DOI: 10.32347/2412-9933.2020.42.125-131
УДК 69:002;69.059

Терентьєв Олександр Олександрович
Доктор технічних наук, професор, професор кафедри інформаційних технологій проєктування та прикладної математики, *orcid.org/0000-0001-6995-1419*
Київський національний університет будівництва і архітектури, Київ
Доля Олена Вікторівна
Кандидат фізико-математичних наук, доцент, доцент кафедри інформаційних технологій проєктування та прикладної математики, *orcid.org/0000-0003-2503-2634*
Київський національний університет будівництва і архітектури, Київ
Лященко Тамара Олексіївна
Старший викладач кафедри інформаційних технологій, *orcid.org/0000-0001-9092-0297*
Київський національний університет будівництва і архітектури, Київ
Кузьмінський Олег Вікторович
Студент, *orcid.org/0000-0002-4528-9210*
Київський національний університет будівництва і архітектури, Київ

ДІАГНОСТУВАННЯ МЕРЕЖ ТА ПРОТИДІЯ МЕРЕЖЕВИМ ЗАГРОЗАМ

Анотація. *Інформаційна безпека у хмарних сервісах – одне з найважливіших питань, яке залучало багато досліджень та розробок за останні кілька років. Зокрема, зловмисники можуть досліджувати вразливості хмарної системи та компроментувати віртуальні машини для розгортання подальших масштабних розподілених зловмисних дій (DDoS). DDoS-атаки зазвичай включають дії на ранній стадії, такі як багатоступінчаста експлуатація, сканування вразливості на низьких частотах та компроментація ідентифікованих вразливих віртуальних машин як підпорядкованих, а також DDoS-атаки через компроментовані віртуальні машини. У хмарній системі, особливо хмарах інфраструктури як послуга (IaaS), виявити атаки таких підпорядкованих машин надзвичайно важко. Це відбувається тому, що користувачі хмари можуть встановлювати вразливі програми на своїх віртуальних машинах. Щоб уникнути компроментації уразливих віртуальних машин у хмарі, запропоновано багатофазний механізм виявлення, вимірювання та вибору контрзаходу, який базується на аналітичних моделях, заснованих на налагоджених контрзаходах на основі віртуальної мережі. Результати оцінювання системи та безпеки демонструють ефективність запропонованого рішення.*

Ключові слова: хмарні розрахунки; хмарні сервіси; інформаційна безпека; DDoS; виявлення вторгнень

Актуальність та аналіз проблеми

Останні дослідження засвідчили, що користувачі, які переходять до хмари, вважають безпеку найважливішим фактором. Нещодавнє опитування Альянсу захисту безпеки (CSA) показує, що серед усіх проблем безпеки зловживання та шкідливе використання хмарних обчислень вважається головною загрозою безпеці [1], в якій зловмисники можуть використовувати вразливості в хмарах, хмарні ресурси системи для розгортання атак. У традиційних центрах опрацювання даних, де системні адміністратори мають повний контроль над хост-машинами, вразливості може виявляти та виправляти системний адміністратор централізовано. Однак виправлення відомих вразливостей у безпеці в хмарних центрах опрацювання даних, де користувачі хмар, як правило, мають право керувати програмним забезпеченням, встановленим на керованих віртуальних машинах, може не працювати ефективно, а також порушувати угоду про рівень обслуговування (SLA). Крім того, користувачі хмари можуть встановлювати вразливе програмне забезпечення на своїх віртуальних машинах, що по суті сприяє пробілам у хмарній безпеці.

Мета статті

Завдання полягає у створенні ефективної системи виявлення вразливих ситуацій / атак, а також реагування на них для точного виявлення атак та мінімізації впливу порушення безпеки на користувачів хмари. У роботі [2]М. Армбруста та ін. вирішено, що захист "Безперервність бізнесу та

доступність послуг" від відключення послуг є однією з головних проблем у хмарних обчислювальних системах. У хмарній системі, де інфраструктуру поділяють потенційно мільйони користувачів, зловживання та нечесне використання спільної інфраструктури приносить користь зловмисникам, які використовують вразливі ситуації хмари та її ресурсу для розгортання атак більш ефективними способами [3]. Такі атаки є більш ефективними у хмарному середовищі, оскільки користувачі хмари зазвичай діляться обчислювальними ресурсами, наприклад, підключаються через один і той же комутатор, обмінюються з тими самими сховищами даних та файловими системами, навіть з потенційними зловмисниками [4]. Подібна установка для віртуальних машин у хмарі (наприклад, методи віртуалізації, встановлене вразливе програмне забезпечення, мережа тощо) залучає зловмисників для компрометації декількох віртуальних машин.

