УДК 004.72

МОДЕЛИРОВАНИЕ СЦЕНАРИЕВ БЕЗОПАСНОСТИ В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ

А. А. Пименова, студент кафедры ВТ ПГУ, Пенза Россия;
orcid.org/0000-0001-6659-2211, e-mail: foxik.alis@yandex.ru
Д. Д. Никитин, студент кафедры ВТ ПГУ, Пенза, Россия;
e-mail: nddpnz@gmail.com
К. И. Никишин, к.т.н., доцент кафедры ВТ ПГУ, Пенза, Россия;
orcid.org/0000-0001-7966-7833, e-mail: nkipnz@mail.ru

В настоящее время к компьютерным сетям накладываются большие требования по повышению быстродействия, производительности и отказоустойчивости. Важным критерием любых сетей являются отказоустойчивость сети и защита от различных атак на сеть. В статье рассматриваются принципы безопасности и исследование атак в программно-конфигурируемых сетях (ПКС). ПКС могут быть подвержены различным видам атак. Одними из самых распространенных атак являются DDoS (распределенный отказ в обслуживании, Distributed Denial of Service) и MITM (Main in the middle) атаки. ***Цель работы*** *– исследование принципов безопасности в ПКС и изучение различных атак, таких как DDoS и MITM, с использованием цветных сетей Петри и пакета моделирования CPN Tools. Задачами исследования являются подходы обнаружения атак на каждом коммутаторе OpenFlow любого пакета с измененными данными, анализ подмены MAC-адреса в потоке пакетов в сети. Иерархические модели на сетях Петри позволили не просто исследовать функционирование и поведение ПКС и ее принципов безопасности, но и верифицировать модель и алгоритм защиты коммутатора и ПКС от DDoS, MITM атак.*

Ключевые слова*: программно-конфигурируемые сети, контроллер, коммутатор, OpenFlow, Flow Table, безопасность, DDoS атака, MITM атака, сети Петри, CPN Tools.*

DOI: 10.21667/1995-4565-2022-82-60-72

Введение

На сегодняшний день компьютерные сети играют большую роль в осовремененном обществе, промышленности. Они позволяют передавать большие данные по сети различного рода: пользовательская информация, управляющая информация. Классические сети строились на основе Ethernet с поддержкой качества обслуживания (Quality of Service, QoS) [1, 2].

В настоящее время к компьютерным сетям предъявляются большие требования по повышению быстродействия, производительности и отказоустойчивости. В классической сети Ethernet существуют ограничения по быстродействию и обработке трафика в сети. Для этих целей были предложены программно-конфигурируемые сети (ПКС) [3, 4], а именно для повышения быстродействия сети, упрощения администрирования сети и упрощения передачи трафика за счет разграничения уровней обработки в ПКС [5, 6].

Однако, кроме повышения быстродействия и производительности, важным критерием являются отказоустойчивость сети и защита от различных атак на сеть. В статье рассматриваются принципы безопасности и задачи исследования атак в ПКС.

Теоретическая часть

ПКС могут быть подвержены различным видам атак. Одними из самых распространенных атак являются DDoS (распределенный отказ в обслуживании, Distributed Denial of Service) и MITM (Main in the middle) атаки. В случае с DDoS атаками вся сеть полностью выходит из строя из-за слишком большого количества подключений.

Атака посредника или атака «человек посередине» (MITM) направлена на перехват сообщений между двумя или более машинами без уведомления атакованной машины. Злоумышленник может прочитать, изменить или заменить трафик. Сетевые устройства используют протокол ARP для создания привязок между MAC и IP-адресами в сети. Этот протокол жизненно важен для любой сети, но он не был разработан для проверки подлинности сообщений и борьбы с вредоносными хостами. Каждый раз, когда пользовательская машина создает пакет для назначения, она записывает правильный IP-адрес назначения, но при этом использует MAC-адрес злоумышленника в пакете заголовка [7].

Поэтому для повышения отказоустойчивости сетей подключают несколько контроллеров для обеспечения бесперебойной работы сети. Одиночный контроллер подвержен таким атакам и может выйти из строя.

На рисунке 1 представлена возможная схема DDoS атаки на ПКС с одиночным контроллером. Основными элементами в ПКС являются коммутаторы с поддержкой протокола OpenFlow и контроллеры [8-10]. Могут также присутствовать дополнительные коммутаторы и роутеры для передачи трафика по оптимальному маршруту [11]. Основная цель DDoS атак – получить информацию, перехватать, изменить трафик на пользовательских машинах, как показано на рисунке 1. Более подробное описание DDoS атак на подобные архитектуры ПКС рассмотрено в статье [12].

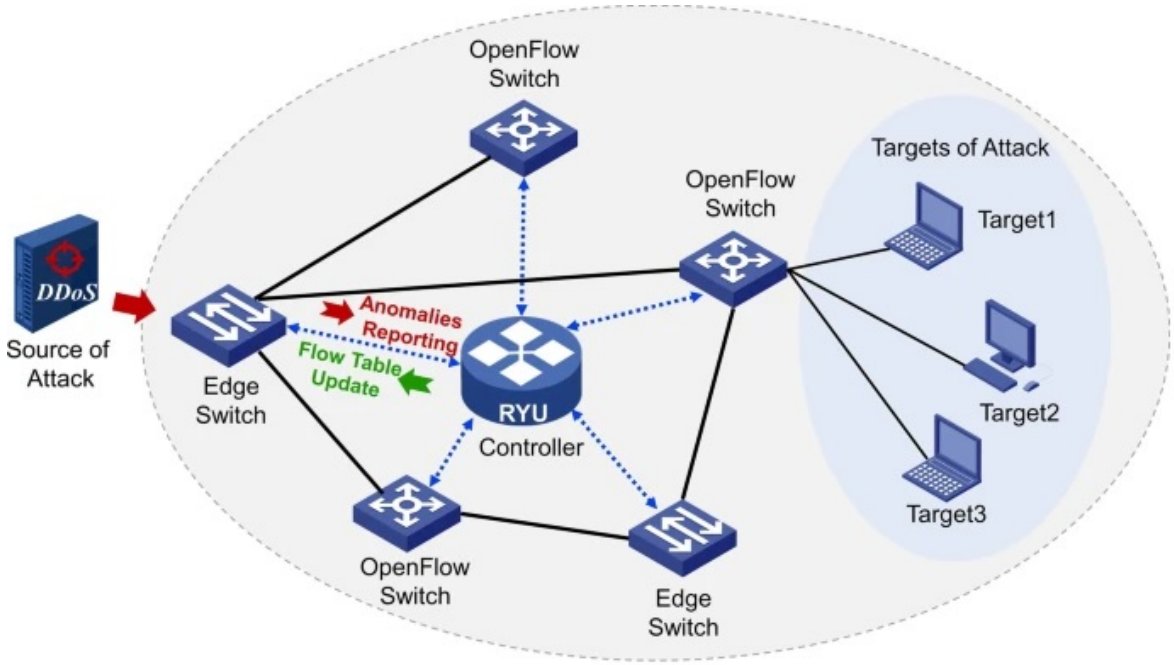

Рисунок 1 – DDoS атака в ПКС
Figure 1 – DDoS attack in SDN

В экспериментальной части будут рассмотрены модели и алгоритмы, демонстрирующие работу сети в условиях различных атак на них. Модели были разработаны на основе цветных временных иерархических сетей Петри с помощью свободно распространяемого пакета CPN Tools, который наилучшим образом подходит для исследования компьютерных сетей, сетевых протоколов, алгоритмов и исследования верификации [13, 14].

