УДК 004.77:004.424
В. В. ВАСИЛЕНКО, аспірант,
Державний університет телекомунікацій, Київ

ВІРТУАЛІЗАЦІЯ ХМАРНИХ ОБЧИСЛЕНЬ І ПИТАННЯ БЕЗПЕКИ В ХМАРНІЙ СИСТЕМІ

Наведено опис сучасних технологій віртуалізації та проаналізовано механізми захисту, необхідні для досягнення надійної ізоляції віртуальних машин, їх опосередкованого спільного використання та налагодження безпечного зв'язку між ними, що має зрештою гарантувати захист приватного трафіку у віртуальних мережах.
Ключові слова: хмарні технології; інформаційні мережі; інформаційна безпека; віртуальна машина.

ВСТУП

Постановка задачі. Хмарні технології виступають сьогодні як представники потужної обчислювальної парадигми. Ідеться про засади нового покоління мережної обчислювальної системи (*Infrastructure as a Service* — **IaaS**), яка підтримує програмне і апаратне забезпечення власних ресурсів, а також надання різноманітних інтернет-послуг. Проте з міркувань безпеки користувачам заборонено ухвалювати хмарні рішення щодо багатьох критично важливих бізнес-обчислень. Адже через спільне використання ресурсів хмари, розрахованих на багатьох користувачів, постають значні загрози її безпеці. Тому останніми роками ця проблематика привертає до себе дедалі більшу увагу науковців-теоретиків і розробників обладнання.

Аналіз останніх досліджень і публікацій. Сучасний стан щодо формування методів аналізу і синтезу хмарних технологій нерозривно пов'язаний із працями таких учених, як O. Sheyner, P. Ammann, X. Ou, L. Wang, H. Poolsappasit, A. Рой.

Мета статті — подати огляд сучасних технологій віртуалізації, спрямованих на захист і забезпечення безпеки приватного трафіку, притаманного віртуальним мережам.

ОСНОВНА ЧАСТИНА

Загальні положення

Віртуалізація являє собою основну технологію хмарних обчислень. Обчислювальна потужність, як і спосіб зберігання інформації в мережі, підлягає віртуалізації для спільного використання в системі IaaS. Ця важлива технологія перетворює абстрактну інфраструктуру й ресурси на такі, що доступні для користувачів у вигляді *ізольованих віртуальних машин* (*VM*) та *віртуальних мереж* (*VNs*). А проте, вона підвищує вразливість системи до атак, оскільки всі користувачі хмари поділяють між собою, а можливо, і нападниками наявні в ній ресурси.

Механізм захисту дає змогу забезпечувати суворо ізоляцію віртуальних машин одна від одної у процесі опосередкованого спільного їх використання, підтримуючи безпечний зв'язок між ними. Окрім того, постає потреба в технологіях із запобігання та виявлення аномального трафіку і захисту нормального трафіку у *VNs*.

Віртуалізація на основі гіпервізора

Гіпервізор, відомий також як *монітор віртуальних машин* (*VMM*), являє собою невеликий фрагмент програмного або програмно-апаратного забезпечення, який працює поверх апаратного забезпечення машини.

Основні функції гіпервізора такі:

- виявлення пасток і реагування на захищені або привілейовані операції, виконувані кожною віртуальною машиною;
- диспетчеризація.

Зауважимо, що адміністративна операційна система (ОС), відома також як *домен привілеїв* (наприклад, dom0 у гіпервізорі Xen), працює поверх гіпервізора так само, як віртуальні машини, відповідаючи за управління віртуальними машинами на одному сервері та працюючи з гіпервізором [1].

Існують два типи гіпервізорів: тип I і тип II. Гіпервізор типу I має бути запущений над апаратними засобами для управління ними та гостьовою ОС. Він відповідає також за більшість комунікацій між усіма гостьовими ОС і апаратними засобами. До представників цього типу окрім згадуваного вже гіпервізора Xen належать VMware ESX Server і Microsoft Hyper-V.

Гіпервізор типу II працює як додаток у рамках хоста ОС, відповідаючи за надання драйверів введення/виведення і управління гостьовою ОС віртуальних машин.

Наприклад, VMware Workstation, VMware Server і VirtualBox — представники архітектури віртуалізації на основі гіпервізора типу II.