Виклад основного матеріалу

Моделі

У цьому розділі описано, як моделювати загрози безпеці та вразливості у віртуальній мережевій системі та запропоновано модель захисту віртуальних машин на основі підходів до конфігурації віртуальної мережі.

Потокова модель

У моделі атаки припущено, що зловмисник може бути розташований як зовні, так і всередині системи віртуальної мережі. Основна мета зловмисника – використовувати вразливі віртуальні машини та компрометувати їх як підпорядковані. Модель захисту зосереджена на рішеннях виявлення та переналаштування на основі віртуальної мережі для підвищення стійкості до скануванняпотенційних вразливих машин. Робота не включає IDS (IntrusionDetectionSystem) на основі хостів і не стосується способів опрацювання зашифрованого трафіку для виявлення атак.

Дане рішення може бути розгорнуто в хмарній мережі “Інфраструктура як послуга” (IaaS), а також припущено, що постачальник хмарних послуг (CSP) є доброякісним, і не може стати причиною атаки. Такожприпущено, що користувачі хмарних сервісів можуть безкоштовно встановлювати будь-які операційні системи або додатки, які вони хочуть, навіть якщо така дія може ввести вразливість до їх керованих віртуальних машин. Фізична безпека хмарного сервера виходить за межі цієї роботи. Припустимо, що гіпервізор захищений від будь-яких уразливостей. Питання зловмисного орендаря, який вирвався з рамок власної віртуальної машини та отримав доступ до фізичного сервера, виходить за межі цієї роботи.

Модель графа атаки

Граф атаки – це інструмент моделювання для ілюстрації всіх можливих багатоступеневих шляхів атаки з декількома хостами, які мають вирішальне значення для розуміння загроз і для вирішення відповідних контрзаходів [5]. У графу атаки кожен вузол являє собою або передумову, або наслідок експлуатації. Дії не обов'язково є активною атакою, оскільки звичайні протокольні взаємодії також можуть використовуватися для атак. Граф атак корисний для виявлення потенційних загроз, можливих атак та відомих уразливостей у хмарній системі.

Оскільки граф атаки містить детальну інформацію про всі відомі вразливості в системі та інформацію про підключення, отримуємо повну картину поточної ситуації в системі, де можна передбачити можливі загрози та атаки, зіставляючи виявлені події чи дії. Якщо подію визнають потенційною атакою, можна застосувати конкретні контрзаходи, щоб пом'якшити її вплив або вжити заходи для запобігання пошкодження хмарної системи.

Визначення 1 (*Граф сценарію атаки*). SAG – це кортеж SAG = (V, E), де

1. $V = N_C \cup N_D \cup N_R$ позначає набір вершин, що включають три типи, а саме: вузол сполучення N_C для подання експлуатувати, вузол диз'юнкції N_D для позначення результату експлуатації, кореневий вузол N_R для показу початкового кроку атаки сценарію.

2. $E = E_{pre} \cup E_{post}$ позначає набір спрямованих ребер. Край $e \in E_{pre} \subseteq N_D \times N_C$ являє собою N_D повинні бути задоволені для досягнення N_C. Край $e \in E_{post} \subseteq N_C \times N_D$ означає, що наслідок, показаний N_D, можна отримати, якщо N_C задоволено.

Вузол $V_C \in N_C$ визначається як три кортежі (хости, вразливості, сповіщення), що представляють набір IP-адрес, інформація про вразливість, таку як CVE [6]. N_D поводиться як логічний АБО операції і містить детальну інформацію про результати дій. N_R являє собою кореневий вузол графа атаки сценарію. Для співвідношення сповіщень використовують підхід, що описано в [7], та визначають граф кореляції сповіщень (ACG) для відображення сповіщень в ACG до відповідних вузлів у SAG. Щоб слідкувати за ходом атаки, потрібно відстежувати джерела та IP-адреси для атаки діяльність.

