♦ сегментацію — виділення (сканування) меж ліній штрих-кодів;

♦ декодування і перевірку коду.

Мета сегментації — визначити межі білих і чорних штрихів для подальшого декодування. Найчастіше для цього застосовують сканування локальних максимумів і мінімумів; метод збалансованого відсікання гістограми; методи бінаризації Ейквіла, Бернсена, Оцу, Ніблека, а також ітеративний метод Рідлера–Калварда (isodata).

Процес декодування має бути веріфікований перевіркою контрольного коду (цифри), котрий здебільшого обчислюється за допомогою деякої модифікації алгоритму Луна.

Список використаної літератури

1. **Pavlidis, T.** *Fundamentals of bar code information theory / T. Pavlidis, J. Swartz, and Y. P. Wang // Computer.— Apr. 1990.— Vol. 23, no. 4.— P. 74–86.*

2. **Anjos, A.** *Bi-Level Image Thresholding — A Fast Method / A. Anjos, H. Shahbazkia // BioSignals.— 2008.— Vol. 2.— P. 70–76.*

3. **Bernsen, J.** *Dynamic thresholding of grey-level images / J. Bernsen // Proc. 8th Int. Conf. on Pattern Recognition, Paris, 1986.— P. 1251–1255.*

4. **Otsu, N.** *A Threshold Selection Method from Gray-Level Histograms / N. Otsu // Automatica.— 1975.— Vol. 11.— P. 285–296.*

5. **Niblack.** *An Introduction to Digital Image Processing. Englewood Cliffs, NJ: Prentice-Hall, 1986.— P. 115–116.*

6. **Zhang, Z.** *Restoration of images scanned from thick bound documents / Z. Zhang and C. L. Tan // Proc. Int. conf. Image Processing.— 2001.— Vol. 1.— P. 1074–1077.*

7. **Ridler, T. W.** *Picture Thresholding Using an Iterative Selection Method / T. W. Ridler and S. Calvard // IEEE Trans. on Systems. Man and Cybernetics.— 1978.— Vol. SMC-8, no. 8.— P. 630–632.*

8. *https://www.wikipedia.org/*

9. **Leedham** *et al. Comparison of some thresholding algorithms for text/background segmentation in difficult document images // IEEE.— 2003.*

Рецензент: доктор техн. наук. професор **А. І. Семенко**, Державний університет телекомунікацій, Київ.

А. А. Мокринцев, В. В. Жебка

СОВРЕМЕННАЯ МЕТОДИКА И АЛГОРИТМЫ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ ОДНОМЕРНЫХ ШТРИХ-КОДОВ

Освещены новейшие тенденции и подходы к разработке методики и алгоритмов автоматического распознавания одномерных штрих-кодов. Исследована общая проблематика распознавания штрих-кодов. Рассмотрены вопросы их сегментации и декодирования с проверкой корректности его выполнения.

Ключевые слова: одномерный штрих-код; цифровая обработка изображений; сегментация изображений; декодирование штрих-кодов.

O. A. Mokrintsev, V. V. Zhebka

MODERN METHODS AND ALGORITHMS OF COMPUTER LINEAR BARCODE RECOGNITION

The work covers latest tendencies and approaches in linear barcodes reading methods development. Studied problems barcode recognition. Considered segmentation or isolation boundaries lines barcodes. Investigated decoding and verification code.

Keywords: one-dimensional barcode; digital image processing; image segmentation; barcode decoding.

УДК 621.391

А. О. МАКАРЕНКО, канд. техн. наук, доцент;
Г. О. ГРИНКЕВИЧ, канд. техн. наук, доцент;
Н. В. КОРШУН, канд. техн. наук, доцент;
В. М. КУКЛОВ, аспірант;
А. І. ПІДРУЧНИЙ, аспірант;
Г. В. ТКАЧ, студентка,
Державний університет телекомунікацій, Київ

Аналіз програмного забезпечення, необхідного для контролю пакетів даних у програмно-конфігурованих мережах

Розглянуто питання щодо вимірювання та моніторингу Software-Defined-Networking, покладені в основу алгоритмів виявлення несправностей, і подано огляд сучасних програмних засобів, які мають на меті забезпечити безпеку й контроль трафіку в програмно-конфігурованих мережах.