Повна віртуалізація

Така віртуалізація передбачає, що гіпервізор містить код, який при потребі емулює базове обладнання, дозволяючи немодифікованим ОС працювати поверх гіпервізора [1]. Одне з відомих програмних забезпечень повної віртуалізації —

VMWare ESX Server — використовує версію Linux (відому як ConsoleService) у ролі своєї адміністративної ОС. Гіпервізор на сервері VMWare ESX, відомий під назвою VMkernel, являє собою гіпервізор товстого шару, якому належить драйвер апаратного забезпечення VMs для кожної віртуальної машини.

На базі повної віртуалізації немодифікована ОС виконує програму користувача, що емулює машину, на якій працює гостьова ОС. Як перевагу цього підходу слід розглядати той факт, що віртуалізована архітектура може повністю відрізнятися від архітектури приймальної. Наприклад, QEMU може імітувати процесор MIPS на хості IA-32 та багато інших чипів [2].

Паравіртуалізація

Тип віртуалізації Xen означає, що створюється тонкий і компактний гіпервізорний шар, який працює безпосередньо поверх апаратного забезпечення, а також надає послуги віртуалізованій ОС. Цей тонкий шар гіпервізора, на відміну від товстого шару, характеризується повною віртуалізацією. Згідно з Xen тільки гіпервізор володіє повними правами, причому він покладається на надійність оцінки ОС, щоб забезпечити апаратні драйвери, ядро та призначений для користувача простір. Відповідний домен управління — згадуваний вже домен 0 (dom0), використовує власне ядро Linux для підтримання свого адміністративного середовища.

Домен 0 як перший домен, створений автоматично при завантаженні системи, має особливі привілеї управління, делеговані гіпервізору. Цей привілейований домен дозволяє гіпервізору отримувати доступ до пристроїв і виконувати функції управління. Усі інші гостьові домени (domUs) не мають привілеїв і перебувають під управлінням dom0. Апаратне середовище для всіх гостей не змодельовано, воно функціонує в їхніх власних ізольованих доменах так, неначе кожний гість працює в окремій системі. Проте гостьова ОС має бути спеціально модифікована, щоб працювати в цьому середовищі.

Паравіртуалізація може забезпечити підвищення продуктивності порівняно з іншими підходами, оскільки операційна система модифікації дозволяє безпосередньо зв'язуватися з гіпервізором і, отже, не зазнає жодних накладних витрат, пов'язаних з емуляцією, що потрібно для інших видів віртуалізації на базі гіпервізора.

Віртуалізація на рівні ядра

Для цього типу віртуалізації гіпервізор не потрібен, але він працює на спеціально модифікованому ядрі Linux, що містить розширення, призначене для управління кількома віртуальними машинами, кожна з яких включає гостьову.

Як приклад технологій віртуалізації на рівні ядра згадаємо користувальницький режим Linux (UML) і режим VM на основі ядра (KVM). Режим UML підтримувався в ядрах Linux досить довгий час, але вимагав спеціального складання ядра Linux для гостьових ОС. Саме тому було введено KVM 2.6.20. Режим UML не потребує будь-якого окремого адміністративного програмного забезпечення для виконання відповідних дій або управління віртуальними машинами, що може бути здійснено з командного рядка Linux. Режим KVM використовує драйвер пристрою в ядрі хоста для зв'язку між основним Linux ядром і віртуальною машиною, вимагаючи підтримки процесорів із метою віртуалізації (Intel VT або AMD-V Pacifica) та використовуючи модифікований процес QEMU як дисплей виконання контейнера для своїх віртуальних машин. Багато в чому віртуалізація KVM на рівні ядра є спеціалізованою версією повної віртуалізації, де ядро Linux використовується як гіпервізор [1].

Віртуалізація з апаратною підтримкою

Віртуалізація з апаратною підтримкою — це спосіб підвищення ефективності апаратної віртуалізації. Вона включає в себе використання спеціально розроблених процесорів і апаратних компонентів, які допомагають поліпшити експлуатаційні характеристики в гостьовому середовищі [1]. Гіпервізор на основі систем, таких як Xen і VMWare ESX Server, а також технології віртуалізації на рівні ядра, такі як KVM, можуть скористатися апаратною підтримкою віртуалізації. Останнє покоління Intel (Intel-VT) і AMD (AMD-V) процесорів підтримують апаратну віртуалізацію. Віртуальні машини в середовищі віртуалізації здатні працювати на немодифікованих операційних системах, оскільки гіпервізор може використовувати підтримку обладнання для віртуалізації та обробки привілейованих і захищених операцій та запитів доступу до обладнання, а також спілкуватися з віртуальними машинами та керувати ними.