Визначення 2 (*Графкореляції сповіщень*). ACG – це три кортежі ACG = (A, E, P), де

1. A – це сукупність зведених попереджень. Попередження $a \in A$, а структура даних (src, dst, cls, ts), що представляють вихідну IP адресу, IP-адресу

призначення, тип сповіщення та часову мітку попередження відповідно.

2. Кожне попередження відображає пару вершин (v_c, v_d) в SAG за допомогою функції map (a), тобто map (a): a → {(v_c, v_d) | (a.src∈vc.Hosts) ∧ (a.dst∈vd.Hosts) ∧ (a.cls = vc.vul)}.

3. E – це сукупність спрямованих ребер, що представляють кореляцію між двома сповіщеннями (a, a) якщо критерії нижче задоволені: i. (a.ts<a.ts) ∧ (a.ts – a.ts<threshold) ii. ∃ (v_d, v_c) ∈ E_{pre}: (a.dst∈v_d.Hosts∧a.src∈v_c.Hosts).

4. P – це набір шляхів в ACG. Шлях S_i⊂ P – це множина пов'язаних сповіщень в хронологічному порядку.

Припущено, що A містить агреговані сповіщення, а не необроблені сповіщення. Сирі сповіщення, що мають однакові IP-адреси джерела та призначення, тип атаки та часову позначку у вказаному вікні, агрегуються як MetaAlerts. Кожна впорядкована пара (a, a') в ACG відображає дві сусідні вершини в SAG з різницею часових позначок у двох попереджувальних межах у межах попередньо визначеного порогу. ACG показує залежність сповіщень у хронологічному порядку, отже можемо знайти пов'язані сповіщення у тому ж сценарії атаки шляхом пошуку шляху оповіщення в ACG. Набір P використовується для зберігання всіх шляхів від кореневого попередження до цільового сповіщення в SAG, а кожен шлях S_i⊂ P являє собою сповіщення, що належать до одного сценарію атаки.

Алгоритм 1

Дано: alertac, SAG, ACG
1: **if** (a_c нове повідомлення) **then**
2: новий вузол $a_c$inACG
3: n_1 ← v_c∈map(a_c)
4: **for all**n_2∈parent(n_1) **do**
5: створити грань (n_2.alert, a_c)
6: **for all**S_iщомістить a **do**
7: **if***a* is останнійелемент in *Si***then**
8: добавитиa_c to *Si*
9: **else**
10: створитигляхS_{i+1} = {subset(*S_i, a), a_c*}
11: **end if**
12: **end for**
13: добавитиa_cдоn_1.*alert*
14: **end for**
15: **end if**
16: **return**S

Модель захисту віртуальних машин

Модель захисту від віртуальних машинскладається з ВМ-профайлера, індексатора безпеки та монітора стану. Визначаємо індекс безпеки для всіх віртуальних машин у мережі залежно від різних факторів, таких як підключення, кількість наявних вразливих місць та їх оцінка. Оцінка впливу вразливості, визначена керівництвом CVSS [8; 9], допомагає судити про вплив конфіденційності, цілісності та доступності вразливості. Метрику підключення віртуальної машини визначають шляхом оцінювання вхідних та вихідних з'єднань.

Визначення 3 (*Стан віртуальної машини*). На основі інформації, зібраної від мережевого контролера, стан ВМ можна визначити таким чином:

1. Стабільний: не існує жодної відомої вразливості для ВМ.

2. Уразливий: наявність однієї або декількох вразливих місць в ВМ, яка залишається невикористаною.

3. Експлуатується: принаймні одна вразливість була використана, а ВМ порушено.

4. Підконтрольна: ВМ знаходиться під контролем нападника.

Система попередження та контрзаходів

Розділ розкриває базову архітектурну модель, а також описує роль та функціонал кожного з модулів хмарної моделі.

