Моделирование и анализ информационных систем. Т. 26, №1 (2019), с. 90–100
Modeling and Analysis of Information Systems. Vol. 26, No 1 (2019), pp. 90–100

DOI: 10.18255/1818-1015-2019-1-90-100

УДК 004.7

Разработка и исследование алгоритмов формирования правил для узлов сетевой безопасности в мультиоблачной платформе

Парфёнов Д. И., Болодурина И. П., Торчин В. А.

Поступила в редакцию 10 января 2019
После доработки 15 февраля 2019
Принята к публикации 17 февраля 2019

Аннотация. В рамках исследования рассмотрены существующие решения, направленные на обеспечение безопасности сетевого периметра мультиоблачной платформы. Установлено, что наиболее острой является проблема эффективного формирования правил на межсетевых экранах. Существующие подходы не позволяют оптимизировать список правил на узлах, контролирующих доступ к сети. Целью исследования является повышение эффективности средств межсетевого экрана путем бесконфликтной оптимизации правил безопасности и применения нейросетевого подхода в программно-определяемых сетях. Предлагаемое решение основано на совместном использовании интеллектуальных математических подходов и современных технологий виртуализации сетевых функций. В ходе экспериментальных исследований проведен сравнительный анализ традиционных средств формирования правил, нейросетевого подхода, а также генетического алгоритма. Для автоматического построения правил сетевой безопасности рекомендуется применять нейросетевой классификатор архитектуры «многослойный персептрон», поскольку он даёт лучшие результаты с точки зрения производительности, и уменьшать размерность списка правил безопасности межсетевого экрана при помощи сети Кохонена, поскольку это средство показывает лучшую производительность. В спроектированную архитектуру был внедрен алгоритм бесконфликтной оптимизации, который производит конечную оптимизацию путем ранжирования и выведения наиболее часто встречаемых исключений из больших запретительных правил, что позволяет увеличить защиту от атак, которые направлены на выявление правил безопасности, находящихся внизу списка межсетевого экрана. На базе предложенного решения в рамках исследования реализован модуль адаптивного межсетевого экрана.

Ключевые слова: адаптивный межсетевой экран, программно-конфигурируемая сеть, мультиоблачные платформы, нейронная сеть, виртуализация сетевых функций, кибербезопасность

Для цитирования: Парфёнов Д. И., Болодурина И. П., Торчин В. А., "Разработка и исследование алгоритмов формирования правил для узлов сетевой безопасности в мультиоблачной платформе", *Моделирование и анализ информационных систем*, **26**:1 (2019), 90–100.

Об авторах: Парфёнов Денис Игоревич, канд. техн. наук, orcid.org/0000-0002-1146-1270
Оренбургский государственный университет,
пр. Победы, 13, г. Оренбург, 460018 Россия, e-mail: parfenovdi@mail.ru

Болодурина Ирина Павловна, д-р техн. наук, профессор, orcid.org/0000-0003-0096-2587
Оренбургский государственный университет, e-mail: prmat@mail.osu.ru

Торчин Вадим Александрович, студент, orcid.org/0000-0002-5315-6047
Оренбургский государственный университет, e-mail: vadim.torchin@gmail.com

Благодарности:
Работа выполнена при финансовой поддержке РФФИ, научные проекты № 18-07-01446 и № 16-29-09639.

Введение

В настоящее время активно развивается рынок телекоммуникационных услуг. Крупные организации для более эффективного ведения бизнеса арендуют виртуальные центры обработки данных (ЦОД) для размещения собственной ИТ-инфраструктуры. Поэтому наиболее востребованным сегментом таких услуг является предоставление пользователям сетевых сервисов на базе мультиоблачных платформ. Популярность использования таких технических решений привела к тому, что пользователи и поставщики телекоммуникационных услуг ежедневно сталкиваются с проблемами, связанными с угрозами в сфере кибербезопасности.