Экспериментальные исследования

Первая модель ПКС состоит из трех коммутаторов (позиции Switch1, Switch2, Switch3), одиночного контроллера ПКС (позиция Controller), одного приложения (позиция Application) и сетевого интерфейса (позиция Network interface). Модель представлена на рисунке 2.

Моделирование различных топологий ПКС [15, 16] в случае атак злоумышленников было рассмотрено в статье [17]. Однако в статье не описаны этапы алгоритма передачи трафика

в сети и влияние атаки на модель, представлен абстрактный вид трафика без привязки к форматам трафика и данным. Все эти недостатки послужили для разработки собственной модели с описанием конкретных позиций, переходов модели и реальными пользовательскими данными.

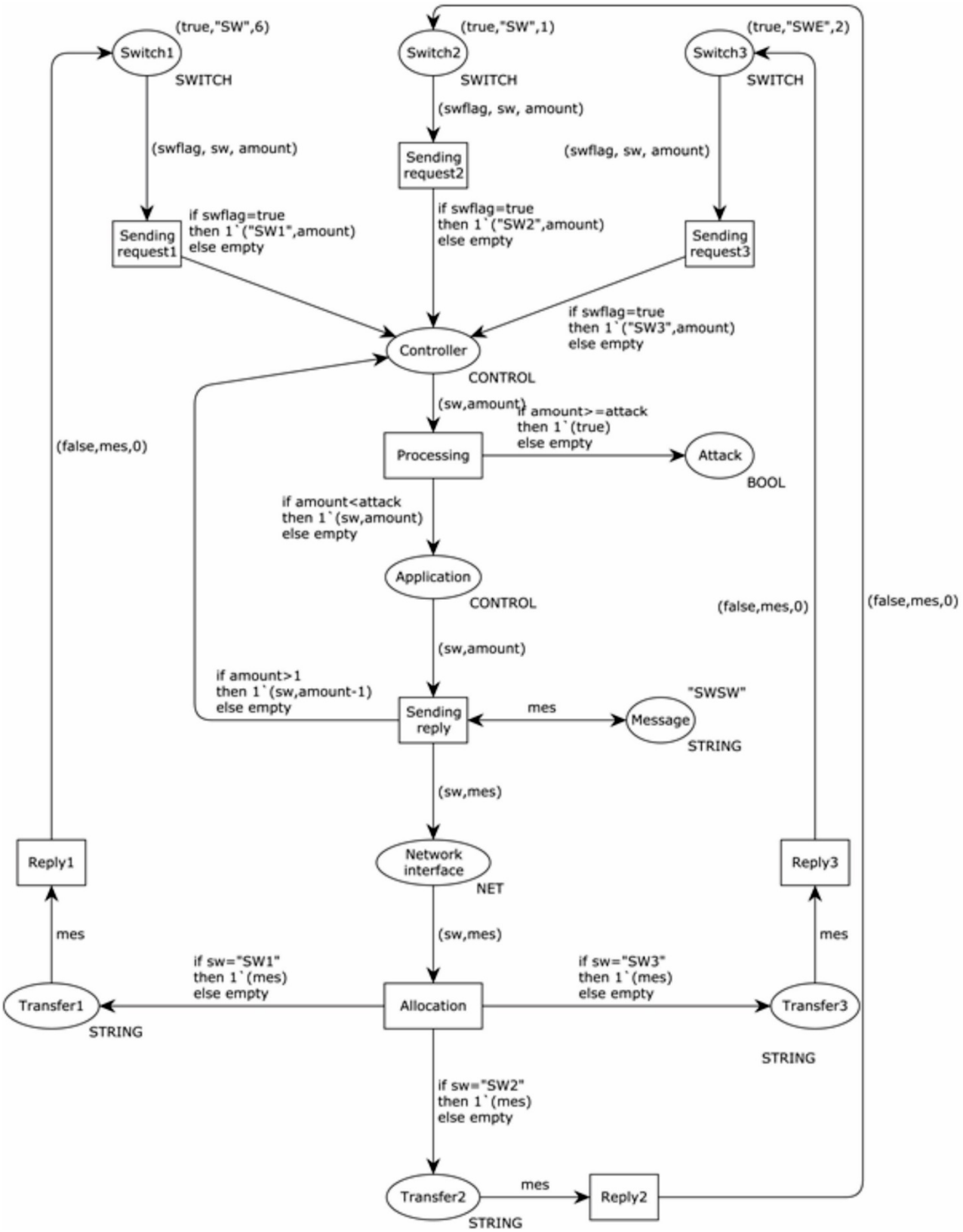

Рисунок 2 – Работа ПКС с одиночным контроллером и DDoS атаками
Figure 2 – SDN operation with a single controller and DDoS attack

Рассмотрим описание некоторых цветов, используемых в модели. Цвет для хранения подтверждения, сообщения и количества запросов – colset SWITCH = product BOOL*STRING*INT. Цвет для хранения имени отправителя и количества запросов – colset CONTROL = product STRING*INT. Цвет для хранения имени получателя и сообщения – colset NET = product STRING*STRING.

Коммутатор отправляет своё имя и количество запросов контроллеру, если переменная swflag флага подтверждения отправки запроса находится в истинном положении. Если данный флаг не готов к отправке запроса, то коммутатор ничего не отправляет контроллеру. Этот процесс идентичен для всех трёх коммутаторов.

Перед отправкой запросов приложению идет проверка их количества, за подсчет количества запросов отвечает переменная amount. Введена константа attack с установленным значением 6, константа служит порогом для определения атак на ПКС.

Если количество запросов amount превышает значение константы attack, то это воспринимается как атака на сеть и приводит к нарушению её функционирования. Если количество запросов меньше значения константы attack, то происходит передача информации и получение ответного сообщения.

Данный процесс повторяется, пока количество запросов не станет равно 0. Затем с помощью условий sw=”SW1”, sw=”SW2” и sw=”SW3” проверяется, какому коммутатору будет отправлено сообщение mes. После этого нужному коммутатору будет отправлено значение флага подтверждения отправки запроса в сброшенном состоянии, сообщение и количество запросов, равное 0.