Архітектура

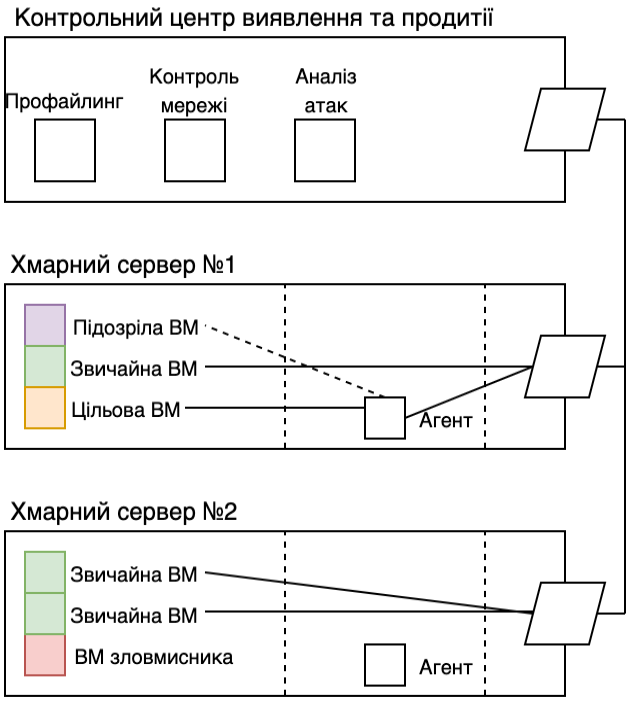

Рисунок 1 – . Модель роботи системи виявлення та протидії у розрізі хмарної мережевої архітектури

Запропонований фреймворк проілюстровано на рис. 1. Він показує структуру в межах одного кластера хмарних серверів. Основними компонентами в цій структурі є розподілений і зважений агент на кожному фізичному хмарному сервері, мережевий контролер, сервер профілювання та аналізатор атаки. Останні три компоненти

розміщені в централізованому центрі управління, підключеному до програмних комутаторів на кожному хмарному сервері (тобто віртуальні комутатори, побудовані на одному або декількох мостових програмних системах Linux). Програмний агент, реалізований на кожному хмарному сервері, підключеному до центру управління через виділений і ізольований захищений канал, який відокремлений від звичайних пакетів даних, використовуючи тунелінгOpenFlow або підхід VLAN. Мережевий контролер відповідає за розгортання контрзаходів на основі рішень, прийнятих аналізатором атаки.

Повідомлення про виявлення вторгнень надсилаються до центру управління, коли виявляється підозрілий або аномальний трафік. Після отримання сповіщення аналізатор атаки оцінює ступінь суворості попередження на основі графіка атаки, вирішує, які стратегії контрзаходу вжити, а потім ініціює його через мережевий контролер. Графік атаки встановлюється відповідно до інформації про вразливість, отриманої як в автономному, так і в режимі реального часу. Сканування в режимі офлайн можна здійснити за допомогою тестів на проникнення та сканування вразливості в режимі реального часу, а також в режимі реального часу може бути запущено мережевим контролером (наприклад, коли нові порти відкриваються та ідентифікуються комутаторами OpenFlow) або коли нові сповіщення створюються агентом. Після виявлення нових уразливостей або розгортання контрзаходів граф атаки буде реконструйований. Контрзаходи ініціюються аналізатором атак на основі результатів оцінювання результатів аналізу ефективності контрзаходів. Потім мережевий контролер ініціює дії контрзаходу шляхом перенастроювання віртуальних або фізичних комутаторів OpenFlow.

Огляд компонентів

Агент

Агент – це програмне забезпечення для виявлення вторгнень, що встановлене на кожному хмарному сервері. Він сканує трафік, що проходить через мости Linux, які контролюють увесь трафік серед віртуальних машин та введення / виведення з фізичних хмарних серверів. Кожен міст утворює ізольовану підмережу у віртуальній мережі та підключається до всіх пов'язаних віртуальних машин. Трафік, генерований від віртуальних машин на дзеркальному мостовому програмному мосту, відображатиметься до конкретного порту на конкретному мосту, використовуючи методи SPAN, RSPAN або ERSPAN. Правила сніфінгуагента були визначені на замовлення відповідно до наших потреб. Ефективніше сканувати трафік в обгортці віртуальної мережі, оскільки весь трафік на хмарному сервері проходить через нього, однак наш дизайн не залежить від встановленої віртуальної машини. Треба зауважити, що якість виявлення оповіщення агенту залежить від впровадження типу агенту, який використовує Snort. Таким чином, помилкова швидкість тривоги індивідуального сповіщення не змінюється. Однак показник помилкової тривоги може бути знижений завдяки нашому дизайну архітектури.