Согласно аналитическим данным ведущих поставщиков сетевого оборудования, таких как Cisco и Huawei, количество активных угроз кибербезопасности ежегодно увеличивается на 15–25%. Оценивая вектор атак на ИТ-инфраструктуру, которая поддерживает работу информационных систем в крупных компаниях, можно составить следующий рейтинг угроз: ограничение доступа легитимных пользователей к ключевым ресурсам компании (25%); нарушение работы технологического оборудования (35%); получение несанкционированного доступа к служебной или конфиденциальной информации, а также ее намеренное или случайное раскрытие, искажение либо уничтожение вследствие нарушения политики безопасности предприятия (45%). На практике список кибератак, направленных на корпоративную сеть, можно разделить на четыре основные группы. Он включает вектор атаки типа отказ в обслуживании (DDoS), вектор атаки от удаленных подключений на локальные (пользовательские) узлы (R2L), вектор атаки от локального пользователя к корневому узлу сети (U2R), а также вектор атаки типа грубого перебора узлов (Probing) [9].

Для предотвращения активных угроз провайдерам необходимы эффективные средства, которые позволяют осуществлять контроль процессов, протекающих в сети, мониторинг сервисов, размещенных в ней, а также проактивное управление элементами безопасности. На сегодняшний день наиболее востребованным и эффективным подходом к организации сети для предоставления услуг на базе виртуальных центров обработки данных является использование технологии программно-конфигурируемых сетей (Software-Defined Network, SDN). Применение данной технологии обусловлено рядом преимуществ. В первую очередь SDN значительно упрощает проектирование и эксплуатацию сети, поскольку она позволяет осуществлять централизованное интеллектуальное управление на уровне контроллера. Во-вторых, SDN позволяет сетевым администраторам быстро конфигурировать и оптимизировать сетевые ресурсы на основе агрегированного набора данных, собираемых в едином центре. В-третьих, использование SDN позволяет обеспечивать защиту с помощью динамического анализа потоков данных, циркулирующих в виртуальном центре обработки данных [10].

Еще одной технологией, применяемой для организации сети на базе виртуальных центров обработки данных, является технология виртуализация сетевых функций (Network Function Virtualization, NFV). Технология NFV предлагает новый способ проектирования, развертывания сетевых сервисов на базе мультиоблачной платформы. Виртуализация сетевых функций позволяет не только отделить сетевые функции, такие как NAT, firewall, IDS, IPS, DNS, от аппаратного уровня, но и объ-

единить все сетевые компоненты, необходимые для поддержки виртуализованной инфраструктуры на программном уровне [11]. Как и SDN, технология NFV также дает преимущества при проектировании защищенной сетевой среды в виртуальном центре обработки данных. Одним из основных преимуществ технологии NFV является масштабируемость. Для быстрого удовлетворения динамически меняющихся потребностей пользователей и предоставления новых услуг, провайдеры телекоммуникационных услуг должны иметь возможность адаптировать свою сетевую архитектуру, не меняя при этом состав аппаратного обеспечения.

Технологии SDN и NFV относятся к технологиям компьютерных сетей нового поколения и могут сосуществовать в одной сетевой среде, используя при этом общую аппаратную платформу на физическом уровне. Поэтому в рамках настоящего исследования нами предложено решение, основанное на гибридном использовании SDN и NFV для организации сетевой безопасности для мультиоблачной платформы, развернутой на базе виртуального центра обработки данных. Целью исследования является повышение эффективности средств межсетевого экрана путем бесконфликтной оптимизации правил безопасности и применения подхода нейронной сети в программно-определяемых сетях.

1. Обзор исследований

На сегодняшний день разработано достаточно много решений, позволяющих осуществлять защиту облачных платформ от кибератак. В рамках настоящего исследования нами проведен обзор таких решений. Большинство из них основано на использовании механизмов защиты на базе межсетевых экранов. Это обусловлено тем, что большинство атак поступают в сеть извне.

Механизм защиты на базе межсетевых экранов основан на использовании листов правил, разрешающих или запрещающих доступ к определенным ресурсам. Основной проблемой безопасности традиционной архитектуры межсетевых экранов является конфликт правил, возникающий в результате неправильной конфигурации. Кроме того, в больших сетях существует проблема длинных листов правил. Это особенно актуально для сетей, обеспечивающих работу платформы облачных вычислений. В таких сетях, как правило, присутствует достаточно большое множество наложенных и пересекающихся потоков, и разграничение правил в таких сетях является не тривиальной задачей. В рамках своего исследования Thawatchai Chomsiri предложил механизм Tree-Rule, который не сталкивается с такими конфликтами правил в пределах своего набора правил и работает быстрее, чем традиционные брандмауэры [1]. Тем не менее, авторы исследования отмечают, что проблемы правильности конфигурирования остаются не решенными в данной работе.