Ключові слова: інформаційні мережі; програмно-конфігуровані мережі; інформаційна безпека; SDN; Linux; Ethernet; OpenFlow.

ВСТУП

Постановка задачі. Техніка захисту може відновити OpenFlow мережі в межах 50 мс. Для цього жодних дій від мережного контролера не знадобиться, бо перемикач має змогу безпосередньо реагувати на несправності. У техніці реставрації контролер має взаємодіяти з мережними пристроями, що потребує більше часу і робить метод менш зручним для великомасштабних мереж із багатьма підвузлами.

Аналіз останніх досліджень і публікацій. Сучасний стан формування методів аналізу і синтезу програмно-конфігурованих мереж нерозривно пов'язаний із працями таких учених, як O. Sheyner, P. Ammann, X. Ou, L. Wang, A. Poй, H. Poolsappasit.

Мета статті — подати огляд сучасних програмних засобів, здатних забезпечити безпеку та контроль трафіку в програмно-конфігурованих мережах.

ОСНОВНА ЧАСТИНА

Загальні положення

Для того щоб виявити неробоче посилання своєчасно, протоколи управління можуть бути використані з метою моніторингу підмкнення. Наприклад, неактивний порт комутатора може бути виявлений за втратою сигналу відмови. Відшукати ламаний шлях між двома вузлами можна за допомогою двонапрямленого Forwarding Detection (BFD) на базі простого протоколу Hello, визначеного в RFC 5880. Як альтернативу можна використовувати Link Layer Discovery Protocol (LLDP), але це викликає високе навантаження на мережному контролері і обмежує масштабованість, оскільки такі повідомлення моніторингу мають бути оброблені на високій частоті.

У літературі [1] пропонується розширення OpenFlow специфікації для розгортання децентралізованого моніторингу мережі, зокрема генерування пакетів і обробки на мережних пристроях. При цьому демонструється підхід до несправності на основі технології MPLS і концентрується увага на відновленнях при збоях у межах 50 мс.

Для того щоб мінімізувати кількість відповідних повідомлень, автори [1] пропонують тільки інформування вимикачів про збої зв'язку. Необхідний алгоритм діє на мережних пристроях, що дозволяє забезпечити менший час відновлення, ніж той, що вимагається на основі схеми повідомлених катіонів. Як і в попередній концепції, знадобляться додаткові функції, які будуть додані до комутатора OpenFlow.

Вимірювання та моніторинг

Моніторинг мережі інформує оператора про поточний стан мережі, а також є основою для алгоритмів виявлення несправностей. Це вимагає наявності відповідних датчиків у мережі, наприклад комутаторів, які забезпечують статистичні дані про поток записів.

NetFuse призначено для виявлення та пом'якшення перевантаження, яке може виникнути з різних причин, скажімо через розподілену відмову в обслуговуванні (DDoS) атаки або в результаті запланованого резервного копіювання. Для того щоб виявити таку раптову несправність, NetFuse встановлюється як додатковий шар між мережними пристроями і мережним контролером для обробки OpenFlow керуючих повідомлень (Packet-in, FlowMod, FlowRemoved тощо). Для того аби діагностувати перевантаження, багатовимірний алгоритм агрегації використовується з метою виявлення підозрілих потоків. Надалі вони скеровуються адаптивним механізмом управління, який модифікує правила контролю на комутаторах, щоб краще справлятися з мережним перевантаженням.

На основі OpenFlow керуючих повідомлень (Packet-in і FlowRemoved, FlowSense) для обчислення лінії зв'язку використовують програми, виконувані у верхній частині мережі контролера. Спрощується розгортання порівняно з підходами, які вимагають додаткового шару (наприклад, NetFuse). Оскільки вимірювання поновлюється тільки після отримання повідомлення FlowRemoved, то потік, який задовольняє правило байдужих полів у поєднанні з тривалим терміном дії, призводить до сповільненого результату. За активної політики встановлення потоку повідомлення Packet може надійти будь-коли, а це означає, що активне опитування знадобиться для виявлення поточного стану.