Профілювання віртуальних машин

Віртуальні машини у хмарному сервісі можуть бути профільовані для отримання точної інформації про їх стан, роботи служб, відкриті порти тощо. Одним з головних факторів, що враховує профіль ВМ, є його зв'язок з іншими віртуальними машинами. Будь-яка ВМ, що підключена до більшої кількості машин, є більш важливою, ніж та, що підключається до меншої кількості ВМ, оскільки захоплення сильно зав’язаної ВМ може завдати більше шкоди. Також необхідні знання служб, що працюють на ВМ, щоб перевірити достовірність сповіщень, що стосуються цієї ВМ. Зловмисник може використовувати програму портативного сканування для інтенсивного обстеження мережі для пошуку відкритих портів на будь-якій віртуальній машині. Таким чином, інформація про будь-які відкриті порти ВМ та історія відкритих портів відіграють важливу роль у визначенні того, наскільки вразлива ВМ. Всі ці фактори, що поєднуються, формуватимуть профіль ВМ. Профілі ВМ підтримуються в базі даних і містять вичерпну інформацію про вразливості, попередження та трафік. Дані походять від:

– генератора графіків атак (під час генерації графіка атаки кожна виявлена вразливість додається до відповідного запису ВМ у базі даних);

– агенту (попередження, пов’язане з ВМ, буде записано в базі даних профілю ВМ);

– мережевий контролер: схеми трафіку за участю ВМ засновані на 5 кортежах (MAC-адреса джерела, MAC-адреса джерела, IP-адреса джерела, IP-адреса пункту призначення, протокол). У нас може бути схема трафіку, коли пакети виходять з одної IP-адреси і доставляються на кілька IP-адрес призначення, і навпаки.

Аналізатор атаки

Основні функції системи виконуються аналізатором атак, який включає такі процедури, як побудова та оновлення графу атак, кореляція попередження та вибір контрзаходу. Процес побудови та використання графу атаки сценарію (SAG) складається з трьох етапів: збирання інформації, побудова графа атаки та аналіз потенційного шляху експлуатації. За допомогою цієї інформації шляхи атаки можна змоделювати через SAG. Кожен вузол у графу атаки представляє вчинок

зловмисника. Кожен шлях від початкового вузла до вузла цілі являє собою вдалу атаку. Підсумовуючи, графік атаки будується на основі такої інформації:

– інформація про хмарну систему збирається з контролера вузла. Інформація включає в себе кількість віртуальних машин на хмарному сервері, сервіси, що працюють для кожного віртуального комп'ютера та інформацію про віртуальний інтерфейс (VIF);

– інформація про топологію та конфігурацію віртуальної мережі збирається з мережевого контролера, що включає в себе топологію віртуальної мережі, з'єднання з хостом, підключення VM, IP-адресу кожного VM, MAC-адресу, інформацію про порт та інформацію про потік трафіку;

– інформація про вразливість генерується скануванням вразливості на вимогу (тобто, ініційованим мережевим контролером) та регулярним тестуванням на проникнення з використанням відомих баз даних вразливості, таких як база даних з вразливістю з відкритим кодом (OSVDB) [9], загальні вразливості та Список експозицій (CVE) [6] та Національна база даних про вразливість NIST (NVD) [10]. Аналізатор атаки також опрацьовує кореляцію попередження та операції аналізу. Цей компонент має дві основні функції:

1 – будує граф кореляції сповіщень (ACG);

2 – надає інформацію про загрозу та відповідні контрзаходи мережевому контролеру для конфігурації віртуальної мережі.