Следующая модель демонстрирует работу ПКС в случае MITM атак и служит для анализа контроля безопасности. За основу была взята модель в статье [7], в модели были доработаны алгоритмы в части поиска правила по таблице потоков и исключения нескольких правил для одного пакета, доработана подсеть Actions в части учета не только MAC-адреса назначения, но и MAC-адреса источника, IP-адреса назначения, IP-адреса источника.

Модель имеет три уровня: верхний уровень, который представляет топологию сети; второй уровень, который детализирует клиента, сервер, атакующую машину и коммутатор; третий уровень, который представляет детали некоторых компонентов коммутатора.

Верхний уровень модели CPN представляет собой топологию ПКС, состоящую из четырех компонентов: клиента, сервера, машины злоумышленника и одного коммутатора OpenFlow. Для каждой машины есть два места: одно для отправки TX, а другое для приема пакетов RX.

В этой модели существует только одна последовательность связи, начинающаяся с машины злоумышленника, за которой следует клиентская машина, и, наконец, серверная машина. Эта последовательность была определена для имитации сценария атаки, в котором злоумышленник запускает атаку до того, как начнется передача трафика между клиентской машиной и сервером. Подсеть коммутатора OpenFlow имеет три порта и соединяет машины в ПКС (представлена на рисунке 3).

Эта модель состоит из шести переходов (прием SW P1, передача SW P1, прием SW P2, передача SW P2, прием SW P3 и передача SW P3) и шести позиций (P1, P2, P3, Warehouse, PreAct и RULE Action).

И дополнительно имеются две подсети ChkRules и Actions, которые будут подробно описаны далее. Каждый порт коммутатора Openflow имеет два подключенных к нему перехода. Один обрабатывает прием пакетов от машин, подключенных к коммутатору (прием SW P1, прием SW P2 и прием SW P3), а другой обрабатывает передачу пакетов на подключенные к нему машины (передача SW P1, передача SW P2 и передача SW P3).

Когда коммутатор получает пакет, он сохраняет пакет в позицию Warehouse. Подсеть ChkRules хранит правила таблиц потоков и отвечает за соответствие одного из правил каждому пакету. Подсеть ChkRules считывает список номеров пакетов и заголовок пакета, срав-

нивает поля заголовка с таблицей в поисках необходимого правила и определяет, какие действия коммутатор будет применять к пакету [18].

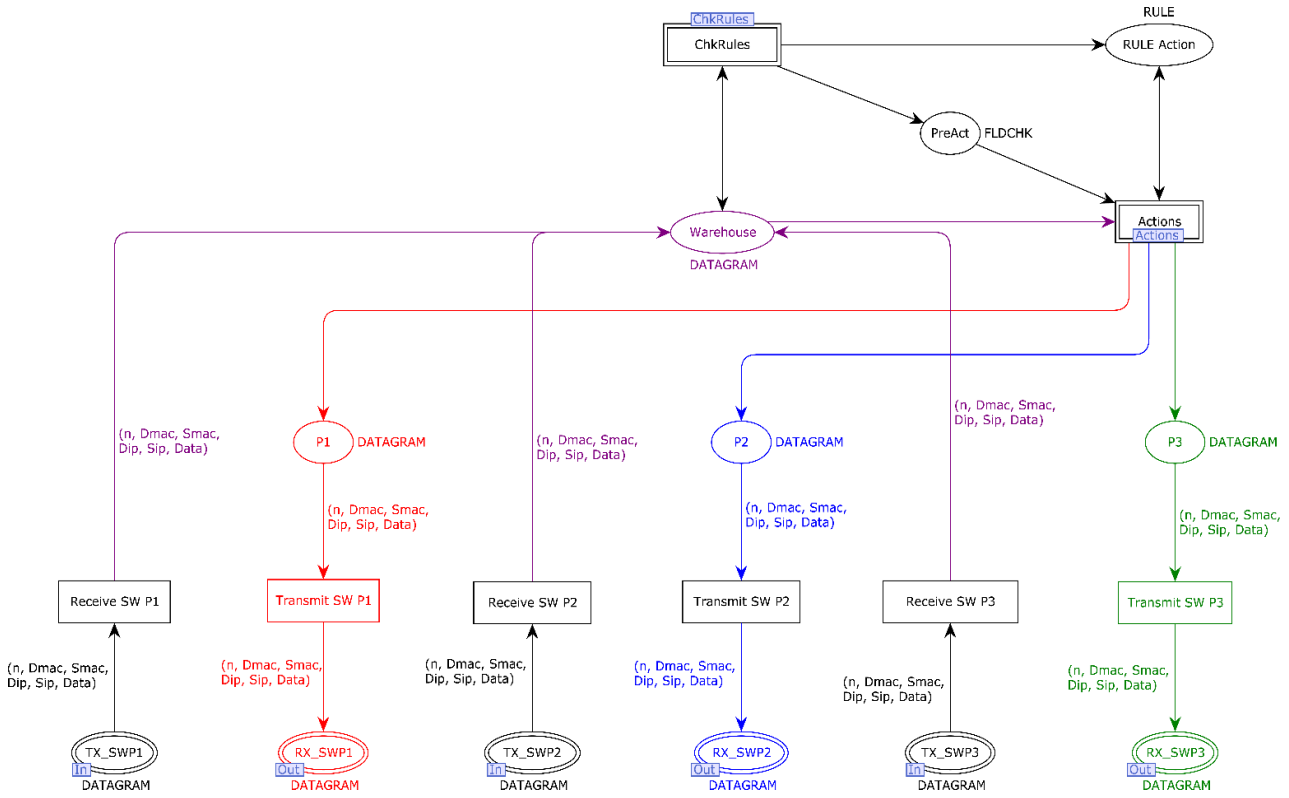

Рисунок 3 – Подсеть коммутатора OpenFlow
Figure 3 – Subnet of OpenFlow switch

Пакет и номер согласованного правила сохраняются в позиции PreAct, а копия согласованного правила потока сохраняется в позиции RULE Actions. Подсеть Actions считывает информацию, хранящуюся в позициях PreAct и RULE Action, применяет действия, определенные правилом потока, и доставляет пакет в один из временных буферов порта коммутатора (места P1, P2 или P3).

Подсеть ChkRules отвечает за проверку того, какое правило потока должно быть применено к конкретному пакету (рисунок 4). Эта модель состоит из шести переходов (getNumb, Check Datagram, ChkDmac, ChkSmac, ChkDip и ChkSip) и семи мест (Rule Count, RULE Table, PreDmac, PreSmac, PreDip и PreSip, n).

Переход Check Datagram считывает пакет, извлекает номер пакета, получает номер первого правила потока, подлежащего проверке, из позиции Rule Count и пересылает эту информацию в позицию PreDmac.

Позиции PreDmac, PreSmac, PreDip и PreSip используются этой моделью в качестве области этапа, которые используются для хранения номера правила потока и номера пакета между каждым полем заголовка пакета утверждение. Переход RULE Table хранит таблицу правил потока, и каждая запись правила потока состоит из: номера приоритета, MAC-адреса назначения пакета, MAC-адреса источника пакета, IP-адреса назначения пакета, IP-адреса источника пакета, MAC-адреса назначения действий, MAC-адреса источника действий, IP-адреса назначения действий, IP-адреса источника действий и порта коммутатора Openflow, на который должен быть доставлен пакет.