Віртуальні мережні архітектури

Мережа є важливим ресурсом обчислювальної і комунікаційної системи. Хмарна система забезпечує не просто надання обчислювальних послуг або ресурсів, вона також слугує каналом зв'язку між усіма віртуальними машинами на одному сервері або на зовнішніх мережних пристроях. Важливе питання полягає в тому, як убезпечити й ізолювати ці канали зв'язку для кожної окремої VM на одному сервері. Далі розглянемо деякі з існуючих архітектур віртуальних мереж, що їх використовують постачальники хмарних платформ.

LinuxEthernetBridge

LinuxEthernetBridge — це спосіб з'єднання двох Ethernet-сегментів разом із протоколом. Пакети

передаються на основі Ethernet-адреси, а не IP-адреси (наприклад, маршрутизатор). Оскільки пересилання здійснюється на рівні 2, всі протоколи можуть іти прозоро через міст [3]. Код моста Linux реалізують стандарти ANSI/IEEE 802.1d [4]. Спочатку LinuxEthernetBridge було задіяно в Linux 2.2, причому код для мостів було інтегровано в серії ядер 2.4 і 2.6.

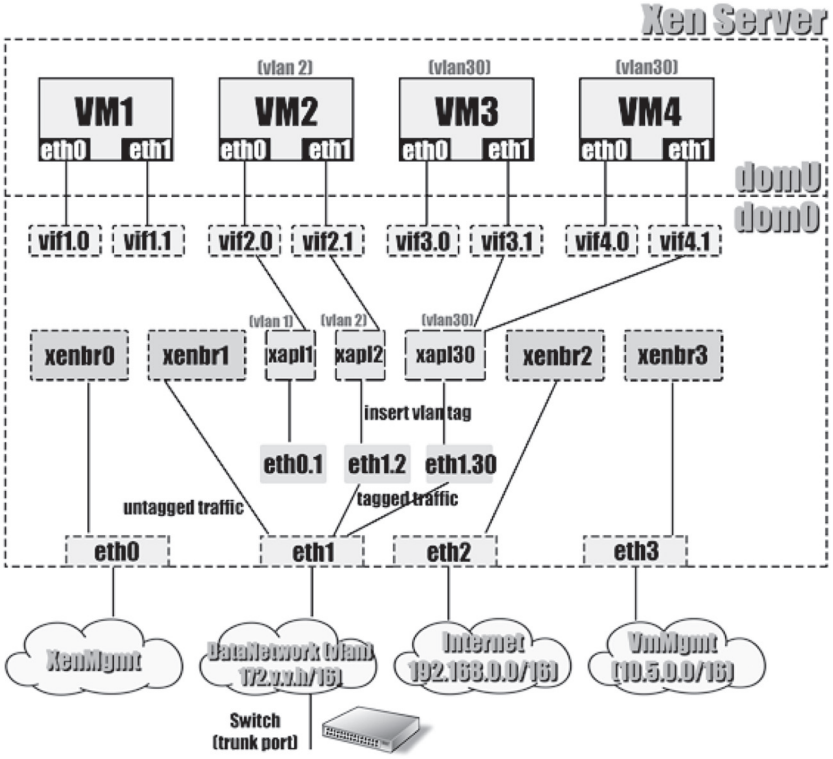
У хмарній системі зв'язок відбувається за участю віртуальних машин на одному хості або віртуальних машин, розташованих на різних комп'ютерах. Цей трафік проходить через віртуальні мережі до віртуального мережного інтерфейсу віртуальної машини або фізичного мережного інтерфейсу хоста. У віртуальній мережі системи Xen гіпервізор підтримує два режими роботи віртуальних мереж: режим моста і режим маршруту. У режимі моста dom0 створює програмне забезпечення Ethernet. При цьому міст є віртуальним інтерфейсом віртуальної машини для пересилання трафіку. Окрім того, він надає віртуальний інтерфейс віртуальної машини до фізичного інтерфейсу на хості для зв'язку із зовнішньою мережею та інтернетом.