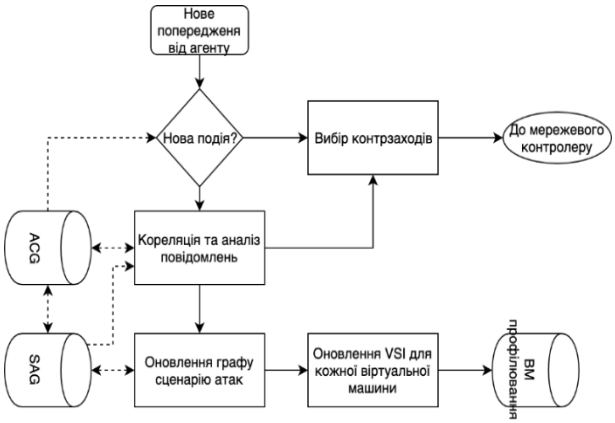

Рисунок 2 – Алгоритм роботи аналізатора атаки

На рис. 2 наведено робочий процес в компоненті аналізатора атаки. Після отримання сповіщення від агента аналізатор сигналів відповідає сигналу в ACG. Якщо попередження вже існує в графіку, і це відома атака (тобто відповідність підпису атаки), то аналізатор атаки виконує процедуру вибору контрзаходу відповідно до алгоритму 2, а потім повідомляє мережевий контролер негайно про розгортання контрзаходів або заходів пом'якшення. Якщо сповіщення нове, аналізатор атаки виконує кореляцію та аналіз попередження відповідно до алгоритму 1 та оновлює ACG та SAG. Цей алгоритм співвідносить кожне нове попередження з відповідним набором кореляції сповіщень (тобто в тому ж сценарії атаки). Вибраний контрзахід застосовується мережевим контролером на основі вираженості результатів оцінки. Якщо попередження являє собою нову вразливість і його немає в графі атаки, то аналізатор атак додає його до графу атаки, а потім реконструює.

Мережевий контролер

Мережевий контролер є ключовим компонентом для підтримки програмованої можливості мереж реалізувати функцію реконфігурації віртуальної мережі на основі протоколу OpenFlow[11]. У системі всередині кожного хмарного сервера є перемикач програмного забезпечення, наприклад, OpenvSwitch (OVS), який використовується як крайній перемикач для VM для обробки трафіку з ВМ та виходу з нього. Зв'язок між хмарними серверами (тобто фізичними серверами) обробляється фізичним перемикачем – OpenFlow (OFS). У роботі було інтегровано функції керування як для OVS, так і для OFS в мережевий контролер, що допомагає хмарній системі встановлювати правила безпеки / фільтрації інтегровано і всебічно. Мережевий контролер відповідає за збирання мережевої інформації поточної мережі OpenFlow та забезпечує введення в аналізатор атаки для побудови графіків атак. Через хмарні модулі внутрішнього виявлення, які використовують DNS, DHCP, LLDP та ініціації потоку [12], мережевий контролер здатний виявити інформацію про мережеве з'єднання з OVS та OFS. Ця інформація включає поточні потоки даних на кожному комутаторі та детальну інформацію про потік, пов'язану з цими шляхами, таку як TCP / IP та MAC-заголовок.

Інформація про зміну мережевого потоку та топології буде автоматично відправлена до контролера, а потім доставлена в аналізатор атаки для реконструкції графіків атаки. Ще одна важлива функція мережевого контролера – допомога модулю аналізатора атаки. Відповідно до протоколу OpenFlow [20], коли контролер отримує перший пакет потоку, він тримає пакет і перевіряє таблицю потоків на відповідність політиці трафіку. У системі мережеве управління також консультується з аналізатором атаки для контролю доступу потоку, встановлюючи правила фільтрації на відповідні OVS та OFS. Після допуску потоку трафіку наступні пакети потоку не обробляються мережевим контролером, а контролюються агентом.