Конечные MAC-адрес, MAC-адрес источника, IP-адрес назначения и IP-адрес источника используются для сопоставления правила с пакетом, а последние четыре поля заголовка пакета используются для хранения значений полей, которые должны быть заменены переключателем Openflow в пакете. Поля правила потока, заполненные буквой N, означают, что правило принимает любое значение для этого поля в заголовке пакета.

Переход ChkDmac получает номер правила и номер пакета, копию пакета из Warehouse и копию выбранного правила потока. Если MAC-адрес назначения пакета совпадает с MAC-адресом назначения правила потока или MAC-адрес назначения правила потока заполнен буквой N, переход создает отметку в позицию PreSmac с указанием правила и номера пакета. В противном случае переход ChkDmac уменьшает номер правила и создает отметку в позицию PreDmac с новым номером правила и тем же номером пакета.

В отличие от модели в [7], где, в случае несовпадения MAC-адреса назначения с MAC-адресом назначения правила или MAC-адреса с буквой N, в позиции PreSmac создается отметка с номером правила и номером пакета, здесь данная отметка не создается. Переходы ChkSmac, ChkDip и ChkSip реализуют ту же логику для MAC-адреса источника пакета, IP-адреса назначения и IP-адреса источника соответственно.

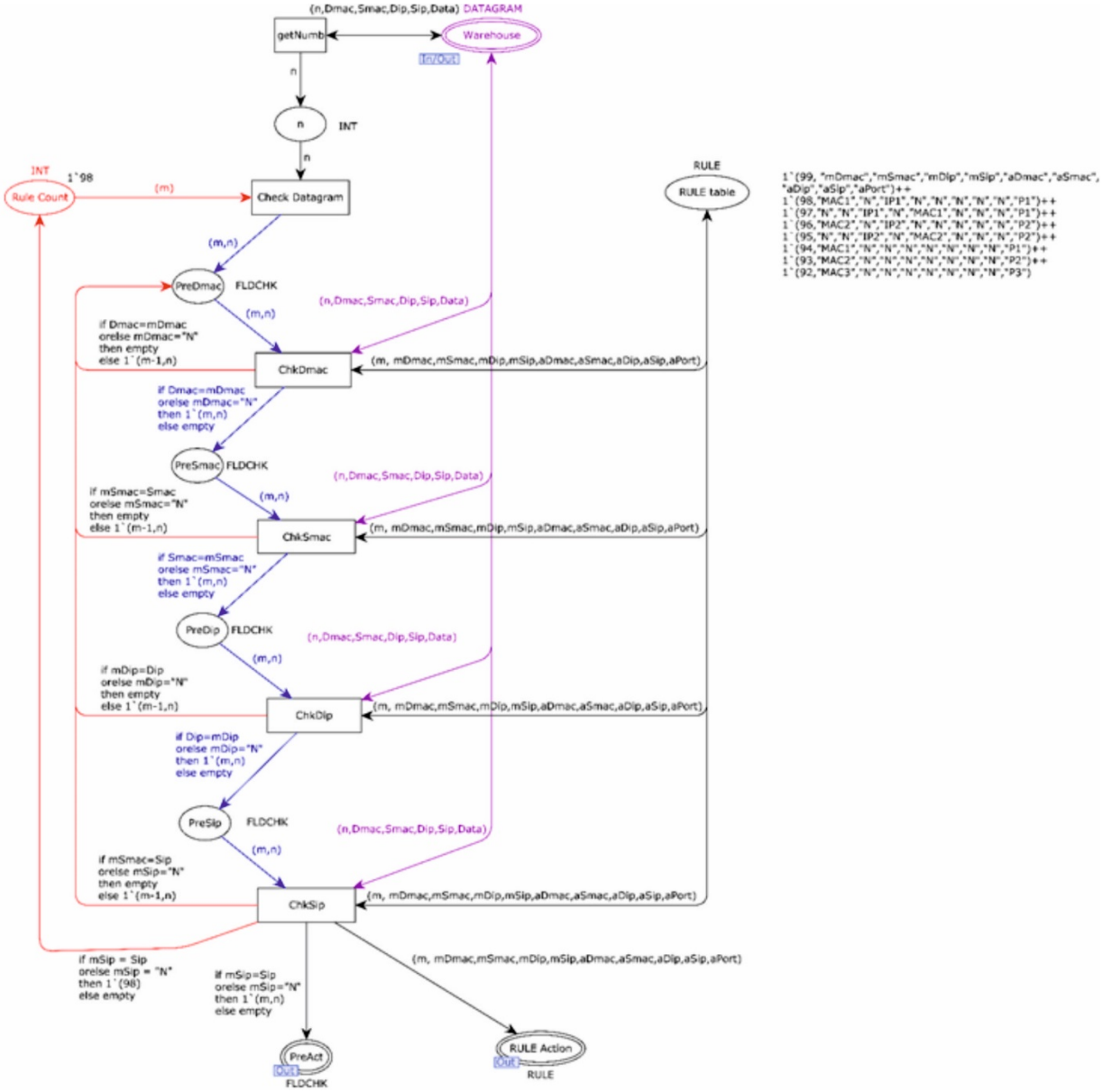

Рисунок 4 – Подсеть поиска и проверки правил ChkRules
Figure 4 – Subnet of search and check rules ChkRules

Переход ChkSip также создает отметку в позиции Rule Count с номером правила первого потока, отметку с соответствующим номером правила и номером пакета в позиции PreAct верхнего уровня и копию согласованного правила на верхнем уровне RULE Actions.

Таким образом, выполнена доработка алгоритма, по которому одному пакету данных соответствует несколько правил. При несовпадении одного из параметров правила и соответствующего параметра пакета данных начинается проверка следующего правила.

Подсеть Actions, изображенная на рисунке 5, отвечает за изменение заголовка пакета, как определено правилом потока, и за доставку пакета на правильный порт назначения коммутатора OpenFlow.

Эта модель состоит из пяти переходов (ActDmac, ActS mac, ActDip, ActSip и Deliver) и четырех позиций (PosADM, PosASM, PosADI и PosASI). Переход ActDmac получает правила и номера пакетов из позиции PreAct, пакет, который хранился в Warehouse, и копию соответствующего правила потока.

Если поле правила aDmac заполнено буквой N, переход перенаправляет исходный пакет в позицию PosADM. В противном случае значение поля правила aDmac переопределяет MAC-адрес назначения пакета и обновленный пакет отправляется в позицию PosADM. Переходы ActSmac, ActDip и ActSip реализуют ту же логику для MAC-адреса источника, IP-адреса назначения и IP-адреса источника соответственно.