У режимі маршруту dom0 створює таблицю маршрутизації, яка включає в себе набір MAC і IP-адрес, заздалегідь забезпечених зв'язком точка-точка між dom0 і кожною віртуальною машиною. В обох режимах dom0 створює віртуальний інтерфейс (vif) для кожного мережного інтерфейсу віртуальної машини. Наприклад, vif1.0 прикріплюється до eth0 з VM1, vif2.1 прикріплюється до eth1 з VM2, і т. д., що ілюструє рисунок.

Конфігурації VLAN

Для управління віртуальними машинами і надання виділених каналів зв'язку для групи віртуальних машин віртуальна локальна мережа (VLAN) здійснює стандартизацію в хмарі. Конфігурація VLAN у системі Xen контролюється та перебуває під управлінням dom0. Структуру VLAN в хмарній системі унаочнює рисунок. Для кожної VLAN dom0 створює унікальний міст XAPI # і відповідний віртуальний інтерфейс eth1. #. Символ # означає ідентифікатор карти VLAN і eth1 на фізичному порті Ethernet-хоста.

Усі мости в системі Xen ґрунтуються на реалізації стандартного моста Linux. XAPI # призначається тільки для внутрішнього використання мережі в dom0. Коли dom0 присвоює VLAN для VM, він пов'язує віртуальний інтерфейс (vif #. #) віртуальної машини до XAPI моста VLAN. Відеозаписи всіх віртуальних машин на віртуальній локальній мережі підімкнено до одного моста. Коли VLAN трафік приходить до моста, він спрямовується до відповідного віртуального інтерфейсу ETH #. #. Віртуальний інтерфейс потім вставляє VLAN тег у кадрі Ethernet відповідно до протоколу 802.1. Теги VLAN трафіку проходять через фізичний порт Ethernet із dom0, передаючи в собі порт зовнішнього фізичного комутатора. Якщо фізичний комутатор дозволяє порту зовнішньої лінії не встановлювати жодного рідного ідентифікатора VLAN (за замовчуванням VLAN ID) або встановити за замовчуванням VLAN ID у невикористовувану VLAN ID, позначений трафік буде проходити через комутатор і залишатиме інформацію тега



Налаштування віртуальної мережі з LinuxBridge

недоторканою. Ідентифікатор VLAN трафіку передається ще до однієї VLAN на інший сервер VPS на основі технології Xen або навіть на віддалений сайт.

Порт, позначений VLAN трафіком, обробляється dom0, а пакети передаються до відповідного віртуального інтерфейсу Eth #. # на підставі VLAN ID. Цей віртуальний інтерфейс буде «здирати» ідентифікатор мітки, а потім передавати пакети тільки за призначенням. Таким чином, призначення мережі VLAN, установлення міток і нормалізація — усе виконується в dom0. VM не знає, до якого VLAN вона належить, і їй не дозволено змінити VLAN тег. Проте в цьому разі можна не маркувати трафік із VLAN-VM, хоча цей тип трафіку буде підмикатися відразу до іншого типу моста xenbr1 і до фізичного порту Ethernet з dom0.

Постачальник послуг, як правило, налаштовує мережні ресурси з різними мережами VLAN для виділення та поділу каналів зв'язку, а також трафіку між різними групами клієнтів у хмарній системі. VLAN забезпечує механізм поділу мережних ресурсів на кілька неперетинних логічних, широкомовних доменів [5]. Трафік може переходити від однієї VLAN до іншої тільки через маршрутизатор. Утім VLAN у хмарному середовищі стикається з багатьма уразливостями від віртуальних машин і серверів. Водночас традиційні питання мережної безпеки не втрачають актуальності.

Позитивні характеристики та недоліки

Linux на базі платформи хмари з убудованою підтримкою LinuxEthernetBridge виступає як комутатор програмного забезпечення для віртуальних машин (на одному сервері є XenServer і KVM компанії Citrix). Міст Linux більш потужний, ніж простий апаратний міст, оскільки він може фільтрувати й формувати трафік. Його легко налаштувати за допомогою команди brctl. Нові можливості моста Linux не припиняють розвитку, можливі майбутні удосконалення: призначені для користувачів простору STP фільтрації, інтерфейс Netlink для управління мостами, підтримка RSTP\MSTP та інші розширення 802.1d STP.

Цей трафік передається безпосередньо з віртуальної машини VM, причому він ніколи не подорожує по фізичному проводу, тому мережні адміністратори не мають змоги контролювати його [6].