Мережевий контролер також відповідає за застосування контрзаходу від аналізатора атаки. Виходячи з індексу безпеки ВМ та суворості попередження, контрзаходи вибираються та виконуються мережевим контролером. Якщо спрацьовує суворий сигнал і виявляє деякі відомі атаки або ВМ виявляється як підконтрольна, мережевий контролер негайно заблокує ВМ. Сповіщення із середнім рівнем загрози спрацьовує підозріла компрометована ВМ. Контрзаходи в такому випадку – перевести підозрілу ВМ з експлуатованим станом в карантинний режим і перенаправити всі його потоки в режим глибокої перевірки пакетів агентом (DPI). Сповіщення з незначним рівнем загрози може генеруватися через наявність вразливої машини. У цьому випадку, щоб перехопити звичайний трафік ВМ, підозрілий трафік до / від ВМ буде переведений в режим огляду, в якому такі дії, як обмеження його пропускної здатності потоку та зміна конфігурацій мережі будуть вживатися для примушування поведінки розвідки атаки та виділятися.

Висновки

У роботі представлено систему, що пропонується як засіб виявлення та пом'якшення атаки направленого розгалуженого спільного характеру у хмарному віртуальному середовищі. Ця система використовує модель графу атаки для виявлення та прогнозування атак. Пропоноване рішення досліджує, як використовувати програмовані рішення програмних комутаторів на основі підвищення точності виявлення та ураження фаз експлуатації жертв спільних атак. Оцінка продуктивності системи демонструє доцільність роботи системи та показує, що запропоноване рішення може значно знизити ризик експлуатації та зловживання хмарною системою внутрішніми та зовнішніми нападниками.

Така система досліджує лише мережевий підхід IDS для протидії DDoS атакам. Для підвищення точності виявлення потрібні хост-рішення IDS, які повинні бути включені та охопити весь спектр IDS в хмарній системі.

Список літератури / References

1. CSA. Top threats to cloud computing v1.0 // cloudsecurityalliance.org. 2010. URL: https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf (access date: 07.10.2019).

2. Ambrust M., Fox A., "A view of cloud computing" // ACM Commun, Vol. 53, No. 4, 04.2019. pp. 50-58.

3. Joshi B., Vijayan A. Securing cloud computing environment against DDoS attacks // IEEE Int'l Conf. Computer Communication and Informatics. 2012. Vol. 12.

4. Takabi H., Joshu B., "Security and privacy challenges in cloud computing environments" // IEEE Security & Privacy, Vol. 8, 12.2010. pp. 24-31.

5. Ou K., Boyer W. A scalable approach to attack graph generation // ACM conf. on Computer and communications security. 2006. Vol. 13. pp. 336–345.

6. Mitre Corporation. Common vulnerabilities and exposures, CVE // Mitre ORG. 2012. URL: http://cve.mitre.org/ (access date: 21.09.2019).

7. Roshe S., Cheng F., "A new alert correlation algorithm based on attack graph" // Computational Intelligence in Security for Information Systems, Vol. 6694, 03.2011. pp. 58-67.

8. Mell P., Scarfone K. Common vulnerability scoring system (CVSS) // First ORG. 2010. URL: http://www.first.org/cvss/cvss-guide.html (access date: 23.09.2019).

9. O. Database. OpenDatabase [Web resource] // O. Database: [website]. [2012]. URL: Open source vulnerability database (OVSDB) (access date: 01.10.2019).

10. NIST. // National vulnerability database, NVD: [website]. [2012]. URL: http://nvd.nist.gov (access date: 02.10.2019).

11. McKiovn N., Anderson T., "OpenFlow: enabling innovation in campus networks" // SIGCOMM Comput. Commun. Rev., Vol. 38, No. 2, 03.2008. pp. 69-74.

12. Gud N., Conolen T., "NOX: towards an operating system for networks" // SIGCOMM Comput. Commun. Rev, Vol. 38, No. 3, 11.2008. pp. 105-110.

Стаття надійшла до редколегії 03.04.2020

Терентьев Александр Александрович
Доктор технических наук, профессор, профессор кафедры информационных технологий проектирования и прикладной математики, *orcid.org/0000-0001-6995-1419*
Киевский национальный университет строительства и архитектуры, Киев
Доля Елена Викторовна
Кандидат физико-математических наук, доцент, доцент кафедры информационных технологий проектирования и прикладной математики, *orcid.org/0000-0003-2503-2634*
Киевский национальный университет строительства и архитектуры, Киев
Лященко Тамара Алексеевна
Старший преподаватель кафедры информационных технологий, *orcid.org/0000-0001-9092-0297*
Киевский национальный университет строительства и архитектуры, Киев