Робота OpenSketch розглядається програмно і включає в себе просту конструкцію для площини даних, які можуть бути реалізовані на апаратному забезпеченні, тоді як функції аналізу даних розташовано в площині управління. Згідно з ідеєю ескізів, які є компактними структурами даних, найбільш гнучкі зіштовхування потоку можна порівняти з OpenFlow специфікацією. У поєднанні з контролером OpenSketch для ескізу бібліотеки можуть бути розроблені нові алгоритми вимірювання, що уможливлюють автоматизовану конфігурацію комутаторів відповідно до вимог додатка вимірювань.

Установлення додаткових потоків записів для цілей моніторингу розглядається у складі деякої схеми, можливої тому, що OpenFlow специфікації підтримують кілька етапів потоку записів. Мережний контролер може зчитувати відповідні лічильники періодично і виявити зрештою важливий Hitters через ієрархічний алгоритм. Переваги цього підходу полягають у використанні апаратного забезпечення і невисоких витратах на комутатор.

Виявлення атак DDoS спирається на аналіз потоку статистики за допомогою нейронних мереж і підходів, реалізованих на платформі контролера NOX. При цьому виконуються такі кроки:

```
-> [FlowCollector] -> [FeatureExtractor] -> [Класифікатор] ->Alarm
| ---------------------------------- <-------------|
LoopDetection
```

Насамперед потік статистики з одного чи кількох комутаторів витягується через певні проміжки часу за допомогою потоку колектора. Далі ідентифікуються відповідні функції С, які вказують на атаку DDoS. Наприклад, темпи зростання окремих притоків в одному напрямі є індикатором початку такої атаки. Кількість перемикачів, що беруть участь у моніторингу, може бути адаптована до нових топологій, але збільшення кількості комутаторів створює додаткові накладні витрати внаслідок збільшення кількості керуючих повідомлень.

Пряме вимірювання лічильників, пов'язаних із входом, необхідне для того, аби побудувати матриці, які характеризують обсяг потоків між пунктами відправлення та призначення пар у мережі. Але запити збільшують навантаження на комутатор, що небажа-

но в мережі, яка складається з кількох перемикачів і великої кількості притоків. OpenTM описує стратегії визначення того, якому комутатору для запиту вздовж шляху потоку слід зменшити накладні витрати. Автори пропонують рівномірний розподіл для запитів потоку лічильників між усіма комутаторами в мережі, включаючи інформацію про маршрутизацію від мережного контролера, здійснювану з цією метою.

Можна описати зазначені запити як P2P трафік, котрий може класифікуватися на основі особливостей мережного рівня. Це означає, що питання конфіденційності пов'язані з глибокою інспекцією невідомих пакетів (DPI) і відпадає потреба у використанні спеціалізованого апаратного забезпечення, оскільки аналіз можливий на мережному контролері. Після визначення відповідних функцій мережного рівня, здатних відрізнити P2P трафік від не-P2P трафіку, вони можуть бути реалізовані як додаток для контролера NOX і оцінені в плані загальнодоступного набору даних.

Налаштування, перевірка й тестування

Виявлення мережних несправностей є серйозною проблемою для кожного оператора, зумовленою людським фактором. Для того щоб зменшити час виявлення несправності, необхідно застосовувати нові методи для SDN архітектур. З метою налагодження мереж і використовується підхід, котрий включає в себе застосування як мережного контролера (програмне забезпечення), так і мережних пристроїв (апаратне забезпечення), що потребують відшукання основної причини несправності. Формальна перевірка програмного забезпечення контролера може визначити, чи гарантуються певні властивості коректності (наприклад, припинення циклів) і чи забезпечується SOFT [2].

Тестування комп'ютерних мереж потрібне для того, аби гарантувати, що всі мережні компоненти працюють як і очікувалося. Має бути розроблено тестовий сценарій, якомога ближчий до операціональних ситуацій. Ця проблема розв'язується за допомогою ATPG [3], який генерує тестові пакети, тоді як OFRewind [4] уможливлює повторне відтворення захоплених сценаріїв.

