УДК 004.272, 004.032.24

Р. І. Банах, А. З. Піскозуб, Я. Я. Стефінко
Національний університет "Львівська політехніка",
кафедра захисту інформації,
кафедра безпеки інформаційних технологій

СТВОРЕННЯ КОНЦЕПЦІЇ ЗАХИЩЕНОЇ ХМАРНОЇ ОБЧИСЛЮВАЛЬНОЇ ІНФРАСТРУКТУРИ З ВИКОРИСТАННЯМ СИСТЕМ ПРИМАНОК

Наведено концепцію захищеної хмарної обчислювальної інфраструктури, а саме опис компонентів, вимоги щодо компонентів згідно із кращими практиками, сценарії взаємодії клієнтів та зловмисників з нею. Запропоновано вирішення проблеми за допомогою як публічних, так і приватних хмарних обчислювальних рішень, порівняно сервіси, які повинні бути використані в публічних та приватних хмарних рішеннях для реалізації цієї схеми.

Ключові слова: хмарні обчислення, системи приманки, системи виявлення вторгнень, інфраструктура, віртуалізація.

Concepti of secured cloud computing infrastructure is presented, namely – description of components, requirements for components according to the best practices, users and attackers interaction scenarios. The current issue is proposed to be addressed either using private or public clouds, services comparison that have to be used in public and private clouds was provided in order to introduce the current scheme.

Key words: cloud computing, honeypots, intrusion detection systems, infrastructure, virtualization.

Постановка проблеми

Сьогодні хмарні обчислення відіграють значну роль у розвитку нових проектів, оскільки відомі можливістю швидкого розгортання середовища розробки будь-якої потужності. Суть концепції хмарних обчислень полягає в наданні кінцевим користувачам віддаленого динамічного доступу до послуг, таких як програмне забезпечення, платформа чи інфраструктура. Та якщо провайдери хмарних сервісів забезпечують захист ресурсів, які надають у користування, то про захист ресурсів, які користувач надає у загальний доступ, він повинен подбати сам.

Аналіз останніх публікацій

Зважаючи на сучасні кращі практики із створення захищеної хмарної обчислювальної мережі, вони повинні володіти такими характеристиками [2]:

– частина, відповідальна за управління інфраструктурою, має бути ізольованою від навколишнього середовища;
– програмним забезпеченням, що постійно оновлюється;
– зі сторони клієнта доступ до усіх ланок інфраструктури повинен здійснюватись лише з однієї точки;
– наявною системою моніторингу;
– наявною централізованою системою реєстрації подій;
– наявністю системи аудиту інформаційної безпеки;
– наявністю системи виявлення вторгнень та запобігання їм.

Саме таку модель пропонують автори (рис. 1). У цій схемі ми також пропонуємо використовувати систему-приманку, що є нетиповим рішенням для хмарної обчислювальної

мережі, оскільки сьогодні провайдери хмаркових сервісів не забезпечують інтегрованих засобів для виявлення вторгнень чи систем приманок, на відміну, наприклад, від моніторингових чи поштових сервісів [1].

Постановка завдання

З метою створення системи захисту корпоративних ресурсів, розроблення програмного забезпечення для створення зв'язку між компонентами захисту, забезпечення захищеності клієнтів і критичних ресурсів, постає задача розроблення моделі захищеної хмарної обчислювальної інфраструктури з використання систем приманок.