Кузьминский Олег Викторович
Студент, *orcid.org/0000-0002-4528-9210*
Киевский национальный университет строительства и архитектуры, Киев

ДИАГНОСТИКА СЕТЕЙ И ПРОТИВОДЕЙСТВИЕ СЕТЕВЫМ УГРОЗАМ

Аннотация. *Информационная безопасность в облачных сервисах – один из важнейших вопросов, который привлекает много исследований и разработок за последние несколько лет. В частности, злоумышленники могут исследовать уязвимости облачной системы и компроментировать виртуальные машины для развертывания дальнейших масштабных распределенных злонамеренных действий (DDoS). DDoS-атаки обычно включают действия на ранней стадии, такие как многоступенчатая эксплуатация, сканирование уязвимости на низких частотах и компроментация идентифицированных уязвимых виртуальных машин как подчиненных, а также DDoS-атаки через компроментируемые виртуальные машины. В облачной системе, особенно облаках инфраструктуры как услуга (IaaS), выявить атаки таких подчиненных машин чрезвычайно трудно. Это происходит потому, что пользователи облака могут устанавливать уязвимые программы на своих виртуальных машинах. Чтобы избежать компроментации уязвимых виртуальных машин в облаке, предложен многофазный механизм выявления, измерения и выбора контрмеры, основанной на аналитических моделях, на налаженных контрмерах на основе виртуальной сети. Результаты оценивания системы и безопасности демонстрируют эффективность предложенного решения.*

Ключевые слова: облачные расчеты; облачные сервисы; информационная безопасность; DDoS; обнаружения вторжений

Terentyev Alexander
DSc (Eng.), Associate Professor, Department of Information Technology of Design and Applied Mathematics, *orcid.org/0000-0001-6995-1419*
National University of Civil Engineering and Architecture, Kyiv
Dolya Olena
PhD., Associate Professor, Department of Information Technology of Design and Applied Mathematics, *orcid.org/0000-0003-2503-2634*
Lyashchenko Tamara
Senior Lecturer of the Department of Information Technology, *orcid.org/0000-0001-9092-0297*
Kyiv National University of Construction and Architecture, Kyiv
Kuzminskyi Oleh
Student, *orcid.org/0000-0002-4528-9210*
Kyiv National University of Construction and Architecture, Kyiv

DIAGNOSING AND COUNTERACTING NETWORK THREATS

Abstract. *Information security in cloud services is one of the most important issues that has attracted much research and development over the last few years. In particular, attackers can investigate the vulnerabilities of the cloud system and compromise virtual machines to deploy further large-scale distributed malware (DDoS). DDoS attacks typically include early-stage actions such as multi-stage exploitation, low-frequency vulnerability scanning, and compromising of identified vulnerable virtual machines as subordinate, and finally DDoS attacks through compromised virtual machines. In a cloud system, especially infrastructure-as-a-service (IaaS) clouds, detecting attacks from such slave machines is extremely difficult. This is because cloud users can install vulnerable programs on their virtual machines. To avoid the compromise of vulnerable virtual machines in the cloud, a multiphase mechanism for detecting, measuring and selecting a countermeasure based on analytical models based on well-established countermeasures based on a virtual network is proposed. System and safety assessments demonstrate the effectiveness and efficiency of the proposed solution.*

Keywords: cloud computation; cloud services; informational security; DDoS; intrusion detection

---

Посилання на публікацію

APA *Terentyev, Alexander, Dolya, Olena, Lyashchenko, Tamara, & Kuzminskyi, Oleh, (2020). Diagnosing and counteracting network threats. Management of Development of Complex Systems, 42, 125 – 131, dx.doi.org\10.32347/2412-9933.2020.42.125-131.*

ДСТУ *Терентьєв О.О. Діагностування та протидія мережевим загрозам [Текст] / О.О. Терентьєв, О.В. Доля, Т.О. Лященко, О.В. Кузьмінський // Управління розвитком складних систем. – 2020. – № 42. – С. 125 – 131; dx.doi.org\10.32347/2412-9933.2020.42.125-131.*