Еще одной не менее важной проблемой обеспечения безопасности на основе Firewall является единая точка отказа. Firewall, как правило, располагается на границе сети и осуществляет фильтрацию входящего трафика. Однако при высокой интенсивности потока трафика, например при DOS или DDOS атаке, Firewall может не справиться с нагрузкой. Авторами исследования [2] предложено решение на базе кластерного подхода, позволяющее осуществлять балансировку нагрузки между узлами при возрастании нагрузки на средства обеспечения безопасности. Однако

авторы не затрагивают в своем исследовании вопросов репликации правил между узлами кластера. Кроме того, данный подход не решает проблему длинных ACL правил.

Еще одним узким местом работы механизмов защиты на базе межсетевых экранов является процесс фильтрации пакетов. При высокой интенсивности запросов очень важным параметром является время принятия решения. В работе [3] предложен подход, позволяющий ускорить процедуру анализа пакетов. В основе решения применен гибридный подход, основанный на совместном использовании двух алгоритмов эффективного геометрического сопоставления и оптимизации смены состояний межсетевого экрана (GEM-iptables & nftables algorithm). Использование GEM-iptables & nftables позволяет ускорить фильтрацию пакетов, при этом не изменяя существующую схему работы Firewall. Однако такой подход эффективен только для традиционных сетей.

На сегодняшний день большинство провайдеров облачных услуг используют для обеспечения работы сети решения, основанные на программно-конфигурируемых сетях. Использование SDN позволяет решить сразу несколько проблем безопасности облачной платформы. Например, группой исследователей под руководством S. Kaur предложен распределенный межсетевой экран (Distributed Firewall), использующий в своей основе технологию SDN. В предложенном решении каждый коммутатор OpenFlow в сети может выступать в качестве брандмауэра [4]. В другом исследовании авторы используют функциональные возможности протокола OpenFlow для записи информации о текущих потоках в сетях для автоматизированного формирования правил на элементах безопасности [5]. Несмотря на все преимущества данной системы, она не в состоянии проводить оптимизацию списка правил, что сказывается на ее работе в больших сетях.

При большом количестве правил на границе сети межсетевой экран вынужден последовательно проверять весь список. Это вносит существенную задержку в работу легитимных пользователей. Для ускорения данного процесса исследователями в работе [6] предложено решение, использующее в качестве анализируемого параметра MAC-адреса. Данный метод более эффективный, чем фильтрация по IP-адресам, но он подходит только для внутренней сети и совершенно не применим для границы сети.

Другое решение в данном направлении предложено в работе [7]. Исследователи предлагают модель объединения правил фильтрации, используя алгоритм, основанный на анализе потоков трафика, классифицируя их по сервисам доступа. Однако такой подход не решает вопросов конфликтов правил для разных групп пользователей.

При возникновении инцидентов безопасности очень важно исследовать журналы доступа и применяемые списки правил межсетевого экранирования. Исследователи F. Ertam и M. Kaya предложили ряд решений для поиска закономерностей в лог-файлах с использованием классификатора векторных машин с поддержкой мультиклассов (SVM) [8]. Данное решение позволяет предотвращать новые неизвестные типы атак.

Обзор исследований показал, что для эффективного построения списка правил для межсетевого экрана необходимо решить две задачи:

– автоматизировать построение правил для межсетевого экрана на основе данных о трафике, циркулирующем в сети;

– осуществить оптимизацию полученного списка правил для межсетевого экрана.

Для решения первой задачи необходимо выполнить классификации сетевого трафика, проходящего через корпоративную сеть. Для решения второй задачи применим итерационный метод, в рамках которого выделим два основных этапа. На первом этапе выполним кластеризацию правил, полученных в результате решения первой задачи, и следующее за ней выведение правил из кластеров. На втором этапе для решения второй задачи применим алгоритмом бесконфликтной оптимизации списка правил межсетевого экранирования.

2. Модель создания адаптивных правил для межсетевого экрана

Для реализации представленного плана необходимо в первую очередь определить модель правил для межсетевого экрана, используемых для обеспечения безопасности корпоративной мультиоблачной платформы.

