ДОСЛІДЖЕННЯ БЕЗПЕКИ ХМАРНИХ СЕРВІСІВ

Довбня А.А., к.т.н., доц. Горелов Д.Ю.
Харківський національний університет радіоелектроніки,
кафедра КРіСТЗІ, м. Харків, Україна
e-mail: andrii.dovbnia@nure.ua.

Abstract. The subject of the study is data located in the cloud infrastructure that requires the appropriate level of security and privacy, which is a priority need of clients served in public clouds.

У сучасному динамічному світі найбільш важлива конкурентна перевага компаній – їх ефективність, яка визначається швидкістю реакції на зміни умов ведення бізнесу та додавання нового функціоналу. Для багатьох використання хмарних сервісів стає найбільш виправданим вибором. Хмарні сервіси дозволяють:

1) істотно збільшити швидкість виділення ресурсів, необхідних для розвитку сервісу чи додавання нового функціоналу;

2) підвищити ефективність використання ресурсів, спрогнозувати необхідні витрати та врахувати їх у вартості послуг;

3) вирішити проблеми забутих активів та застарілого обладнання, яке може бути задіяне зловмисниками при проведенні атак.

Однак переваги хмарних сервісів супроводжуються новими ризиками, які потрібно мати на увазі при ухваленні рішення про міграцію у хмару.

Ризик №1. Конфіденційність даних або ймовірність несанкціонованого доступу до конфіденційної інформації (баз даних, віртуальних серверів, даних, що передаються у незахищеному вигляді, інші об'єкти орендованої хмари) з боку провайдера послуг. Це пов'язано з тим, що обробка інформації здійснюється на обладнанні провайдера, без можливості фізично контролювати його дії.

Для мінімізації цього ризику рекомендується вживати наступні заходи.

1. Шифрувати конфіденційні дані, що зберігаються у базі даних. Якщо неможливо гарантувати безпеку даних, що зберігаються на фізичному рівні, то потрібно зробити такий доступ не вартим зусиль. Цей захід також дозволить мінімізувати ризик несанкціонованого доступу до опублікованої бази даних. При цьому шифрування можна реалізувати як на рівні програмного забезпечення (ПЗ), так і вбудованими засобами бази даних.

2. Шифрувати дані під час передачі. Імовірність цього ризику мала через високу складність здійснення атаки. Проте, якщо вартість даних висока, то логічним буде використання шифрування на рівні ПЗ, оскільки ціна такого заходу порівняно невелика. Такий захід пред'являє підвищені вимоги до процесів управління ключами та сертифікатами шифрування, оскільки при раптовому закінченні терміну дії сертифіката робота сервісу може бути порушена.

3. Видаляти системних користувачів та/або пакети, створені провайдером, з віртуальних серверів. Дуже часто провайдер додає до віртуальних серверів та інших сервісів додаткові облікові записи або програми для забезпечення можливості зручного адміністрування всіх сервісів прямо з консолі управління хмарною інфраструктурою. Тому використання цього заходу потрібне лише у випадку, коли забезпечення конфіденційності є головним пріоритетом (наприклад, у сегменті обробки даних банківських карток).

4. Заборонити на рівні мережевих сегментів публічний доступ до баз даних. З метою уникнення можливості випадкової публікації бази даних в Internet рекомендується використовувати архітектурні обмеження. Наприклад, коли доступ з Internet можливий лише до одного мережного сегменту, в якому заборонено розміщувати бази даних та інші подібні послуги.

5. Моніторинг подій на рівні хмари. З метою своєчасного виявлення порушень вимог щодо публікації баз даних або ознак компрометації облікових записів необхідно контролювати все, що відбувається у хмарній інфраструктурі. Як правило, провайдер надає спеціалізовані сервіси, що дають змогу реалізувати такий моніторинг (наприклад, Cloudtrail або Audit Trails). Всі події з цих сервісів можна експортувати до SIEM та налаштувати там необхідні правила обмеження доступу.

6. Обов'язково використовувати багатофакторну аутентифікацію для доступу до хмари. Цей захід є обов'язковим при використанні хмари, оскільки при компрометації привілейованого облікового запису можна повністю втратити контроль над хмарою і, отже, усіма бізнес-процесами, реалізованими за допомогою хмарних сервісів.

7. Контролювати цілісність контейнерів та ПЗ. Використання принципів безпечної розробки – найважливіший фактор забезпечення необхідного рівня безпеки сучасних веб-сервісів. Оскільки впровадити шкідливий код можна на всіх етапах життєвого циклу ПЗ, дуже важливо забезпечити безпеку коду, починаючи з етапу проектування та розробки та до введення додатка в експлуатацію.

Ризик № 2. Несанкціонована зміна програмного забезпечення. В умовах використання хмарних сервісів дуже важливо забезпечити цілісність програми, від написання первинного коду до запуску цього коду на сервірі. При використанні зовнішніх сервісів з'являються такі додаткові загрози, як внесення несанкціонованих змін у зібрані контейнери та використання шкідливого коду в процесі компіляції. Для мінімізації даного ризику слід забезпечити повний контроль за репозиторієм первинного коду та регістром контейнерів, а також за цілісністю програмного забезпечення протягом усього його життєвого циклу.

Ризик № 3. Доступність хмарного сервісу або ймовірність порушення доступності сервісу (обмеження доступу через санкційні

обмеження, будь-яка інша відмова у наданні послуги з боку провайдера, технічний збій на устаткуванні провайдера) з вини провайдера.

Для мінімізації цього ризику рекомендується вживати наступні заходи.