OpenFlow дозволяє використовувати стандартне обладнання, яке потребує правильного виконання агентів на всіх мережних пристроях. ONF має визначити випробування із серією тестів, що передусім корисно для виробників задля уникнення помилок у програмному забезпеченні на етапі розробки нової моделі комутатора. Аналогічний підхід реалізує OFTest [3], який є частиною проекту Project Floodlight і дозволяє розробляти платформи SPECI тестових сценаріїв.

SOFT [5] використовує тести на сумісність, щоб забезпечити правильну реалізацію на всіх комутаторах у мережі. З огляду на символічне виконання всі можливі шляхи в програмі (firmware) відтворюють і визначають поведінку кожного мережного пристрою. Далі здійснюється перехресна перевірка між мережними пристроями з використанням головного пристрою, що виявляє невідповідності між вхідними наборами. Це дозволяє мережним операторам знаходити помилки реалізації перед розгортанням і гарантує безпомилкову роботу мережних пристроїв.

Рівні стану можуть бути коректно відображені на будь-який інший рівень (еквівалентності). У протилежному випадку помилку можна локалізувати між рівнями, які відрізняються один від одного і можуть бути виявлені в проміжному рівні коду.

Системний підхід до пошуку та усунення несправностей SDN висвітлено в [2], де пропонується потік, який відокремлює стек SDN на рівні стану і на кодових рівнях, що ілюструє рис. 1. Рівні стану являють собою мережу конфігурації, тоді як рівні стека описують відображення між двома рівнями стану. Методика дозволяє автоматичне визначення кодового рівня, де міститься несправність. Нормальна поведінка мережі приводить до еквівалентності між усіма рівнями стану таким чином, що кожний рівень може бути відображений на будь-який інший. У разі несправності мережі помилку можна локалізувати в код рівня, розташованого між рівнями стану, які не є еквівалентними. Після цього причину помилки може бути визначено за допомогою методів, описаних раніше.

Мережний налагоджувач ndb [5] допомагає оператору визначити причину збоїв програмного забезпечення (або помилки) у мережі. Кожний комутатор надсилає повідомлення, які спрацьовують, коли пакет з інформацією про збіг потоку вводу надходить на комутатор. Такі повідомлення збираються з усіх перемикачів у централізований колектор, який може бути використаний для визначення пакетів, що стосуються пакета точки зупинки. Наприклад, для того аби досліджувати помилку досяжності для пакета, надісланого від хоста *A* до хоста *B*, може бути використаний такий запит:

OFRewind [3] забезпечує запис і відтворення налагоджених засобів для SDN. Він вставляється як проксі-сервер між мережним контролером і мережними пристроями, а також дозволяє перехоплення і видозміну повідомлень управління з метою запису або відтворення фабричного калібрування. Для зберігання трафіку користувачів сховища даних, керовані компонентом OFRewind, підмикаються до комутаторів, як це зображено на рис. 2. Існують режими, наприклад Replay, які дозволяють управляти різні тестові сценарії, аби отримати тільки керуючі повідомлення.

VeriFlow [3] забезпечує достатньо сфокусований вигляд на площині даних VERI. Це додає додатковий шар між мережним контролером і мережними пристроями, щоб можна було перевірити нові правила в режимі реального часу, перш ніж вони

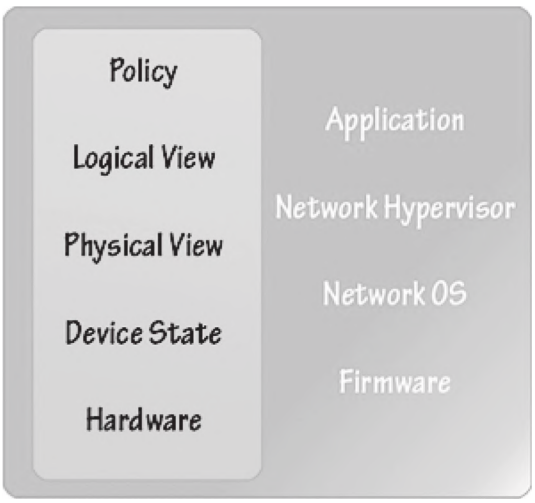

Рис. 1. Рівні стану та кодові рівні стека SDN у разі безпомилкової роботи в мережі