Правило межсетевого экрана обычно представляет собой строку, состоящую из определенных характеристик сетевого соединения и решения о допустимости такого соединения. В рамках исследования из множества характеристик, описывающих сетевые соединения, были выбраны следующие: IP-адрес отправителя и получателя, соответствующие им порты и протокол, по которому осуществляется передача данных. При выборе этих характеристик правило межсетевого экрана в общем виде будет иметь вид

$$< rn, id\_src\_ip, id\_dst\_ip, src\_p, dst\_p, p\_id, tg, p\_cnt >, \tag{1}$$

где rn – номер правила в списке; id_src_ip – идентификатор IP-адреса отправителя пакета; id_dst_ip – идентификатор IP-адреса получателя пакета; sr_p – порт отправителя пакета; dst_p – порт получателя пакета; p_id – сетевой протокол, через который осуществляется соединение; tg – решение о допустимости или недопустимости соединения; p_cnt – количество пакетов трафика.

Список записей о пакетах представим в виде множества следующего вида:

$$X = \{x_k\}, k = \overline{1, n}, \tag{2}$$

где n – длина списка правил межсетевого экранирования, применяемого для обеспечения безопасности мультиоблачной платформы.

Определим области допустимых значений для элементов данного вектора, а именно характеристики трафика, по которым будет осуществляться фильтрация передаваемых данных в сети мультиоблачной платформы. Они представлены в виде списка строк вида (1). Каждую такую строку представим в виде вектора вида

$$x_k = \{x_{k1}, x_{k2}, x_{k3}, x_{k4}, x_{k5}\}, \tag{3}$$

где k – номер записи в списке; $x_{k1} \in [0; 2^{32} - 1]$ – IP-адрес отправителя пакета; $x_{k2} \in [0; 2^{32} - 1]$ – IP-адрес получателя пакета; $x_{k3} \in [0; 65535]$ – порт отправителя пакета; $x_{k4} \in [0; 65535]$ – порт получателя пакета; $x_{k5} \in \{0, 1, 2\}$ – сетевой протокол, по которому осуществляется соединение.

Далее представим список правил межсетевого экрана, где правила соответствуют модели (2) в виде множества

$$R = \{r_i : r_i = \{r_{i,1}, r_{i,3}, r_{i,4}, r_{i,5}, r_{i,6}, r_{i,7}, r_{i,8}\}\} \tag{4}$$

где $r_{i,1} \in [0; m]$ – номер правила в списке; $r_{i,2} \in [0; 2^{32} - 1]$ – IP-адрес отправителя пакета; $r_{i,3} \in [0; 2^{32} - 1]$ – IP-адрес получателя пакета; $r_{i,4} \in [0; 65535]$ – номер порта отправителя пакета; $r_{i,5} \in [0; 65535]$ – номер порта получателя пакета; $r_{i,6} \in N$ – номер условного обозначения протокола, хранящегося в БД контроллера сети; $r_{i,7} \in \{0; 1\}$ – решение о допустимости или недопустимости соединения, где 0 – соединение запрещено, 1 – соединение разрешено; $r_{i,8} \in N$ – количество пакетов трафика.

После приведения входных данных к необходимому виду, сформулируем постановку задачи. Пусть дано множество записей о трафике, проходящем через сеть мультиоблачной платформы, в виде множества вида (3). Требуется построить множество неконфликтных правил $R = r_i, i = \overline{1, m}, m \to min$ вида (4).

Данная работа предполагает решение задачи итерационным методом в два этапа. На первом этапе осуществляется построение первоначального списка правил R_1 путем классификации множества X на два класса. На втором этапе осуществляется построение множества R_{opt} путем кластеризации множества R_1 с последующим выведением правил из кластеров, то есть построением множества R_2 с последующей оптимизацией этого множества алгоритмом бесконфликтной оптимизации, то есть построением множества R_{opt}.

Пусть дано множество X вида (4), требуется построить список правил, формирующих множество R_1. Это означает, что нужно построить классифицирующую функцию вида $f(x) : X \to R$.

Существует множество различных методов для решения данной задачи. В рамках настоящего исследования будем использовать классификацию на основе нейронной сети. Это обусловлено тем, что классификация является классической задачей для нейросетевых методов. Оптимальной для решения подобной задачи является архитектура нейронной сети типа многослойный перцептрон.