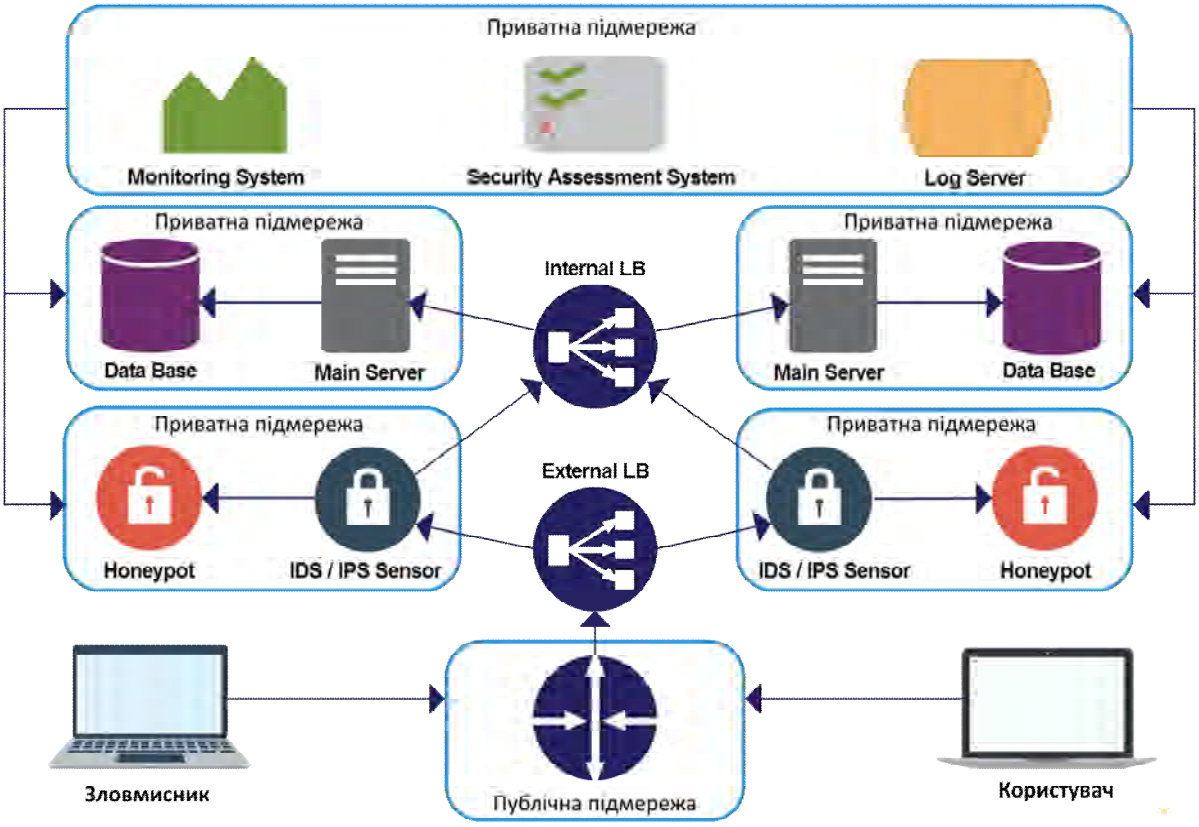

Рис. 1. Схема взаємодії клієнтів із захищеною інфраструктурою

Побудова інфраструктури

На сучасному етапі розвитку суспільства суттєво збільшився попит на використання сервісів публічних хмарних провайдерів. Сьогодні більшість із них володіють великою кількістю інтегрованих засобів моніторингу, розподілу навантаження, базами даних. Для побудови інфраструктури достатньо лише вибрати потрібні компоненти і підтвердити їх створення.

Та зовсім не обов'язково купувати ресурс у компаній-провайдерів публічних хмарних обчислень, його можна реалізувати самотужки за допомогою серверної робочої станції та програмного комплексу OpenStack [3], або ж, наприклад, гіпервізора KVM [4]. Звичайно ж, гіпервізори не зможуть забезпечити таких послуг, як програмний комплекс OpenStack, але вони споживатимуть значно менше ресурсів.

У випадку приватної хмарної інфраструктури її власнику доведеться заміняти інтегровані засоби програмними, імплементуючи їх на базі віртуальних машин (таблиця).

Та не всі інтегровані сервіси публічних хмарних обчислювальних мереж можуть повністю задовольнити потреби клієнтів. Наприклад, сервіс моніторингу CloudWatch в Amazon[5] буде безкорисним у разі, якщо потрібно вести моніторинг не лише апаратного забезпечення, але й

програмного забезпечення на віртуальних машинах. У такому випадку все одно доведеться розгортати додатковий сервіс моніторингу на базі віртуальної машини.