будуть розгорнуті в мережі. Правила мережі поділено на множину класів еквівалентності (EKM), де аналогічні дії з переадресації зосереджуються в певному EKM. Вони зберігаються в деревоподібних структурах даних, котрі описують пересилання пакетів у мережі. Нові правила аналізують VeriFlow шляхом виконання запитів на основі інваріантів для того, щоб виявити їх порушення. Такі інваріанти можуть бути встановлені через API і охоплювати широкий спектр умов, наприклад досяжності, петлі вільності та консистенції.

Anteater [6] також фокусується на площині даних аналізу, що являє собою збірну топологію мережі та пересилання інформаційних баз (FIBs) із мережних пристроїв як виконання логічних функцій. Використовуючи цей метод, автори змогли виявити 23 помилки в мережі. Перевага такого механізму полягає в тому, що реальна поведінка системи може бути проаналізована без залучення протоколів маршрутизації, завдяки чому вся процедура значно спрощується. Окрім того, цей підхід дозволяє виявляти помилки програмного забезпечення маршрутизатора.

Просторовий аналіз (HSA) [4] зосереджується на виявленні негараздів, таких як збої досяжності, цикли пересилання, трафік з ізоляцією, проблеми витоку. Кожний заголовок пакета подається як точка в просторі, в якому моделюються мережні пристрої та порти. Коли пакет перетинає мережу, вона трансформується з однієї ділянки до іншої. Мережні пристрої, такі як комутатори і маршрутизатори, додатково можуть бути змодельовані за допомогою відповідних функцій передавання даних. У поєднанні із заголовком простору це дозволяє виявляти причини відповідних відмов. І хоча така структура здатна локалізувати джерело помилки (наприклад, непослідовні таблиці маршрутизації), вона не може визначити причину, з якої сталася помилка.

Випробування нових застосувань контролера потрібне OpenFlow для належної конфігурації контролера і перемикачів. Оскільки лише невелика кількість користувачів має доступ до таких ресурсів, системи зі складовими у вигляді віртуальних машин (VM) набувають популярності. Через величезну витрату пам'яті, необхідної для створення великої мережі, масштабованість поширюється лише на кілька перемикачів. Легший тип віртуалізації можливий із Mininet [5] мережним емулятором, що дозволяє використовувати мережу як альтернативу для тестування на стандартному ноутбуці (рис. 2). У разі створення нової мережі наведена далі команда дає приклад тесту на віртуальній мережі з топологією дерева глибини 2 та розгалуження 8, що використовує мережний контролер на основі NOX:

```
mn --switchovsk --controllernox --topotree, depth=2,\\ fanout=8 --testpingAll
```