Количество нейронов на входном слое вычисляется в зависимости от входных данных. В данном случае размер входного слоя нейронной сети составит 99 нейронов. На запись IP-адреса приходится по 32 бита, а на запись портов по 16 бит, число рассматриваемых протоколов = 7, следовательно, для их записи необходимо 3 бита. Для обучения выбранной модели нейронной сети будем использовать алгоритм обратного распространения ошибки.

2.1. Кластеризация правил межсетевого экранирования

Одним из нередко встречаемых типов атак на компьютерные сети является постоянно повторяющаяся попытка получить доступ к определенному ресурсу с расчетом, что типовой пакет трафика данной атаки на сеть будет удовлетворять определенному правилу межсетевого экрана. Цель атаки – найти правило, находящееся

внизу списка. В результате такой атаки происходит стремительный рост нагрузки на процессор и увеличение объема потребляемой оперативной памяти на межсетевом экране. Это приводит к падению производительности системы безопасности в целом. В целях снижения затрат памяти и времени обхода списка правил межсетевым экраном, необходимо решить задачу сокращения списка правил, не потеряв при этом характеристики, отвечающие за защищенность мультиоблачной платформы. Для этих целей имеет смысл разбить правила на кластеры с целью выведения новых, обобщенных правил из них.

Сформулируем математическую постановку задачи кластеризации правил межсетевого экранирования. Пусть дано множество правил $R = \{r_i\}, i = \overline{1, m}$, требуется построить разбиение выборки на непересекающиеся подмножества, называемые кластерами, таким образом, чтобы каждое подмножество состояло из близких по некоторой метрике объектов, другими словами, построить кластеризующую функцию $f(r) : R \to Y$, которая ставит каждому элементу множества R в соответствие элемент множества $Y = \{y_1, y_2, \ldots\}$ – множества номеров кластеров.

Важным аспектом при решении задачи кластеризации является выбор функции расстояния, или метрики. Метрика является мерой близости, которой пользуются алгоритмы. В рамках исследования в качестве метрики было взято евклидово расстояние по следующим признакам: IP-адреса отправителя и получателя, соответствующие им порты, протокол, по которому осуществляется соединение и решение о допустимости соединения. Таким образом, формула функции расстояния имеет следующий вид:

$$D(r_1, r_2) = \sqrt{a(r_{1,2} - r_{2,2})^2 + b(r_{1,3} - r_{2,3})^2 + c(r_{1,5} - r_{2,5})^2 + d(r_{1,7} - r_{2,7})^2}. \quad (5)$$

Эта функция использовалась с эмпирически подобранными параметрами $a = 0.55$; $b = 0.55$, $c = 38745.6$; $d = 2^4 - 1$.

2.2. Алгоритм выведения правил межсетевого экрана из кластеров

Следующим этапом является выведение правил из кластеризованного списка правил. Входными данными будет являться список правил, разбитый на кластеры $R_{k,l}$. На выходе алгоритма ожидается список обобщенных правил R_{opt}. В рамках данного исследования был разработан алгоритм, приведенный в виде псевдокода ниже. Список кластеров обозначим u. Тогда для всех кластеров из u справедливо:

Вход: Кластеризованный список правил R_{kl}, список кластеров u

Выход: Общий список правил R_{opt}

$r_{i,1} = min(r_{k,1})/mask = 32 - log_2(max(r_{k,1}) - min(r_{k,1}))$; $r_{i,2} = min(r_{k,2})/mask = 32 - log_2(max(r_{k,2}) - min(r_{k,2}))$; $r_{i,3} = \{r_{1,3}, \ldots, r_{k,3}\}$; $r_{i,4} = \{r_{1,4}, \ldots, r_{k,4}\}$; $r_{i,5} = \{r_{1,5}, \ldots, r_{k,5}\}$; $r_{i,6} = r_{i,6}$.