В данной модели меняется не только MAC-адрес назначения, но и MAC-адрес источника, IP-адрес назначения, IP-адрес источника, как в модели [7].

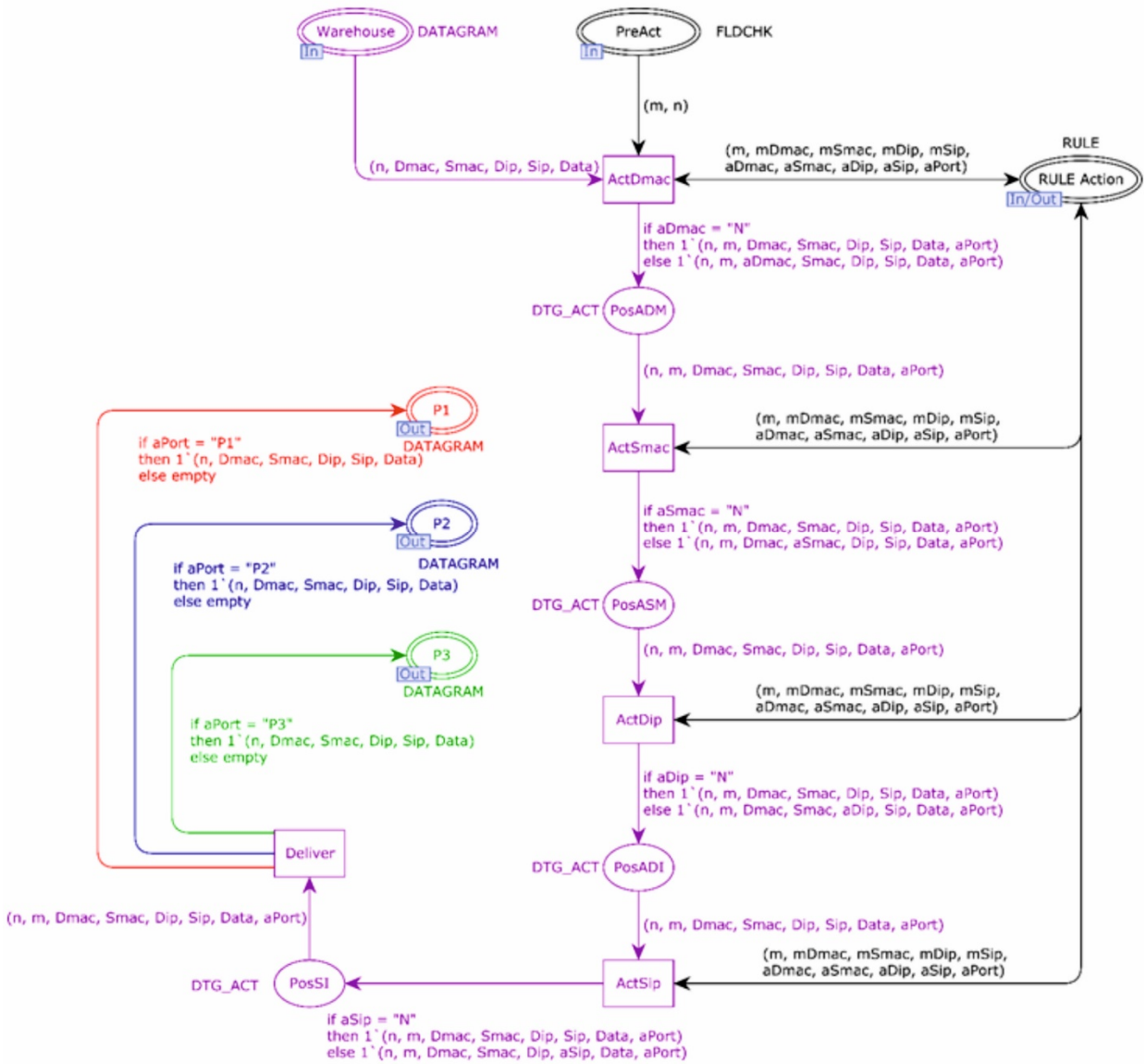

Рисунок 5 – Подсеть выполнения действий над пакетом Actions
Figure 5 – Subnet of action with package Actions

Переход Deliver считывает поле правила aPort и пересылает пакет во временные буферы порта коммутатора Openflow (P1, P2 или P3). Позиции PosADM, PosASM, PosADI и PosASI используются в качестве промежуточной площадки между переходами ActDmac, ActSmac, ActDip, ActSip и Deliver.

Передается исходный пакет в виде кортежа – (1,"MAC2","MAC2","IP1","IP2","STRINGA"). Он приходит на порт TX_SWP1 и имеет неверный MAC-адрес назначения пакета, верные IP-адрес назначения, MAC-адрес и IP-адрес источника, пакет данных.

В процессе работы модели было определено, что для данного пакета необходимо использовать правило № 97, которое содержит правильный MAC-адрес назначения и порт, на который должен прийти пакет данных (рисунок 6).

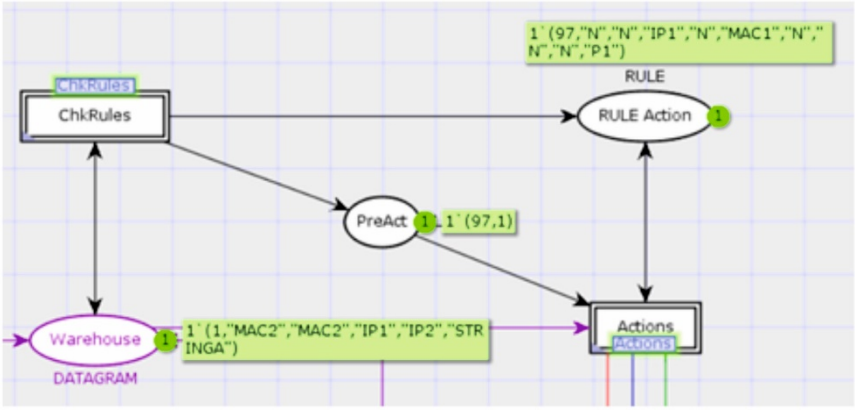

Рисунок 6 – Состояние фишек после выполнения перехода ChkRules для пакета данных № 1
Figure 6 – The state of the chips after performing ChkRules transition for data packet № 1

В результате выполнения перехода Actions в позиции P1 сделана отметка с пакетом данных, в котором изменен MAC-адрес назначения (рисунок 7).