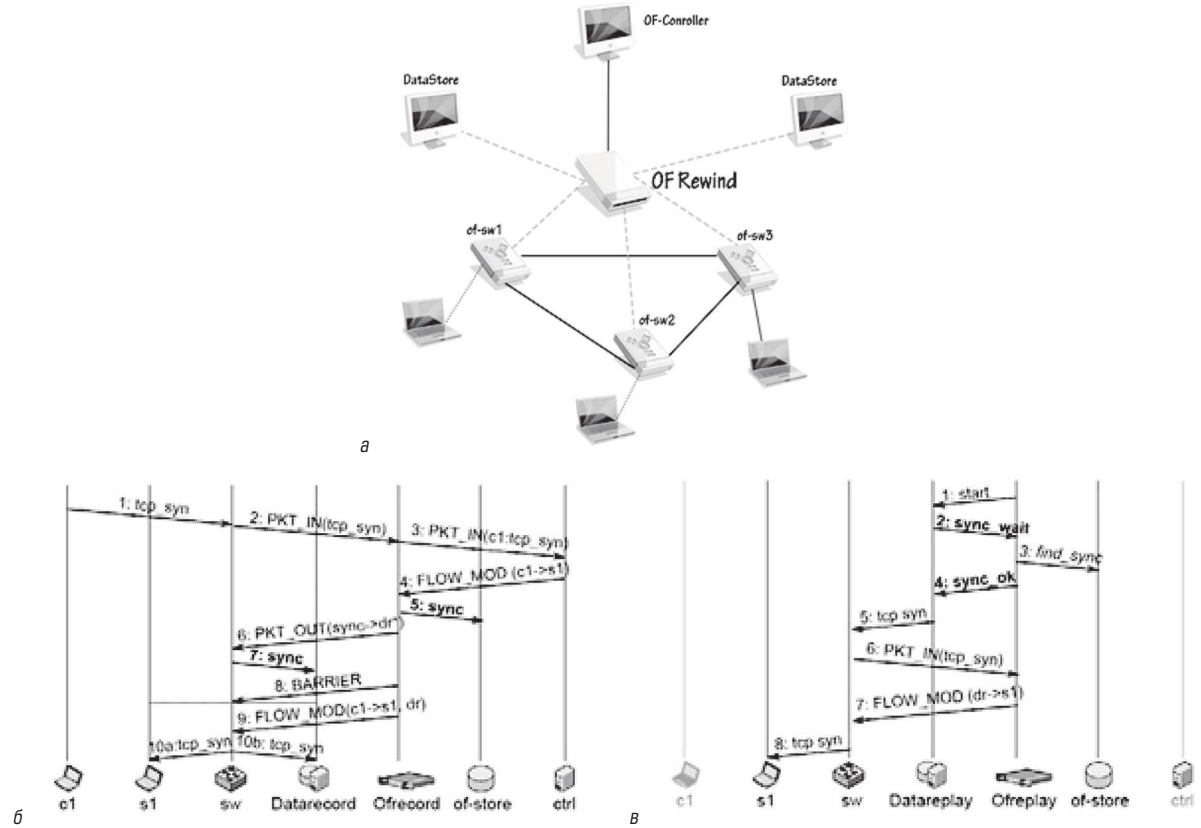

Рис. 2. Системні компоненти OFRewind: *а* — взаємозв'язок компонентів системи; *б* — Ofrecord; *в* — Ofreplay (запит TCP SYN відправляється від хоста c1 до s1 і може бути записаний за допомогою Ofrecord (б) механізму Ofreplay (в); при цьому запит надходить із системи зберігання Datareplay від c1)

Усі номери, хости, комутатори й контролери емулюються, причому команди оболонки дозволяють використовувати основні засоби мережі (наприклад, пінг). Mininet дає змогу розробникам виконувати й тестувати нові програми контролера в різних мережних топологіях. Обмеження цього підходу виникає через те, що продуктивність віртуалізації машини зазнає впливу високих навантажень, швидкості передавання пакетів, складності O(n) порівняно з O(1) для таблиці пошуку в апаратному перемикачі на основі асоціативної пам'яті (TCAM). Завдяки своїй зручності Mininet є найбільш поширеним емулятором для SDN і OpenFlow.

Мережа відлагодження необхідна й для того, щоб виявляти помилки, коли збої програмного забезпечення не є очевидними. Відомий метод [6] може бути використаний для тестування програм контролера з метою виявлення порушень коректності, викликаних помилками в програмному забезпеченні контролера. Перевірка моделі використовується для опису топології мережі та дослідження простору станів, який включає в себе контролер, перемикачі та хости. Для того щоб зменшити розмір вхідного простору, обробку подій на контролері виконують автоматично символічним двигуном, який генерує тестові входи і впускає пакети в мережу. Метод-прототип був оцінений на трьох реальних OpenFlow додатках, написаних у Python для контролера NOX. Типовим прикладом є помилка в застосуванні комутатора MAC-навчання, який має надсилати пакети конкретного пристрою, навіть якщо цей пристрій змінює розташування в мережі. Якщо жорсткий тайм-аут нотифікації пропущений, то всі пакети передусім доставляються на колишнє місце, а відповідні записи потоку не оновлюються з новими параметрами визначення місцезнаходження. Інструмент дозволяє виявити такі конструкції помилок реалізації, що є загальнодоступними.

