кібербезпеки є важливим завдання, визначеним Указом Президента України Про рішення Ради національної безпеки і оборони України від 14 травня 2021 року «Про Стратегію кібербезпеки України», Законом України «Про основні засади забезпечення кібербезпеки України», Законом України «Про критичну інфра-