Open vSwitch

Відкриття vSWITCH (OVS) [7] являє собою OpenFlow на основі перемикача багатошарового програмного забезпечення, із ліцензією на відкритий вихідний код Apache 2. Це перемикач програмного забезпечення, який слугує для реалізації багатьох хмарних систем, наприклад Citrix XenServer [8] і OpenStack [9]. OVS надає стандартні інтерфейси управління і видимість віртуального мережного рівня. Його було розроблено для підтримки між кількома фізичними серверами. Усі розподілені віртуальні комутатори, такі як NOX/POX, RYU тощо, централізовано керуються мережним контролером на основі OpenFlow. OVS підтримує багато технологій Linux на основі віртуалізації, включаючи Xen, XCP, KVM і VirtualBox.

Open vSWITCH використовує граничний комутатор, який наближає до розв'язання проблеми віртуального мережного управління. Відповідний підхід використовує переваги, що випливають із наявності моста гіпервізора, оскільки компонент гіпервізора може безпосередньо пов'язувати мережні пакети з віртуальними машинами та їх конфігураціями. Граничні комутатори розширюють можливості щодо видимості й контролю, доступні раніше тільки в корпоративних комутаторах високого класу. Вони підвищують видимість між трафіком VM завдяки стандартним методам, таким як NetFlow і віддзеркалення. Додаткові крайові комутатори здійснюють політику трафіку для забезпечення безпеки та якості в обслуговуванні (QoS) [6]. Для того щоб керувати численними перемикачами, сучасні крайові комутатори підтримують централізовану конфігурацію політики. Це дозволяє адміністраторам управляти багатьма мостами на окремих гіпервізорах. Політика та конфігурації централізовано поширюються на віртуальні інтерфейси, мігруючи з їхніми віртуальними машинами.

Open vSWITCH сумісний із мостом Linux, причому він використовує багато Linux на основі гіпервізора для граничного комутатора. Це дозволяє йому бути заміненим у багатьох віртуальних середовищах. OVS надає безліч функцій мережного управління та моніторингу. Щодо видимості монітора, то він підтримує NetFlow, SFlow, протоколи дзеркального відображення (SPAN/RSPAN/ERSPAN). OVS може бути налаштований для перенесення моніторингу трафіку на віддалений колектор або аналізатор. Керованість OVS забезпечує централізоване управління по протоколу OpenFlow [10]. Це те саме, що й комутатор OpenFlow під контролем NOX на основі контролера для режиму руху потоку. Від контролера OVS системний адміністратор проводить список контролю доступу та політики QoS.

Для функцій переадресації OVS підтримує багато протоколів, зокрема **LACP** (*Link Aggregation Control protocol*), з'єднаних портів для балансування навантаження, 802.1Q модель VLAN із магістральним доступом до портів, 802.1ag управління несправностями підімкнення, а також кілька протоколів **GRE** (*Generic Routing Encapsulation*). За протоколами GRE для двох просторово рознесених мостів різної хмарності сервера віртуальні машини, підімкнені до цих мостів, можуть утворити двошарові з'єднання, розташовані навіть у різних

місцях і такі, що мають різні загальнодоступні IP-адреси.

Проблеми безпеки у віртуальному середовищі

У хмарних системах джерелом ураження для каналу зв'язку можуть бути віртуальні машини, сервер хмари, а також процес підімкнення до фізичної мережі.

Розглянемо головні механізми ураження.

Атаки з віртуальної машини. Віртуалізація виступає ключовою технологією в хмарних обчисленнях. При цьому головний осередок проблем щодо безпеки хмарної системи припадає на VM з'єднання [11]. Найбільшу небезпеку становлять такі ситуації.

• VM Hopping — процес переходу з однієї VM на іншу. Тоді зловмисник, перебуваючи на одній VM, може отримати несанкціонований доступ через іншу VM;

• VM Escape — доступ до VM через гіпервізор (VMM) із нападом на іншу частину віртуальних машин. Зловмисник, отримавши доступ до вузла запуску кількох віртуальних машин, може отримати доступ до ресурсів, які є спільними для інших віртуальних машин;

• VM mobility — вміст VM зберігається у файлі зображення на гіпервізорі. Зазначений файл можна перемістити (або скопіювати) в інше місце. Зловмисник, модифікувавши вміст цього файла, зможе змінити діяльність віртуальної машини;

• VM Template — VM клонують за допомогою шаблона, аби прискорити створення хмарної системи. Точніше, якщо шаблон використовують, зловмисник може змінити налаштування деякої VM, щоб стежити за всіма віртуальними машинами.