Безпека

Архітектура SDN дозволяє реалізувати нові концепції безпеки, які неможливо було впровадити раніше. Наприклад, кожний мережний пристрій може бути налаштований так, щоб блокувати пакети аналогічним чином до мережі Брандмауер, хоча для виявлення вторгнень традиційно потрібні дорогі апаратні вирішення. У [7] показано, як алгоритми виявлення аномалій можуть бути адаптовані до мереж OpenFlow на основі реалізації в мережному контролері. Проте архітектура SDN забезпечує новий простір для діяльності зловмисників і вимагає адекватних механізмів захисту. Наприклад, FortNOX [8] забезпечує механізм безпеки, який захищає потік від супротивників.

У [9] розглядається виявлення аномалій OpenFlow і стверджується, що потік повинен бути переміщений від ядра до домашньої мережі (близько до користувача), щоб отримати якнайкращі результати виявлення. Автори адаптували чотири відомі алгоритми для використання з OpenFlow, у тому числі для виявлення сканувальних інфекцій на хостах, що лімітують швидкість за умов інфікування, та аномалій з використанням максимальної ентропії і детектора аномального значення. Оскільки навантаження з обробки поширюється на домашніх користувачів, такий підхід послаблює вимоги до обробки, що стосуються постачальника послуг інтернету (ISP) і знижує витрати. При цьому розгортання алгоритмів виявлення на мережному контролері не впливає на продуктивність пересилання пакетів на площині даних.

FRESCO [10] структура дозволяє розгортати служби безпеки для розімкненого Flow. Вона включає в себе мову сценаріїв, що сприяє розвитку служб безпеки на базі API та бібліотек, що складаються з 16 багаторазових модулів. Саму структуру FRESCO реалізовано як додаток, побудований на контролері NOX.

Захист від атак IP-сканування описується в OpenFlow у випадковому вузлі комутації [2]. Ідеться про реалізацію ідеї, що статичні IP-адреси є легкою мішенню для зловмисників, але таких загроз можна уникнути за допомогою проактивних методів, які часом змінюють IP-адреси господарів — рухома мішень оборони (MTD). У разі розімкненого потоку реальна IP-адреса зберігається господарем, але замінюється віртуальною IP-адресою, яка часто зазнає перепризначення до мережних пристроїв за допомогою мережного контролера. Для цього потрібен механізм трансляції адрес, а також гарантії обмежень, таких як мутації непередбачуваності. Підхід оцінювали з використанням Mininet проти зовнішнього мережного сканера (NMAP15) та атаки хробака.

FortNOX [4] — нове виконання ядра політики безпеки, що підвищує захист потоку під час процедури налаштування в OpenFlow. Це може бути використано суперником для того, щоб взяти під своє керування мережу. FortNOX вимагає цифрового підпису для авторизації. Залежно від застосування, він визначає рівень пріоритету потоку з правилом у потік таблиці. Окрім того, FortNOX виявляє, коли потік може обійти політику безпеки. Це можливо не тільки в разі перекриття діапазонів IP, а й тоді, коли правило встановлює новий заголовок пакета і пакет може досягти пункту призначення. В іншому випадку відбувається блокування.

ВИСНОВКИ

♦ Розглянуто питання вимірювання та моніторингу Software-Defined-Networking. Моніторинг мережі інформує оператора про її поточний стан, а також є основою для алгоритмів виявлення несправностей. Це вимагає наявності відповідних датчиків у мережі.

♦ Здійснено аналіз сучасних програмних засобів, здатних забезпечити безпеку та контроль трафіку в програмно-конфігурованих мережах.

♦ Показано, що для налагодження програмного забезпечення необхідно здійснювати контроль за пакетами в мережі, хоча формально перевірка може бути використана для визначення правильності потоку. Це необхідно для того, аби генерувати докладні мережні знімки, а також інспектувати параметри функціонування за певних умов у мережі.

♦ Потенціал OpenFlow, реалізований на нових схемах безпеки, ще потребує вивчення.

♦ Аналіз пакетів у повідомленні Packet на контролері мережі показує, що вони можуть бути використані для ідентифікації та визначення проникнення зв'язку, перш ніж потік буде налаштовано.