Порівняння сервісів у приватних та публічних хмарних обчислювальних мережах

Сервіс	Інтегровані засоби Amazon Web Services	Аналогічні програмні засоби
Моніторинг інфраструктури	Cloud Watch	Nagios, Zabbix
Автоматичне створення віртуальних машин за потреби	Auto Scaling / Cloud Formation	Vagrant, Ansible, Chef, Puppet
Гіпервізор	Virtual Servers in the Cloud (EC2)	KVM, OpenStack
Бази даних	RDS	MySQL, MongoDB, OracleDB
Сервіс оповіщення поштою	Email Sending Service	Postfix, Dovecot
Балансування навантаження	Elastic Load Balancer	Haproxy

У цій схемі (див. рис. 1) наведена систему реєстрації подій (Log Server), яка дає змогу централізовано збирати і частково аналізувати дані з віртуальних машин чи інших сервісів інфраструктури. Аналізують дані за допомогою регулярних виразів. Система реєстрації подій реалізовується на окремій віртуальній машині, на якій встановлюється такий програмний засіб, як, наприклад, Logstash [6].

Також у схемі присутня система оцінювання захищеності. Вона повинна бути реалізована як окрема ланка, тобто окрема віртуальна машина в мережі зі встановленим програмним засобом, таким як, наприклад, OpenVAS [7], якщо це стосується оцінювання операційної системи і встановленого на неї програмного забезпечення, або, наприклад, Suricata [8], якщо потрібна додаткова перевірка веб-аплікації.

Систему виявлення вторгнень можна реалізувати за допомогою пакета Snort. Snort володіє великою кількістю стандартних сигнатур для виявлення вторгнень. Окрім того, можна створювати власні сигнатури, базуючись на розбиранні вхідних чи вихідних пакетів.

Системою приманкою може бути будь-що, що надається у відкритий доступ. Це може бути база даних, веб-сторінка чи будь-який інший вразливий сервіс, який може бути легкою здобиччю для зловмисника. Основною ціллю застосування системи приманки є реєстрація дій зловмисника та збирання інформації про нього.

Взаємодія користувачів з інфраструктурою

Клієнт, який має намір взаємодіяти з цією інфраструктурою, для початку повинен підключитись до публічної мережі за допомогою публічної IP-адреси або ж за публічним доменним іменем.

Вхідний трафік, що проходить через зовнішній розподілювач навантаження (External LB), використовується як HTTPS-термінатор, тобто це проксі-трафік, який перетворюється на HTTP, перед тим, як пройти через систему виявлення вторгнень та запобігання їм (IDS/IPS Sensor). Якщо ж опустити цю операцію, то IDS не зможе ідентифікувати атаку.

У схемі без систем приманок (Honeypot) IDS аналізує пакети, а після цього вирішує, прийняти цей пакет чи відхилити. За цією схемою, якщо сигнатура спрацює на пакет від певного користувача, для його сесії буде прокладено маршрут на систему-приманку (рис. 2).

Рис. 2. Взаємодія зловмисника з інфраструктурою

Якщо ж відправлена користувачем інформація не містить жодних даних, на які реагують сигнатури IDS, то для такого користувача буде встановлено зв'язок із внутрішнім розподілювачем навантаження (Internal LB). Після цього Internal LB оцінить навантаження прилеглих до нього ресурсів, тобто з одним із блоків, у яких розташовано головний сервер (Main Server) та базу даних (DataBase) (рис. 3). З'єднання клієнта буде встановлено з тим із блоків, який на той момент буде менш навантажений.

Рис. 3. Взаємодія користувача з інфраструктурою

Після того, як запит буде здійснено, вихідні дані передаватимуться у зворотному напрямку. Вони повинні бути зашифровані з використанням протоколу SSL на виході з мережі. Цим займається зовнішній розподілювач навантаження. Він власне і розшифровує дані на вході в мережу, в інакшому випадку IDS/IPS не зміг би проаналізувати вхідних даних, а отже, і виявити атаку.

Ця архітектура містить декілька підмереж, основною метою яких є забезпечення контролю над маршрутизацією і перескерування трафіку на систему-приманку (Honeypot), у разі ідентифікації вторгнень системою виявлення вторгнень та запобігання їм.