2.3. Алгоритм оптимизации списка правил ранжированной сортировкой

Важным параметром для списка правил, помимо широты охвата защищаемых ресурсов и величины списка, является также расстановка правил. Защитить сеть мультиоблачной платформы от атак, направленных на выполнение правила, находящегося в конце списка, возможно при помощи сортировки списка правил. Цель сортировки – поместить наиболее часто используемые правила вверх списка, чтобы исключить расходование ресурсов системы на последовательную проверку большого количества правил. За счет оптимизации данного процесса возможна существенная экономия и машинного времени, и снижение нагрузки на устройстве, осуществляющем функции межсетевого экрана.

Для решения этой задачи был разработан следующий алгоритм, представленный ниже

Дано:

– Множество заголовков трафика, проходящего по правилу «Запретить всё» – X;

– Неконфликтный список правил $R = \{R_i\}, i = 1, n$, где Rn – «Запретить всё».

Шаг 1: Задать каждому правилу и каждому элементу множества X вес по формуле $w_i = \frac{k_i}{k}$, где k_i – количество трафика, проходящего по правилу i, k – общее количество пакетов, проходящих через сеть.

Шаг 2: IF $w_n > w^*$ THEN перейти к **шагу 3**, ELSE перейти к **шагу 5**.

Шаг 3: Создать запрещающее правило R_{n-1}, по данным элемента X.

Шаг 4: Отсортировать множество R по величине, поставить правило «Запретить всё» на последнее место. Перейти к **шагу 2**.

Шаг 5: Конец алгоритма.

Данный алгоритм не уменьшает размер списка, но позволяет составить оптимальный список правил. При этом обеспечивается защищенность против атак, направленных на возникновение сбоев в работе средств межсетевого экранирования.

3. Экспериментальные исследования

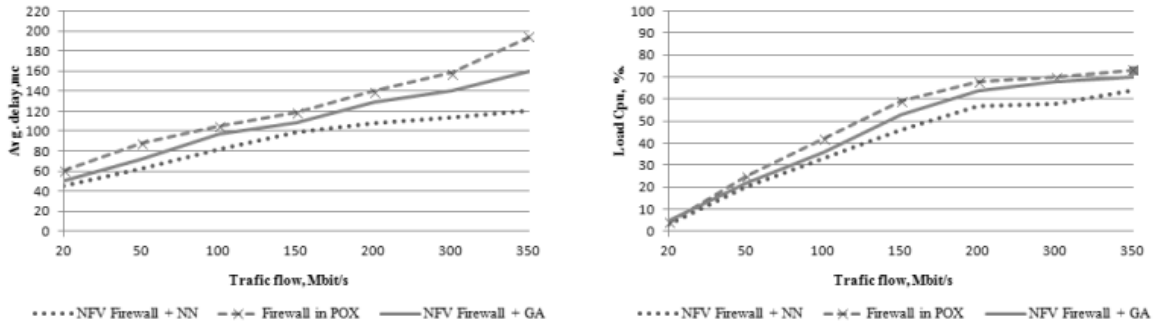
На базе предложенного решения в рамках исследования реализован модуль адаптивного межсетевого экрана. Программное обеспечение реализовано в виде виртуальной сетевой функции на базе Open Platform for NFV (OPNFV), собранной в виде контейнера Docker. Для сравнения был выбран традиционный межсетевой экран, являющийся пакетным фильтром, реализованный внутри платформы POX.

Сравнительный анализ проводился путем сопоставления результатов работы традиционного межсетевого экрана и разработанного программного модуля, а также с использованием оптимизации списка правил сетевой безопасности на основе генетического алгоритма. Перед проведением эксперимента разработанный программный модуль был обучен, а также был разработан сопоставимый набор правил для традиционного межсетевого экрана, что позволяет проводить корректное сравнение сопоставляемых средств. В рамках исследования оценивалась работа при различной нагрузке по двум ключевым показателям: время отклика в сети; нагрузка центрального процесса на межсетевом экране.

Для проведения нагрузочного тестирования сценариев экспериментальных атак была создана виртуальная сеть в облачной системе OpenStack. Она включает 4 коммутатора OpenFlow (2 HP 3500yl, 2 Netgear GSM7200), 8 вычислительных узлов (32 ГБ ОЗУ, 4 ядра), 1 сервер (32 ГБ ОЗУ, 8 ядер) с контроллером OpenFlow и 1 сервер (32 ГБ ОЗУ, 4 ядра) для мониторинга работы виртуальных сетевых функций. В качестве топологии выбрана fat tree с тремя уровнями. Маршрутизаторы имеют скорость соединения 1000 Мбит/с. Вычислительные узлы подключаются к маршрутизатору третьего уровня через сетевые соединения второго уровня со скоростью 1000 Мбит/с. В развернутой инфраструктуре было подготовлено 100 виртуальных машин (атакующих узлов). Среди них случайным образом выбирался один узел, который контролировал атаку. Так же случайным образом выбирались пять атакованных виртуальных машин (сервисный хост).

