УДК 004.424

Кучмійова Т. С.,
к.е.н., доцент;
Макеєв В. В.,
здобувач вищої освіти;
Гулак О. Г.,
здобувач вищої освіти,
Миколаївський національний аграрний університет

ОСНОВНІ ФОРМИ ПОВНОЇ ВІРТУАЛІЗАЦІЇ ТА ПРИНЦИПИ ЇХ РОБОТИ

У сучасних умовах стрімкого розвитку інформаційних технологій зростає потреба в ефективному використанні обчислювальних ресурсів. Одним із ключових підходів до вирішення цього завдання є віртуалізація – процес симуляції програмного та апаратного забезпечення, на якому функціонує інше програмне забезпечення. Цей підхід дозволяє створювати ізольовані середовища – віртуальні машини (ВМ), які можуть паралельно виконувати різні задачі без взаємного впливу. Серед багатьох форм віртуалізації, саме повна віртуалізація є предметом особливого дослідження. У даній моделі є одна або декілька операційних систем (ОС) та їх додатки виконуються поверх віртуального обладнання, функціонуючи як гостьові ОС у межах окремих ВМ [1].

Центральним компонентом архітектури повної віртуалізації є гіпервізор (або монітор віртуальних машин, VMM). Його фундаментальна роль полягає у контролі потоку інструкцій між гостьовими ОС та фізичним апаратним забезпеченням, ефективному розподілі системних ресурсів та забезпеченні ізоляції гостьових ОС. Гіпервізор надає гостьовим ОС інтерфейси віртуального обладнання, які значною мірою імітують фізичну платформу, дозволяючи запускати стандартні, немодифіковані ОС та додатки. Слід також згадати паравіртуалізацію – варіацію повної віртуалізації, за якої гіпервізор пропонує гостьовій ОС спеціалізовані інтерфейси для оптимізованого доступу до ресурсів (наприклад, дискової підсистеми чи мережі), що, вимагає модифікації самої гостьової ОС для використання цих інтерфейсів.

Для безпеки віртуалізація пропонує систему снапшотів, в яких фіксується збережений стан віртуальних машин в конкретний момент часу, яка дозволяє миттєво повернути віртуальну машину до попереднього стану. Це надзвичайно корисно перед встановленням критичних оновлень, програмного забезпечення, зміною конфігурації або в разі збою/зараження шкідливим ПЗ. Якщо щось пішло не так, ви просто відкочуєтеся до снапшоту, створеного перед цією дією. Розробники та тестувальники можуть створювати снапшоти перед тестуванням нового коду або конфігурацій, або використовувати їх як форму швидкого збереження стану, хоча це і не є повноцінною заміною професійних систем резервного копіювання[2].

Повна віртуалізація реалізується у двох основних архітектурних формах, вибір між якими є значним як з операційної, так і з безпекової точок зору. Перша форма – bare metal (нативна) віртуалізація – характеризується тим, що гіпервізор встановлюється безпосередньо на фізичне апаратне забезпечення сервера, фактично замінюючи традиційну операційну систему хоста. Даний нативний гіпервізор може бути інтегрований у прошивку обладнання. Такий підхід є основним для віртуалізації серверів у масштабах дата-центрів та корпоративних інфраструктур, де критично важливі ефективність та надійна ізоляція.

Типовими представниками нативних гіпервізорів, що використовуються в галузях, є VMware ESXi, Microsoft Hyper-V (у серверних редакціях), а також KVM та Xen. З безпекової точки зору, нативний гіпервізор, маючи зазвичай значно менший обсяг коду порівняно з повноцінною хостовою ОС, надає меншу поверхню атаки. Однак, його безпека залежить передусім від нього самого, роблячи гіпервізор єдиною, критично важливою точкою захисту.

Натомість, друга форма – hosted віртуалізація – реалізується шляхом запуску гіпервізора як звичайного додатку поверх існуючої, повноцінної хостової операційної системи. Це може бути практично будь-яка поширена ОС, як-от Windows, Linux чи MacOS. Цей підхід більш характерний для віртуалізації на робочих станціях та персональних комп'ютерах (наприклад, за допомогою Oracle VirtualBox або VMware Workstation/Fusion), оскільки він дозволяє користувачеві вільно використовувати інші додатки хостової ОС (як веб-браузери чи поштові клієнти) паралельно з віртуальними машинами. Зазвичай hosted рішення також включають встановлення додаткового програмного забезпечення (гостьових доповнень) всередині гостьової ОС, яке надає допоміжні утиліти, наприклад, для спрощеного обміну файлами чи спільного використання буфера обміну з хостом.

З точки зору безпеки, hosted віртуалізація додає значний крок складності та потенційних вразливостей через наявність та взаємодію з хостовою ОС. Безпека гостьових ВМ у цьому випадку безпосередньо залежить від рівня захищеності базової хостової ОС, що, як правило, підвищує загальний ризик порівняно з нативною архітектурою. Саме тому організації мають чітко визначати політики використання кожного типу віртуалізації, зважаючи на необхідний рівень безпеки та функціональності.

Емуляція обладнання (hardware emulation), яка є типом hosted віртуалізації, відрізняється тим, що гіпервізор емулює апаратне забезпечення, яке може значно відрізнятися від фізичного, дозволяючи запускати ОС, призначені для інших апаратних платформ (наприклад, запуск ОС для PowerPC на обладнанні з архітектурою x86). Це розширює можливості сумісності, але також залежить від безпеки хостової ОС.

В обох архітектурах – як нативній, так і hosted – кожна гостьова ОС отримує ілюзію наявності власного набору віртуального обладнання, що включає віртуальні CPU, пам'ять, сховище, мережеві адаптери, пристрої відображення та вводу. Деякі гіпервізори також можуть пропонувати

можливості паравіртуалізації для певних типів віртуального обладнання, найчастіше для контролерів сховища та мережевих адаптерів, щоб забезпечити більш високу продуктивність за рахунок прямої взаємодії гостьової ОС зі спеціалізованими інтерфейсами гіпервізора.

Отже, забезпечення безпеки віртуалізованого середовища вимагає глибокого розуміння архітектури віртуалізації, специфічних ризиків, пов'язаних як з типом гіпервізора, так і з взаємодією його компонентів, та послідовного застосування комплексних заходів захисту на всіх рівнях рішення. Вибір між нативною та hosted віртуалізацією має ґрунтуватися не лише на функціональних потребах, але й на ретельному аналізі безпекових компромісів, які несе кожен з підходів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Scarfone K. S., Souppaya M., Wallace A. Guide to Security for Full Virtualization Technologies. NIST Special Publication 800-125. 2011. URL: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-125.pdf (дата звернення: 19.04.2025).
2. What is virtualization? IBM. URL: https://www.ibm.com/think/topics/virtualization (дата звернення: 19.04.2025).