За допомогою такої архітектури ми можемо спростити спостереження та ізолювати певну групу ресурсів, а аномальну активність фіксувати за допомогою сервера реєстрації подій. У результаті у такій мережі ми можемо вирішувати, якого клієнта допускати до обробки інформації, з яким припинити з'єднання, а до якого застосовувати обмеження швидкості. Це можна зробити як за допомогою веб-сервера (наприклад, Nginx[9]), так і за допомогою іншого, віртуального мережевого обладнання.

Після збереження статичних файлів, баз даних або іншої інформації дані повинні бути захищені і зашифровані. Ми можемо виявити аномалії на наших машинах за реєстром змін файлів. Наприклад, деякі файли є статичними, і ми знаємо, що вони не повинні змінюватись. Якщо передбачають виявлення незаконних змін, то додатково до системи моніторингу потрібно додати інструмент, який сигналізуватиме про вторгнення і працюватиме в режимі реального часу.

Перевірку цілісності файлів, зокрема оповіщення в режимі реального часу можна отримати за допомогою локальної системи виявлення вторгнень OSSEC [10].

Динамічний антивірусний захист є обов'язковим для машин на базі операційних систем сімейства Windows. Такий захист повинен передбачати агрегацію журналів подій.

Основним засобом, який забезпечує живучість цієї інфраструктури, є реплікація основних елементів, за доступ до яких відповідають розподільники навантаження: зовнішній і внутрішній.

Будь-які події, які відбуваються у цій мережі, реєструє централізований сервер реєстрації подій (Log Server, див. рис. 1). Головний пріоритет у реєстрації подій:

- пов'язаних з IDS;
- подій антивірусів;
- пов'язаних із системою-приманкою.

Висновок

Проблема безпеки у публічних хмарних обчислювальних мережах є однією із найгостріших у сучасному світі інформаційних технологій, оскільки цей ресурс став доступний кожному за своїм кошторисом. Та провайдери хмарних сервісів забезпечують захист інформації лише на рівні надання своїх послуг.

Системи приманки у приватних хмарних обчислювальних інфраструктурах можуть бути легко інтегрованими в роботу інших корпоративних ресурсів. Вони можуть бути, наприклад, доступними із безпровідних мереж, або ж і бути їх складовою. Тому, подальшим напрямком дослідження буде інтегрування систем приманок у безпровідні мережі за допомогою приватних хмарних обчислень.

1. Monjur Ahmed. Cloud computing and security issues in the cloud / A. Monjur, A. H. Mohammad. // International Journal of Network Security & Its Applications (IJNSA). – 2014. – С. 25–36. 2. LahavSavir Best Practice for Ultra Secure Deployment on Amazon Cloud: [Електрон. ресурс]. – Режим доступу: http://www.emind.co/how-to/best-practice-for-ultra-secure-deployment-on-amazon-cloud. 3. OpenStack [Електронний ресурс]: портал OpenStack. – Режим доступу: https://www.openstack.org/. – Назва з титул. екрана. 4. KernelVirtualMachine [Електронний ресурс]: порталKVM– Режим доступу: http://www.linux-kvm.org. – Назва з титул. екрана. 5. AmazonWebService[Електронний ресурс]: порталAmazon. – Режим доступу: http://aws.amazon.com/. – Назва з титул. екрана. 6. Logstash | Collect, Enrich&TransportData [Електронний ресурс]: порталElastic. – Режим доступу: https://www.elastic.co/products/logstash. – Назва з титул. екрана. 7. OpenVAS, OpenVulnerability AssessmentSystem [Електронний ресурс]: порталOpenVAS. – Режим доступу: http://www.openvas.org/. – Назва з титул. екрана. 8. SuricataOpenSourceIDS / IPS / NMSengine [Електронний ресурс]: портал Suricata. – Режим доступу: http://suricata-ids.org/. – Назва з титул. екрана. 9. Nginx[Електронний ресурс]: порталNginx. – Режим доступу: http://nginx.org/. – Назва з титул. екрана. 10. OpenSourceSECurity, OSSEC[Електронний ресурс]: порталOSSEC. – Режим доступу:http://www.ossec.net/. – Назва з титул. екрана.