На выбранные узлы при помощи специально предназначенного для этого средства – hping3 создавалась нагрузка, характерная для сетевых атак типа DDoS. В рамках исследования генерировались пакеты различной длины с предварительным выбором узла назначения. Это позволило максимально приблизить структуру экспериментального трафика к реальным данным. Сам эксперимент состоял в замере показателей производительности сети и оборудования при последовательном увеличении интенсивности потоков трафика в рамках разовой нагрузки в диапазоне от 20 до 350 Мбит/с.

В рамках эксперимента проведена оценка времени отклика – одного из ключевых параметров, применяемых при анализе производительности сети и сетевых устройств. Она показывает, сколько времени проходит с момента отправки запроса пользователем до момента ответа на запрос сервисом. В соответствии с правилами эксперимента были сняты данные о времени отклика с сети.



a) b)

Рис. 1. Результаты экспериментального исследования характеристик сети и параметров работы модулей узлов формирования правил для межсетевого экрана

Fig 1. The results of an experimental study of the characteristics of the network and the parameters of the modules of the nodes forming the rules for the firewall

Список правил безопасности, созданный построенным модулем формирования правил для межсетевого экрана при нагрузке, в том числе лавинообразной, дает сокращение во времени отклика до 20%, что говорит об эффективности разра-

ботанного в исследовании подхода (Рис. 1, a). Нагрузка на центральный процессор межсетевого экрана является не менее важным параметром при рассмотрении средств безопасности. Это показатель того, является ли устройство загруженным или неэффективно используемым. Исходя из полученных результатов имеем, что оптимизированный набор правил, построенный с использованием рассмотренного в исследовании подхода, существенно снижает нагрузку на процессор межсетевого экрана (Рис. 1, b).

Заключение

В ходе исследования был проведен обзор существующих решений в области межсетевого экранирования, был осуществлен выбор архитектуры нейронной сети для достижения поставленной цели, а именно гибридной искусственной нейронной сети, состоящей внутри из двух сетей – классификатора на основе многослойного персептрона и кластеризатора на основе нейронной сети Кохонена, исследованы существующие подходы и алгоритмы для решения задачи обеспечения безопасности информационных ресурсов сети мультиоблачной платформы. В рамках исследования была разработана модель описания правил безопасности межсетевого экрана и на базе нее разработаны алгоритмы автоматического построения и оптимизации списка безопасности межсетевого экрана.

В ходе испытаний было выявлено, что предлагаемый данным исследованием подход дает прирост производительности по двум ключевым параметрам: по времени отклика – в среднем на 20% с ростом нагрузки и по нагрузке центрального процесса экранирующего устройства – в среднем на 4,5% с ростом нагрузки. Таким образом, разработанный подход является эффективным для решения практических задач.

Список литературы / References

[1] Chomsiri T., et al., “An Improvement of Tree-Rule Firewall for a Large Network: Supporting Large Rule Size and Low Delay”, *Proceedings of 2016 IEEE Trustcom/BigDataSE/ISPA* (Tianjin, Aug 23–26), IEEE, 2016, 178–184.

[2] Zhichao P., et al., “A Load-Balancing and State-Sharing Algorithm for Fault-Tolerant Firewall Cluster”, *Proceedings of 2017 4th International Conference on Information Science and Control Engineering (ICISCE)* (Changsha, July 21–23), IEEE, 2017, 34–37.

[3] Nivedita, Kumar R., “An improved Linux firewall using a hybrid frame of netfilter”, *2017 International Conference on Trends in Electronics and Informatics (ICEI)* (Tirunelveli, May 11–12), IEEE, 2017, 657–662.

[4] Kaur S., et al., “Implementing openflow based distributed firewall”, *Proceedings of 2016 International Conference on Information Technology (InCITe) – The Next Generation IT Summit on the Theme – Internet of Things: Connect your Worlds* (Noida, Oct 6–7), IEEE, 2016, 172–175.

