УДК 621.396

РЕКОМЕНДАЦІЇ ДЛЯ ПІДВИЩЕННЯ РІВНЯ БЕЗПЕКИ ХМАРНИХ СИСТЕМ НА ПРИКЛАДІ OPENSTACK

Кибенко А.В., Лящук А.А.
Інститут телекомунікаційних систем КПІ ім. Ігоря Сікорського, Україна
E-mail: endryu-kibenko@outlook.com, aliashchuk@outlook.com

Recommendations for improving the security of cloud systems by example of OpenStack

Different types of attacks on the elements of the cloud are considered. Solutions concerning the protection from cloud computing security threats are presented.

Не секрет, що хмарні технології, в даний час, перебувають на хвилі популярності: економічність, легкість розгортання, розрахована на багато користувачів архітектура – все це сприяє швидкому поширенню хмар і захоплення ними більшої частини ринку IT. Економічність хмар робить їх особливо популярними для зберігання інформації. Однак хмарна інфраструктура також представляє підвищені ризики і більш обмежену можливість контролю. В цьому і полягають головні проблеми хмарних обчислень – захист інформації та довіру користувачів по відношенню до хмарних провайдерів.

Ключовими елементами хмарної системи є гіпервізор, який керує віртуальним середовищем хмари, центр обробки даних, на якому міститься велика частина конфіденційної інформації, канал зв'язку між споживачем хмарного сервісу, а також ПЗ, встановлене на комп'ютері споживача (зокрема, інтернет-браузер) [1].

Як і будь-яка система, яка функціонує за допомогою Інтернету, хмарна система часто піддається атакам. Основними видами таких атак є: традиційні атаки на ПЗ, атаки на клієнта, мережеві атаки, атаки на сервери хмари і комплексні атаки.

Далі докладніше розглянемо рекомендації щодо підвищення рівня безпеки на серверах хмари, так як система OpenStack встановлюється безпосередньо на них. Для убезпечення даної системи, необхідно забезпечувати безпеку кожного компонента OpenStack, вибираючи для цього найбільш ефективні рішення відповідно до ролі кожного з компонентів і його важливістю в рамках комплексного середовища OpenStack.

Безпека OpenStack CLI. Використання інструментів OpenStack CLI вимагає застосування імені користувача та пароля. Рекомендовано використовувати змінні OS_USERNAME і OS_PASSWORD, встановлені у файлі OpenStack RC. Проте, зберігання в звичайному, незашифрованому файлі реквізитів для ідентифікації суперечить зарекомендованим практикам забезпечення безпеки.

Для додаткового забезпечення захисту, інструменти OpenStack CLI можуть запитувати ім’я користувача та пароль для кожного запиту, або ж

використовувати наданий токен ідентифікації. Також для цих цілей можуть бути використані вузол або віртуальна машина, розташовані в окремій внутрішній демілітаризованій зоні, або інструменти OpenStack CLI, розташовані в цьому додатковому вузлі виключно для подібних цілей. Таким чином важливо забезпечити три основні дії:

- впевнитись в тому, що всі непотрібні сервіси вимкнені;
- дозволити доступ тільки по SSH і через перевірену мережу;
- відключити bash history і зберігати всі логи в ізольованому, віддаленому, захищеному і високодоступному репозиторію зберігання.

Безпека Nova. Сервіс Nova є найскладнішим сервісом OpenStack, оскільки він пов'язаний майже зі всіма сервісами платформи і має велику кількість власних опцій конфігурації. Тому для даного сервісу потрібно приділити найбільшу увагу, пов’язану з посиленням заходів безпеки. Для досягнення найбільшої безпеки для сервісу Nova, потрібно зробити наступне:

- адміністратору системи, який займається конфігурацією файлів, потрібно надати права суперкористувача (root-права). Також потрібно дозволити читати та перезаписувати файли для власника та читати файли для групи;
- відключити PCI-транзитний шлюз на гіпервізорі для обмеження доступу з віртуальної машини до апаратної частини інфраструктури;
- для зниження рівня загроз на адресу тенантів, незалежних користувачів системи, потрібно відключити всі механізми оптимізації пам’яті гіпервізора;
- всі логи необхідно зберігати в захищеному і віддаленому сховищі;
- використовувати TLS для VNC-сесій;
- необхідно переконатись, що Nova безпечно з’єднується з іншими сервісами OpenStack, використовуючи TLS.