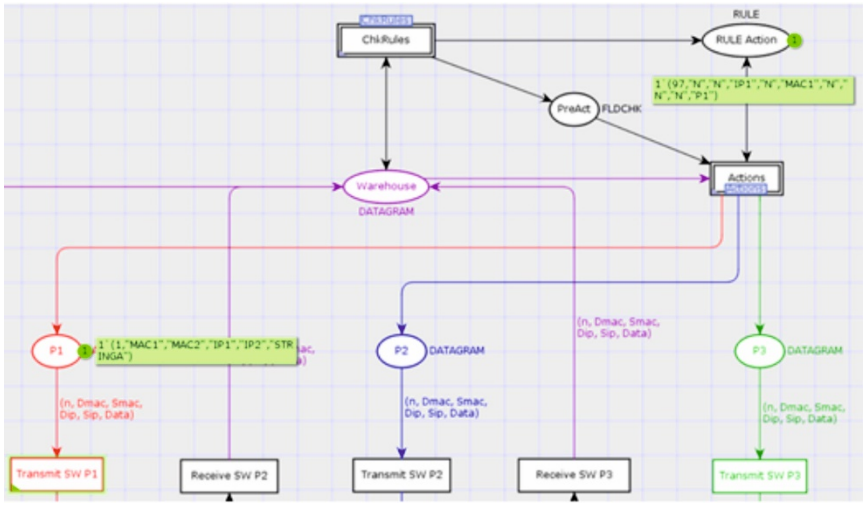

Рисунок 7 – Результат выполнения перехода Actions для пакета данных № 1
Figure 7 – The result of Actions transition for data package № 1

В результате на выход RX_SWP1 приходит пакет с измененным MAC-адресом назначения. Результатом является пакет вида – (1, "MAC1", "MAC2", "IP1", "IP2", "STRINGA"). На рисунке 8 под портом TX_SWP1 показан исходный пакет, под портом RX_SWP1 показан конечный пакет.

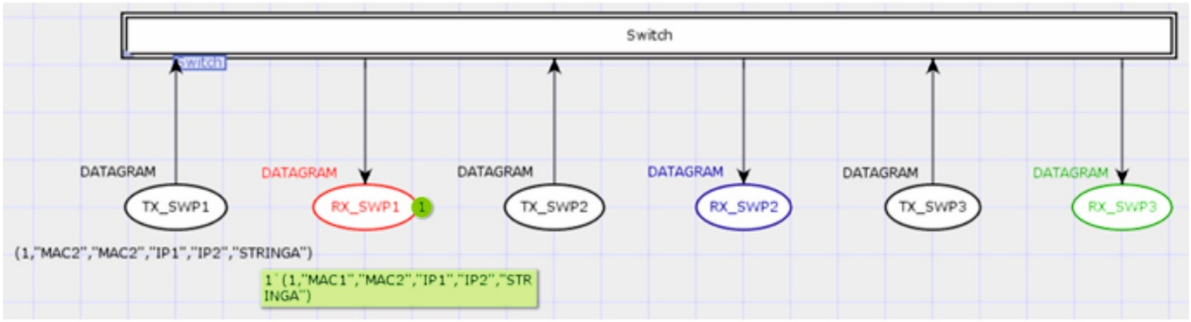

Рисунок 8 – Результат работы модели для пакета № 1
Figure 8 – The result of the model for package № 1

Передается исходный пакет в виде кортежа – (2,"MAC3","MAC2","IP3","IP4","STRINGA"). Он приходит на порт TX_SWP1 и имеет неверный MAC-адрес назначения пакета, верные IP-адрес назначения, MAC-адрес и IP-адрес источника, пакет данных.

В процессе работы модели было определено, что для данного пакета необходимо использовать правило № 92, которое содержит порт, на который должен прийти пакет данных (рисунок 9).

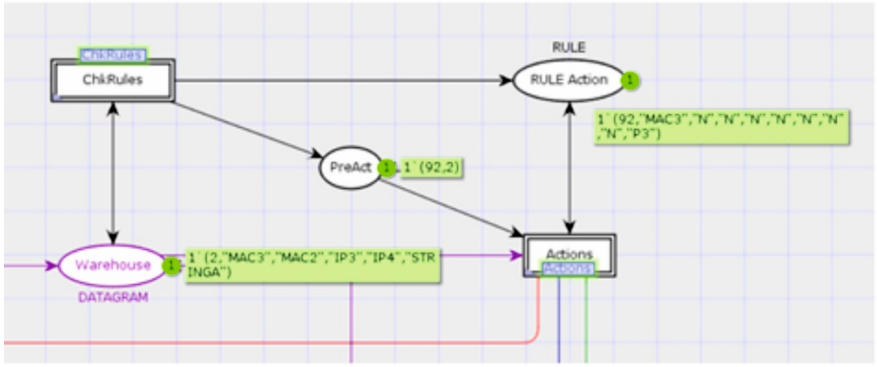

Рисунок 9 – Состояние фишек после выполнения перехода ChkRules для пакета данных № 2
Figure 9 – The state of the chips after performing the ChkRules transition for data packet № 2

В результате выполнения перехода Actions в позиции P3 сделана отметка с пакетом данных, в котором изменен MAC-адрес назначения (рисунок 10).

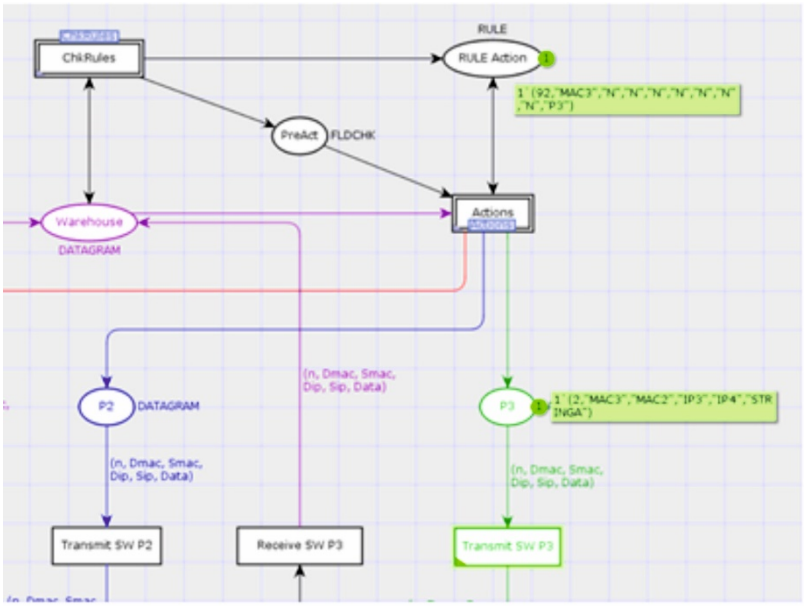

Рисунок 10 – Результат выполнения перехода Actions для пакета данных № 2
Figure 10 – The result of performing the Actions transition for data package № 2

В результате на выход RX_SWP3 приходит исходный пакет. Результатом является пакет: (2, "MAC3", "MAC2", "IP3", "IP4", "STRINGA"). На рисунке 11 под портом TX_SWP1 показан исходный пакет, под портом RX_SWP3 показан конечный пакет.