♦ Для забезпечення безпеки мережі було досліджено нові вектори атак, але централізований контролер мережі та канал управління залишаються частково незахищеними.

Список використаної літератури

1. ***Metaswitch*** *Networks. PCE — an evolutionary approach to SDN [Електронний ресурс].— Режим доступу: http://www.metaswitch.com/sites/default/files/metaswitch-white-paper-pce-an-evolutionary-approach-to-sdn.pdf (2012.)*

2. ***Смелянский, Р. Л.*** *Программно-конфигурируемые сети / Р. Л. Смелянский // Открытые системы. СУБД 9.— 2012.— С. 23–26.*

3. ***Захаров, А. А.*** *Аспекты информационной безопасности архитектуры SDN [Електронний ресурс] / А. А. Захаров, Е. Ф. Попов, М. М. Фучко.— Режим доступу:*
http://vestnik.sibsutis.ru/uploads/1459328716_7845.pdf
4. ***Розробка*** *методу балансування навантаження в SDN мережах на основі модифікованого протоколу STP [Електронний ресурс] / [М. М. Климаш, М. І. Бешлей, Ю. Л. Дещинський, О. М. Панченко] // Комп'ютерні технології друкарства.— 2015.— С. 146–155.— Режим доступу:*
http://ctp.uad.lviv.ua/images//ktd/34_klymach.pdf
5. ***Шалимов, А. В.*** *Технологии SDN/OpenFlow [Електронний ресурс] / А. В. Шалимов.— Режим доступу:*
http://lvk.cs.msu.su/~sveta/SDN_OpenFlow_basics_lecture1.pdf
6. ***Fabric:*** *a retrospective on evolving SDN / [M. Casado, T. Koponen, S. Shenker, and A. Tootoonchian] // Proceedings of the First Workshop on Hot Topics in Software Defined Networks (HotSDN'12).— New York, NY, USA: ACM.— 2012.— P. 85–90.*
7. ***Hoelzle, U.*** *OpenFlow @ Google [Електронний ресурс] / U. Hoelzle.— Режим доступу:*
http://opennetsummit.org/archives/apr12/hoelzle-tue-openflow.pdf (2012.)
8. ***HP Networking.*** *Software defined networks (SDN) [Електронний ресурс].— Режим доступу:*
http://h17007.www1.hp.com/us/en/mobile/solutions/tech/sdn.html
9. ***Juniper*** *Networks OpenFlow Switch Application (OF-APP) for Juniper MXSeries Routers [Електрон-ний ресурс].— Режим доступу:*
https://developer.juniper.net/shared/jdn/ docs/ProgrammableNetworks/OpenFLow APP JDN Overview.pdf
10. ***NOX:*** *towards an operating system for networks [Електронний ресурс] / [N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown and S. Shenker].— Режим доступу:*
http://www.cs.yale.edu/homes/jf/nox.pdf

Рецензент: доктор техн. наук **В. Ф. Заїка**, Державний університет телекомунікацій, Київ.

А. А. Макаренко, А. А. Гринкевич, Н. В. Коршун, В. М. Куклов, А. И. Пидручный, А. В. Ткач

АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, НЕОБХОДИМОГО ДЛЯ КОНТРОЛЯ ПАКЕТОВ ДАННЫХ В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ

Представлена концепция Software-Defined-Networking. Приведены сведения из истории программируемых сетей и дан обзор связанных с ними новых стандартов, один из которых — протокол OpenFlow, сыграл важную роль в продвижении концепции SDN и значительно повлиял на научные исследования в теоретическом и практическом аспектах.

Ключевые слова: информационные сети; программно-конфигурируемые сети; информационная безопасность; SDN; Linux; Ethernet; OpenFlow.

A. O. Makarenko, G. A. Grynkevych, N. V. Korshun, V. M. Kuklov, A. I. Pidruchnyi, G. V. Tkach

ANALYSIS SOFTWARE NEEDED TO ENSURE CONTROL OVER DATA PACKETS IN THE SOFTWARE-CONFIGURED NETWORKS

This paper presents the problems in reference to Software-Defined-Networking monitoring.

Keywords: information network; software-configured network; information security; SDN; Linux; Ethernet; OpenFlow.