1. Обов'язкове резервування даних, первинного коду програм та опису інфраструктури в місці, що не залежить від поточного провайдера хмарних сервісів. Якщо виникне ситуація, коли доступ до хмари стане неможливим, це означатиме втрату всього, що знаходиться в хмарі. При цьому для відновлення роботи сервісу на інших ресурсах обов'язково необхідно мати резервну копію даних та первинного коду вживаних сервісів. Якщо ці дані відсутні, то роботу сервісу практично неможливо відновити. Резервні копії необхідно зберігати в незалежному від хмари місці, наприклад, в іншому хмарному сервісі або власному фізичному сервері. Для прискорення процесу відновлення роботи сервісу інших ресурсах рекомендується використовувати підхід "Інфраструктура як код" з обов'язковим резервуванням опису використовуваної інфраструктури.

2. Заборона використання рутового облікового запису для адміністрування. Рутовий обліковий запис – це обліковий запис, який використовувався для реєстрації в хмарному сервісі. Він має найширші права, і якщо його втратити, то буде вкрай складно відновити контроль над хмарою. Тому рекомендується максимально знизити ризик компрометації такого облікового запису, що виникає при використанні його для адміністрування або виконання інших задач.

3. Моніторинг повідомлень від провайдера про технічне обслуговування або деградацію обладнання. Провайдер досить часто здійснює технічне обслуговування обладнання, іноді це пов'язано, наприклад, з деградацією жорстких дисків. Буває, що для проведення робіт провайдеру потрібно вимкнути сервіс. Як правило, він попереджає про це, щоб можна було оперативно перейти на резервне обладнання. Але якщо ігнорувати повідомлення провайдера, то можлива ситуація, коли сервіс перестане працювати через те, що ключовий сервер раптом вимкнувся.

4. Окремі організаційні підрозділи (Organizational Units) для адміністраторів, фахівців з безпеки, тестувальників, розробників та основної інфраструктури. Розділяти функціонал важливо як мінімум з двох причин. Насамперед, інфраструктура, що використовується для розробки або тестування, дуже часто змінюється, а нижчі вимоги безпеки підвищують можливість компрометації сервісів або облікових записів. Тому відділення цих сегментів від виробничого дозволить мінімізувати вплив даних сегментів на роботу сервісу. Щодо відділення функцій безпеки в окремий сегмент, то це насамперед необхідно для забезпечення безпеки журналів аудиту.

Ризик № 4. Висока вартість. Простота та швидкість виділення ресурсів у хмарі – незаперечна перевага такого підходу. Це дозволяє

швидко масштабувати послуги зі збільшенням кількості запитів від клієнтів. З іншого боку, важливо пам'ятати, що не обмеження виділення ресурсів призведе до значного росту вартості використовуваних ресурсів. Особливо гостро проблема необмеженого масштабування проявляється під час DDoS-атак. Якщо сервіс намагатиметься обробити всі запити, то це призведе до того, що вартість ресурсів перевищить можливі межі, а бази даних будуть забиті марною інформацією.

Важливо звернути увагу на ефективність використання ресурсів. Якщо, наприклад, були орендовані потужні сервери, які завантажені менш ніж на 10%, це також призведе до зниження прибутку.

Для мінімізації можливих наслідків слід:

– лімітувати сервіси, що масштабуються (як правило, всі хмарні послуги виділяють ресурси в рамках квот, тому достатньо вказати такі значення квот на ресурси, вартість яких буде прийнятною);

– використовувати засоби захисту від атак на прикладному рівні (ця вимога є обов'язковою як мінімум з двох причин: по-перше, запити від зловмисників не приносять жодного доходу, отже, немає сенсу витрачати ресурси на їх обробку, по-друге, відсутність захисту на прикладному рівні істотно підвищує ризик успішної атаки на сервіс. При цьому можливі різні варіанти реалізації захисту на прикладному рівні, починаючи від опенсорсного або промислового міжмережевого екрану програмного рівня та закінчуючи сервісами очищення трафіку);

– контролювати завантаження віртуальних серверів (при оренді віртуального сервера сплачується його вартість. Отже, чим менш завантажений сервер, тим дорожче в результаті обходиться його експлуатація. Дані щодо використання ресурсів можна отримати із логів операційної системи сервера).

Список використаних джерел.

1. Tari, Z. Security and Privacy in Cloud Computing. IEEE Cloud Comput. 2014, 1, 54–57.

2. Mollah, M.B.; Azad, M.A.K.; Vasilakos, A. Security and privacy challenges in mobile cloud computing: Survey and way ahead. J. Netw. Comput. Appl. 2017, 84, 38–54.

3. Abdulsalam, Y.S.; Hedabou, M. Decentralized Data Integrity Scheme for Preserving Privacy in Cloud Computing. In Proceedings of the 2021 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), Chengdu, China, 18–20 June 2021; pp. 607–612.

4. Sgandurra, D.; Lupu, E. Evolution of attacks, threat models, and solutions for virtualized systems. ACM Comput. Surv. 2016, 48, 1–38.

5. Subramanian, N.; Jeyaraj, A. Recent security challenges in cloud computing. Comput. Electr. Eng. 2018, 71, 28–42.

6. Cook, A.; Robinson, M.; Ferrag, M.A.; Maglaras, L.A.; He, Y.; Jones, K.; Janicke, H. Internet of cloud: Security and privacy issues. In Cloud Computing for Optimization: Foundations, Applications, and Challenges; Springer: New York, NY, USA, 2018; pp. 271–301.

7. Zhang, Y.; Xu, C.; Lin, X.; Shen, X.S. Blockchain-based public integrity verification for cloud storage against procrastinating auditors. IEEE Trans. Cloud Comput. 2019, 9, 923–937.