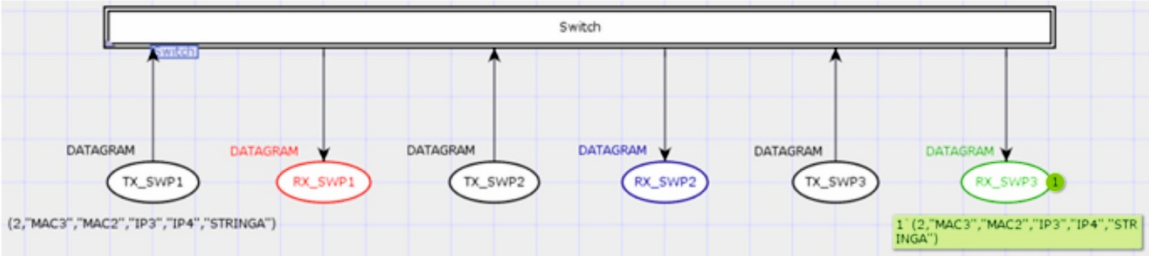

Рисунок 11 – Результат работы модели для пакета № 2
Figure 11 – The result of the model for package № 2

Модель на рисунке 2 позволяет продемонстрировать функциональное поведение работы ПКС с одиночным контроллером в случае DDoS атак, выявить потенциальные места угроз в работе основных узлов ПКС.

Модель на рисунке 3 позволяет выявлять MITM атаки на нескольких уровнях модели OSI. Как видно из моделирования, представленного выше, в анализе выявления изменений в данных участвуют MAC-адреса назначения и источника, IP-адреса назначения и источника, а также осуществляется контроль со стороны таблицы потоков в поиске правила с измененными данными.

Разработанные модели и заложенные в них принципы безопасности помогают обнаруживать DDoS и MITM в ПКС. Модели были построены на основе цветных сетей Петри. Моделирование в пакете моделирования CPN Tools позволило дополнительно верифицировать разработанные модели на тупиковые состояния, определить живость и ограниченность заложенных алгоритмов безопасности в моделях, построить пространство и граф состояний моделей для выявления возможных уязвимостей в алгоритмах.

Заключение

Было проведено исследование принципов безопасности в ПКС и изучение различных атак, таких как DDoS и MITM. С учетом анализа данных атак были построены модели по исследованию атак в ПКС с использованием цветных сетей Петри и пакета моделирования CPN Tools.

Было проведено исследование и построение иерархической модели контроля безопасности для ПКС. Модель позволила обнаруживать измененные данные любого пакета в коммутаторе OpenFlow; анализируется подмена MAC-адреса в потоке пакетов в ПКС. С помощью моделирования на сетях Петри были верифицированы модель и алгоритм защиты коммутатора и ПКС от MITM атак.

Библиографический список

1. **Никишин К. И.** Механизм управления трафиком реального времени в коммутаторе Ethernet // Вестник компьютерных и информационных технологий. 2015. № 10. С. 32-37.

2. **Kizilov E., Konnov N., Nikishin K., Pashchenko D., Trokoz D.** Scheduling queues in the Ethernet switch, considering the waiting time of frames // MATEC Web of Conferences. 2016, vol. 44, pp. 01011-p.1-01011-p. 5.

3. **McKeown N., Anderson T., Balakrishnan H. et al.** Openflow: enabling innovation in campus networks, ACM SIGCOMM Computer Communication Review, 2008, vol. 38, no. 2, pp. 69-74.

4. **Kobayashi M., Seetharaman S., Parulkar G., Appenzeller G., Little J., Van Reijendam J., McKeown N.** Maturing of OpenFlow and Software-Defined Networking Through Deployments // Computer Networks, 2014, vol. 61, pp. 151-175.

5. **Корячко В. П., Перепелкин Д. А., Иванчикова М. А., Бышов В. С., Цыганов И. Ю.** Программная инфраструктура и визуальная среда распределенной обработки потоков данных в програм-

мно-конфигурируемых сетях // Вестник Рязанского государственного радиотехнического университета. 2018. № 65. С. 44-54. DOI: 10.21667/1995-4565-2018-65-3-44-54.

6. **Леохин Ю. Л., Фатхулин Т. Д.** Оценка возможности предоставления гарантированной скорости передачи данных в программно-конфигурируемой оптической сети // Вестник Рязанского государственного радиотехнического университета. 2020. № 71. С. 45-59. DOI: 10.21667/1995-4565-2020-71-45-59.

7. **Bastos D. A., Guelfi A. E., Azevedo M. T., Silva A. A.** Formal modelling and analysis of man in the middle prevention control in Software Defined Network // Revista eletrônica de sistemas de informação. 2021, vol. 11, pp. 16-40.

8. **Shalimov A. et al.** Advanced study of SDN/OpenFlow controllers // Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia. ACM, 2013.

9. **Перепелкин Д. А.** Концептуальный подход динамического формирования трафика программно-конфигурируемых телекоммуникационных сетей с балансировкой нагрузки // Информационные технологии. 2015. Т. 21. № 8. С. 602-610.

10. **Ушакова М. В., Ушаков Ю. А.** Исследование сети виртуальной инфраструктуры центра обработки данных с гибридной программно-конфигурируемой коммутацией // Вестник Рязанского государственного радиотехнического университета. 2021. № 75. С. 34-43. DOI: 10.21667/1995-4565-2021-75-34-43.

11. **Никульчев Е. В., Паяин С. В., Плужник Е. В.** Динамическое управление трафиком программно-конфигурируемых сетей в облачной инфраструктуре // Вестник Рязанского государственного радиотехнического университета. 2013. № 3 (45). С. 54-57.

12. **Yu S., Zhang J., Liu J. et al.** A cooperative DDoS attack detection scheme based on entropy and ensemble learning in SDN // Wireless Com Network. 2021. DOI: 10.1186/s13638-021-01957-9.

13. **Nikishin K., Konnov N.** Schedule Time-Triggered Ethernet // International Conference on Engineering Management of Communication and Technology, EMCTECH 2020. DOI: 10.1109/EMCTECH 49634.2020.9261540.

14. **Никишин К. И., Коннов Н. Н.** Генератор трафика Ethernet на основе цветных сетей Петри // Модели, системы, сети в экономике, технике, природе и обществе. 2016. № 1 (17). С. 299-307.

15. **Никишин К. И.** Моделирование и верификация топологий программно-конфигурируемых сетей // Вестник Рязанского государственного радиотехнического университета. 2022. № 80. С. 67-74. DOI 10.21667/1995-4565-2022-80-67-74.

16. **Никишин К. И.** Моделирование контроллера и верификация процесса передачи данных в программно-конфигурируемых сетях // Вестник Рязанского государственного радиотехнического университета. 2022. № 80. С. 75-83. DOI 10.21667/1995-4565-2022-80-75-83.

17. **Ameen A.** Assuring the SDN security by modelling and comparing SDN proposed topologies using Petri nets // Journal of Engineering Science. 2021, vol. XXVIII, no. 4, pp. 93-105.