[5] Papagrigoriou A., et al., “A firewall module resolving rules consistency”, *Proceedings of 2017 13th Workshop on Intelligent Solutions in Embedded Systems (WISES)* (Hamburg, June 12–13), IEEE, 2017, 47–50.

[6] Rengaraju P., et al., “Investigation of security and QoS on SDN firewall using MAC filtering”, *Proceedings of 2017 International Conference on Computer Communication and Informatics (ICCCI)* (Coimbatore, Jan 5–7), IEEE, 2017, 1–5.

[7] Zhang L., Huang M., "A Firewall Rules Optimized Model Based on Service-Grouping", *Proceedings of 2015 12th Web Information System and Application Conference (WISA)* (Jinan, Sept 11–13), IEEE, 2015, 142–146.

[8] Ertam F., Kaya M., "Classification of firewall log files with multiclass support vector machine", *Proceedings of 2018 6th International Symposium on Digital Forensic and Security (ISDFS)* (Antalya, March 22–25), IEEE, 2018, 1–4.

[9] Atighetchi M., Adler A., "A Framework for Resilient Remote Monitoring", *Proceedings of 2014 7th International Symposium on Resilient Control Systems (ISRCS)* (Denver, Aug 19–21), IEEE, 2014, 1–8.

[10] Parfenov D., Bolodurina I., "Methods and algorithms optimization of adaptive traffic control in the virtual data center", *Proceedings of 2017 International Siberian Conference on Control and Communications (SIBCON 2017)* (Astana, June 29–30), IEEE, 2017, 1–6.

[11] Bolodurina I., Parfenov D., "Development and research of models of organization distributed cloud computing based on the software-defined infrastructure", *Procedia Computer Science*, **103** (2017), 569–576.

Parfenov D. I., Bolodurina I. P., Torchin V. A., "Development and Study of Algorithms for the Formation of Rules for Network Security Nodes in the Multi-Cloud Platform", *Modeling and Analysis of Information Systems*, **26**:1 (2019), 90–100.

DOI: 10.18255/1818-1015-2019-1-90-100

Abstract. As part of the study, existing solutions aimed at ensuring the security of the network perimeter of the multi-cloud platform were considered. It is established that the most acute problem is the effective formation of rules on firewalls. Existing approaches do not allow optimizing the list of rules on nodes that control access to the network. The aim of the study is to increase the effectiveness of firewall tools by conflict-free optimization of security rules and the use of a neural network approach in software-defined networks. The proposed solution is based on the sharing of intelligent mathematical approaches and modern technologies of virtualization of network functions. In the course of experimental studies, a comparative analysis of the traditional means of rule formation, the neural network approach, and the genetic algorithm was carried out. It is recommended to use the multilayer perceptron neural network classifier for automatic construction of network security rules since it gives the best results in terms of performance. It is also recommended to reduce the size of the firewall security rule list using the Kohonen network, as this tool shows the best performance. A conflict-free optimization algorithm was introduced into the designed architecture, which produces finite optimization by ranking and deriving the most common exceptions from large restrictive rules, which allows increasing protection against attacks that are aimed at identifying security rules at the bottom of the firewall list. On the basis of the proposed solution, the adaptive firewall module was implemented as part of the research.

Keywords: adaptive firewall, software-defined network, multi-cloud platforms, neural network, network function virtualization, cyber security

On the authors:
Denis I. Parfenov, PhD, orcid.org/0000-0002-1146-1270
Orenburg State University,
13 Pobedy pr., Orenburg 460018, Russia, e-mail: parfenovdi@mail.ru

Irina P. Bolodurina, PhD, orcid.org/0000-0003-0096-2587,
Orenburg State University,
13 Pobedy pr., Orenburg 460018, Russia, e-mail: prmat@mail.osu.ru

Vadim A. Torchin, graduate student, orcid.org/0000-0002-5315-6047
Orenburg State University,
13 Pobedy pr., Orenburg 460018, Russia, e-mail: vadim.torchin@gmail.com

Acknowledgments:
The work was supported by Russian Foundation for Basic Research, Projects No. 18-07-01446, No. 16-29-09639.