Атаки на основі віртуальної мережі. Обидва режими моста і режим маршруту dom0 відіграють тут певну роль.

• Sniffing: у режимі моста VM здатна «нюхати» віртуальну мережу на тому самому мості, використовуючи Sniffing інструменти, такі як Wireshark.

• Spoofing: VM-нападниця може скласти протокол дозволу адрес (ARP), аби перехоплювати пакети та прослуховувати трафік потерпілої VM.

На додаток до можливих засобів ураження віртуальної мережі слід згадати таку дію, як компрометація dom0 (Compromised dom0). Це важливе питання безпеки в мережі хмарної системи. Адже dom0 може контролювати всі зв'язки між VM і підмикатися до зовнішньої мережі.

Тільки-но dom0 скомпрометовано, зловмисник може змінити трафік у віртуальній мережі і заволодіти важливою інформацією.

Зазначені атаки тягнуть за собою такі загрози:

• усі віртуальні машини контролюються VM-нападницею;

• зв'язок між віртуальними машинами контролюється зловмисниками;

• Denial of Service (DoS) проти хмарних сервісів.

Безпека VLAN

Для підвищення безпеки VLAN реалізовано у плані посилення ізоляції мережі, а також розширення можливостей управління системою. Проте з реалізацією VLAN кожне повідомлення, як і раніше, може охопити всі частини однієї і тієї самої мережі. Це означає, що повідомлення можуть бути прочитані будь-яким хостом на тій самій VLAN. Тоді зловмисник може пасивно підслуховувати пакети, що проходять через мережу. Окрім того, слід пам'ятати, що Ethernet являє собою систему мовлення, яка не передбачає механізму для перевірки справжності відправника. Це дозволяє зловмисникові генерувати нові пакети або відтворювати раніше підслухані [12].

Найчастіше відомі атаки проти кінцевих хостів у мережі рівня 2 спираються на Medium Access Control (MAC), Addressspoofing чи Address Resolution Protocol (ARP).

Це формує основу багатьох атак, таких як DoS і Man-In-The-Middle. Інші можливі атаки у VLAN на основі мережі [13] включають у себе:

• MAC flooding attack — це напад, пов'язаний з обмеженням перемикачів, що працюють, і мостів. Вони володіють кінцевою таблицею апаратних засобів для зберігання вихідних адрес усіх прийнятих пакетів. Коли ця таблиця заповнюється, трафік, спрямований на адреси, які не можуть бути вилучені, буде постійно повний;

• 802.1Q tagging attack — атака, що відбувається через неправильні настроювання перемикачів, які встановлюють порт комутатора як небажаний порт з'єднувальної лінії. Таку ситуацію називають VLAN leaking, що дозволяє користувачеві VLAN дістати несанкціонований доступ до іншої VLAN;

• Double-Encapsulated 802.1Q/ VLAN атаки — атаки, що відбуваються через неправильне настроювання перемикачів, які встановлюють рідний ідентифікатор VLAN магістрального порту, так само як VLAN ID порту доступу зловмисника. Зловмисник може інкапсулювати цільовий ідентифікатор VLAN як внутрішній шар протоколу 802.1Q. Після того як зовнішній тег відігнано, відбувається неправильний вибір параметра комутатора, а через це комутатор просто пересилає пакети за призначенням підробленого внутрішнього тега. Таким чином, зловмисник на VLAN може отримати несанкціонований доступ до іншої VLAN, що згадується як VLAN Hopping;

• ARP-атаки — це атаки, пов'язані з ARP-запитом і відповіддю, які несуть інформацію про ідентичність шару 2 (MAC-адресу) та ідентифікатор рівня 3 (IP-адресу) хоста. Зловмисник може обдурити комутатор у переспрямовуванні пакетів

на пристрій або хост в іншій мережі VLAN, посилаючи ARP-пакети, що містять підроблені тотожності.

Варто наголосити, що в тій самій VLAN так звані ARP-атаки є дуже ефективний спосіб обдурювання кінцевих станцій або маршрутизаторів для вивчення підробки ідентичних пристроїв. Це може дозволити зловмисникові поставити людину як посередника та виконавця своїх намірів.