18. **Никишин К. И.** Исследование и моделирование таблицы потоков коммутатора Openflow в программно-конфигурируемых сетях // Вестник Рязанского государственного радиотехнического университета. 2022. № 81. С. 42-50. DOI 10.21667/1995-4565-2022-81-42-50.

UDC 004.72

MODELING OF SECURITY PRINCIPLES IN SOFTWARE DEFINED NETWORKS

A. A. Pimenova, student, department of computer science, PSU, Penza, Russia;
orcid.org/0000-0001-6659-2211, e-mail: foxik.alis@yandex.ru
D. D. Nikitin, student, department of computer science, PSU, Penza, Russia;
e-mail: nddpnz@gmail.com
K. I. Nikishin, Ph.D. (Tech.), associate professor, department of computer science, PSU, Penza, Russia;
orcid.org/0000-0001-7966-7833, e-mail: nkipnz@mail.ru

Currently, large requirements are imposed on computer networks to increase speed, performance and fault tolerance. An important criterion of any network is network fault tolerance and protection against vari-

ous attacks on the network. The article discusses the principles of security and the study of attacks in software defined networks (SDN). SDN can be subject to various types of attacks. One of the most common attacks are DDoS (Distributed Denial of Service) and MITM (Main in the middle) attacks. ***The aim of the research*** *is to study the principles of security in SDN and to research various attack, such as DDoS and MITM using color Petri nets and CPN Tools modeling package. The objectives of the study are to detect attacks on each Open-Flow switch of any packet with fake data, to analyze the substitution of MAC address in a packet stream of the network. Hierarchical models on Petri nets made it possible not only to investigate the functioning and behavior of network control system and its security principles, but also to verify the model and the algorithm for protecting switch and control system from DDoS, MITM attacks.*

Key words*: Software Defined Networks, controller, switch, OpenFlow, Flow Table ,safety, DDoS attack, MITM attack, Petri Nets, CPN Tools.*

DOI: 10.21667/1995-4565-2022-82-60-72

References

1. **Nikishin K. I.** Mehanizm upravlenija trafikom real'nogo vremeni v kommutatore Ethernet. *Vestnik komp'juternyh i informacionnyh tehnologij*. 2015, no. 10, pp. 32-37. (in Russian).

2. **Kizilov E., Konnov N., Nikishin K., Pashchenko D., Trokoz D.** Scheduling queues in the Ethernet switch, considering the waiting time of frames. *MATEC Web of Conferences*. 2016, vol. 44, pp. 01011-p.1-01011-p. 5.

3. **McKeown N., Anderson T., Balakrishnan H. et al.** Openflow: enabling innovation in campus networks, ACM SIGCOM. *Computer Communication Review*, 2008, vol. 38, no. 2, pp. 69-74.

4. **Kobayashi M., Seetharaman S., Parulkar G., Appenzeller G., Little J., Van Reijendam J., McKeown N.** Maturing of OpenFlow and Software-Defined Networking Through Deployments. *Computer Networks*. 2014, vol. 61, pp. 151-175.

5. **Korjachko V. P., Perepelkin D. A., Ivanchikova M. A., Byshov V. S., Cyganov I. Ju.** Programmnaja infrastruktura i vizual'naja sreda raspredelennoj obrabotki potokov dannyh v programmno-konfiguriruemyh setjah. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2018, no. 65, pp. 44-54. DOI: 10.21667/1995-4565-2018-65-3-44-54 (in Russian).

6. **Leohin Ju. L., Fathulin T. D.** Ocenka vozmozhnosti predostavlenija garantirovannoj skorosti peredachi dannyh v programmno-konfiguriruemoj opticheskoj seti. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2020, no. 71, pp. 45-59. DOI: 10.21667/1995-4565-2020-71-45-59 (in Russian).

7. **Bastos D. A., Guelfi A. E., Azevedo M. T., Silva A. A.** Formal modelling and analysis of man in the middle prevention control in Software Defined Network. *Revista eletrônica de sistemas de informação*. 2021, vol. 11, pp. 16-40.

8. **Shalimov A. et al.** *Advanced study of SDN/OpenFlow controllers. Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia*. ACM, 2013.

9. **Perepelkin D. A.** Konceptual'nyj podhod dinamicheskogo formirovanija trafika programmno-konfiguriruemyh telekommunikacionnyh setej s balansirovkoj nagruzki. *Informacionnye tehnologii*. 2015, vol. 21, no. 8, pp. 602-610. (in Russian).

10. **Ushakova M. V., Ushakov Ju. A.** Issledovanie seti virtual'noj infrastruktury centra obrabotki dannyh s gibridnoj programmno-konfiguriruemoj kommutaciej. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta.* 2021, no. 75, pp. 34-43. DOI: 10.21667/1995-4565-2021-75-34-43 (in Russian).

11. **Nikul'chev E. V., Pajain S. V., Pluzhnik E. V.** Dinamicheskoe upravlenie trafikom programmno-konfiguriruemyh setej v oblachnoj infrastructure. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2013, no. 3 (45), pp. 54-57. (in Russian).

12. **Yu S., Zhang J., Liu J. et al.** A cooperative DDoS attack detection scheme based on entropy and ensemble learning in SDN. *Wireless Com Network*. 2021. DOI: 10.1186/s13638-021-01957-9.

13. **Nikishin K., Konnov N.** Schedule Time-Triggered Ethernet. *International Conference on Engineering Management of Communication and Technology, EMCTECH* 2020. DOI: 10.1109/EMCTECH 49634.2020.9261540.

14. **Nikishin K.I, Konnov N.N.** Generator trafika Ethernet na osnove cvetnyh setej Petri. *Modeli, sistemy, seti v jekonomike, tehnike, prirode i obshhestve*. 2016, no.1 (17), pp. 299-307. (in Russian).

15. **Nikishin K. I.** Modelirovanie i verifikacija topologij programmno-konfiguriruemyh setej. *Vestnik Rjazanskogo gosudarstvennogo radiotehnicheskogo universiteta.* 2022, no. 80, pp. 67-74. DOI 10.21667/1995-4565-2022-80-67-74 (in Russian).

16. **Nikishin K. I.** Modelirovanie kontrollera i verifikacija processa peredachi dannyh v programmno-konfiguriruemyh setjah. *Vestnik Rjazanskogo gosudarstvennogo radiotehnicheskogo universiteta.* 2022, no. 80, pp. 75-83. DOI 10.21667/1995-4565-2022-80-75-83 (in Russian).

17. **Ameen A.** Assuring the SDN security by modelling and comparing SDN proposed topologies using Petri nets. *Journal of Engineering Science*. 2021, vol. XXVIII, no. 4, pp. 93-105.

18. **Nikishin K. I.** Issledovanie i modelirovanie tablicy potokov kommutatora Openflow v programmno-konfiguriruemyh setjah. *Vestnik Rjazanskogo gosudarstvennogo radiotehnicheskogo universiteta*. 2022, no. 81, pp. 42-50. DOI 10.21667/1995-4565-2022-81-42-50 (in Russian).