Безпека Glance. Даний сервіс дозволяє зберігати образи, які необхідні для запуску нових віртуальних машин. Тому для уникнення замахів на цілісність цих образів важливо підтримувати безпеку даного сервісу. Це забезпечується наступним шляхом:

- використання підписаних образів Glance;
- невикористання вбудованих образів чи контейнерів з неперевірених джерел, оскільки останні можуть містити вразливості або шкідливі програми.

Безпека Neutron. Сервіс Neutron необхідний для забезпечення зв’язаності мережі і IP-адрес для віртуальної машини в хмарі. Даний сервіс заснований на плагінах, тому важливо розуміти, які саме плагіни потрібні для виконання цих завдань, а які використовуються для сторонніх цілей. Тому потрібно забезпечити наступні умови:

- використовуйте лише ізольовану мережу управління для сервісів OpenStack;
- використовуйте ізоляцію L2 з сегментацією VLAN або GRE-тунелі;
- всі API-запити груп безпеки Nova повинні перенаправлятися в Neutron;
- використовуйте мережеві квоти для пом'якшення впливу DoS-атак.

Безпека черги повідомлень Message Queue (RabbitMQ). Черга повідомлень спрощує комунікації для сервісів OpenStack, а RabbitMQ є найпопулярнішим рішенням в цьому класі для хмар на основі OpenStack. Платформа OpenStack не

підтримує підписування повідомлень – таким чином, черга повідомлень повинна забезпечувати захищену передачу в рамках обміну між сервісами OpenStack. Забезпечте наступні умови:

- захистіть RabbitMQ API за допомогою TLS;
- зберігайте логи RabbitMQ в захищеному і віддаленому сховище;
- видаліть гостьового користувача RabbitMQ;
- використовуйте окремий віртуальний хост RabbitMQ для кожного OpenStack-сервісу;
- використовуйте унікальні ідентифікатори повноважень і відповідні права доступу для кожного віртуального хосту RabbitMQ.

Безпека Keystone. Даний сервіс ідентифікує всі сервіси OpenStack, тому повинен бути захищений від, так званого, спуфінгу та від інших атак. Сервіс не надає методів реалізації політики надійності паролів та невдалих спроб аутентифікації, проте він може використовувати зовнішню систему аутентифікації. Для досягнення найбільшої безпеки для сервісу Keystone, потрібно зробити наступне:

- мультифакторна аутентифікація повинна бути задіяна через систему зовнішньої аутентифікації, типу такої, яка присутня в Apache HTTP Server;
- використовуйте токени Fernet, які розроблені спеціально для REST API, так як вони є більш захищеними в порівнянні зі звичайними токенами, а також вимагають менше ресурсів;
- використовуйте домени Keystone для більш точного розмежування прав доступу для тенантів. Власник домену може створювати додаткових користувачів, групи, а також ролі всередині домену.

Безпека Cinder. Cinder забезпечує високий рівень API для управління блоковими пристроями зберігання даних і активно використовується сервісом Nova. Його слід захищати від DoS-атак, витоків інформації, несанкціонованого доступу та інших загроз. Для цього потрібно зробити наступне:

- встановіть розмір максимального запиту;
- для безпечного видалення томів Cinder використовуйте повну очистку томів.

Підсумовуючи можемо констатувати, що на додаток до згаданих вище рекомендацій, вкрай важливо постійно бути в курсі вразливостей OpenStack і прагнути підтримувати робоче середовище оновленим. Зміцнення безпеки середовища OpenStack має відбуватися на декількох рівнях, починаючи з фізичного і продовжуватись на рівні додатків і рівні організації процесів [2].

Література

1. БЕЗОПАСНОСТЬ ОБЛАКА В ДЕТАЛЯХ / П. В. Ивонин. // Безопасность информационных технологий. – 2013. – №2. – С. 37–40.
2. БЕЗОПАСНОСТЬ В OPENSTACK: ШАГ ЗА ШАГОМ [Електронний ресурс] // Сервионика. – 2016. – Режим доступу до ресурсу: http://servionica.ru/pressCenter/view/814/.