ВИСНОВКИ

Віртуалізація є основною технологією хмарних обчислень, згідно з якою обчислювальна потужність мережі підлягає віртуалізації для спільного використання в системі IaaS. Ця важлива технологія перетворює абстрактні інфраструктури та ресурси на доступні для користувачів у ролі ізольованих віртуальних машин (VM) та віртуальних мереж (VNs). Проте віртуалізація підвищує уразливість і можливість атак у системі, оскільки всі користувачі у хмарі поділяють наявні ресурси з іншими користувачами або навіть із нападниками. Механізм захисту слугує для забезпечення суворої ізоляції, опосередкованого спільного використання та безпечного зв'язку між віртуальними машинами. При цьому постає потреба в технологіях для виявлення аномального трафіку та захисту нормального трафіку у віртуальних мережах.

Отже, забезпечення безпеки та захист приватного трафіку у віртуальних мережах, із запобіганням зловмисному трафіку в спільних ресурсах — це головна проблематика безпеки в хмарній системі.

Список використаної літератури

1. ***Hagen, W.*** *Professional Xen Virtualization / W. Hagen.— Wiley Publishing, Inc., 2008.*

2. ***Takemuraand, C.*** *The Book of XEN / C. Takemuraand, L. Crawford.— No Starch Press, 2009.*

3. ***The Linux*** *Foundation [Електронний ресурс] // Linuxbridge, 2012.— Режим доступу:*
http://www.linuxfoundation.org/collaborate/workgroups/networking/bridge;

4. ***IEEE 802.1d*** *standard, 2012 [Електронний ресурс].— Режим доступу:*
https://standards.ieee.org/about/get/802/802.1.html;

5. ***Saha, A.*** *Thinking out side the box: extending 802.1x aut hentication to remote «splitter» ports by combining physical and data linklayer techniques / A. Saha and M. Molle // 28th Annual IEEE International Conference on Local Computer Networks, 2003.LCN'03. Proceedings, 2003.— P. 324–333.*

6. ***Pettit, J.*** *Virtual switching in anera of advanced / [J. Pettit a. o.] // 2nd Workshop on Data Center — Converged and Virtual Ethernet Switching (DC-CAVES), ITC.— 2010.— Vol. 22.*

7. ***Open*** *Switch project [Електронний ресурс].— Режим доступу:*
http://openvswitch.org, May 2012.

8. ***Citrix*** *Xen Server [Електронний ресурс].— Режим доступу:*
http://www.citrix.com/xenserver

9. ***Open*** *Stack [Електронний ресурс].— Режим доступу:*
http://www.openstack.org/

10. ***Open*** *flow [Електронний ресурс].— Режим доступу:*
http://www.openflow.org/wp/learnmore/

11. ***Wu, H.*** *Network security for virtual machine in cloud computing / [H. Wu, Y. Ding, C. Winer and L. Yao] // Computer Sciences and Convergence Information Technology (ICCIT), 2010. 5th International Conferenceon, Dec. 2010.— P. 18 –21.*

12. ***Kumar, S.*** *Service cloaking and authentication at data link layer / S. Kumar // 2nd International Symposium on Advanced Networks and Telecommunication Systems, 2008. ANTS'08, 2008.— P. 1–3.*

13. ***Cisco*** *Systems. Virtuallan security best practices, 2002 [Електронний ресурс].— Режим доступу:*
http://www.cisco.com/warp/public/cc/pd/si/casi/ca6000/prodlit/vlnwpwp.pdf

Рецензент: доктор техн. наук, професор **В. В. Вишнівський**, Державний університет телекомунікацій, Київ.

В. В. Василенко

ВИРТУАЛИЗАЦИЯ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ И ВОПРОСЫ БЕЗОПАСНОСТИ В ОБЛАЧНОЙ СИСТЕМЕ

Приведено описание современных технологий виртуализации и проанализированы механизмы защиты, необходимые для обеспечения строгой изоляции, опосредованного совместного использования и безопасной связи между виртуальными машинами с целью обеспечения безопасности частного трафика в виртуальных сетях.

Ключевые слова: облачные технологии; информационные сети; информационная безопасность; виртуальная машина.

V. V. Vasylenko

VIRTUALIZATION AND CLOUD COMPUTING SECURITY ISSUES IN CLOUD SYSTEM

Description of the currently existing types of virtualization technologies, and the analysis of defense mechanisms that are necessary to ensure strict isolation mediated sharing and secure communications between virtual machines to provide security and protection of private traffic on the virtual networks.

Keywords: cloud; information networks; information security; virtual machine.