УДК 681.3.062

Ю.Г. ЧАШИН, С.А. ЛАЗАРЕВ  
Y.G. CHASHIN, S.A.LAZAREV

**АНАЛИЗ УГРОЗ, МЕТОДОВ И СИТЕМ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ**

**ANALYSIS OF SECURITY THREATS, METHODS AND SYSTEMS IN SOFTWARE-DEFINED NETWORKING**

*В данной статье авторы освещают проблему обеспечения безопасности в программно-конфигурируемых сетях. Анализируются группы угроз безопасности, акцент делается на актуальности поиска скомпрометированных коммутаторов. Рассматриваются возможные атаки с такого коммутатора. Описываются признаки компрометации коммутатора. Исследуются механизмы, используемые в современных системах обнаружения скомпрометированных коммутаторов, анализируются их недостатки.*

*Ключевые слова: безопасность в программно-конфигурируемых сетях, протокол OpenFlow, методы обнаружения скомпрометированных коммутаторов, системы обнаружения скомпрометированных коммутаторов*

In this article, the authors highlight the security issue in software-configurable networks. Security threat groups are analyzed, the emphasis is on the relevance of searching for compromised switches. Possible attacks from such a switch are considered. Describes the signs of compromise of the switch. The mechanisms used in modern systems for detecting compromised switches are investigated, their shortcomings are analyzed.

Keywords: security in software-defined networking, OpenFlow protocol, methods for detecting compromised switches, systems for detecting compromised switches

***Введение***

В условиях сложной эпидемиологической ситуации повышается спрос на различные онлайн сервисы, которые охватывают практически все сферы нашей жизни. Это приводит к существенному повышению нагрузки на крупные центры обработки данных (ЦОД). Одновременно с этим увеличивается и количество атак на такие центры, так за 2021 год их количество, например в Евросоюзе, возросло примерно в 6 раз. ЦОД подверженные успешным атакам могут понести как материальные потери, так и репутационные. Поэтому задача, совершенствования различных методов, направленных как на увеличение производительности, так и на улучшение безопасности является актуальной [1, 2].

**Принципы используемые в программно-конфигурируемых сетях (ПКС*)***

В настоящее время использование ПКС в крупных ЦОД становится необходимой нормой. Концепция построения ПКС позволяет объединять коммутаторы различных производителей, что ведет к снижению стоимости, но и увеличивает возможные уязвимости. В ПКС контур управления отделен от контура передачи данных, что позволяет передать всю логику управления на специализированное программное обеспечение (ПО), находящееся на сервере, или на контроллер ПКС [3]. Использование специализированного ПО несет дополнительные риски, как по уязвимости самого такого ПО, так и по доступу к нему. Однако использование выделенного контура управления позволяет получать глобальное видение топологии, состояние коммутаторов, загрузку каналов передачи и др., что дает возможность более эффективно и централизованно управлять сетью.

Основным элементом ПКС является протокол OpenFllow [4], позволяющий задавать каждому коммутатору правила маршрутизации/модификации, которые хранятся в таблицах потоков коммутаторов. Такие таблицы потоков образуют конвейер обработки пакета, который позволяет последовательно обрабатывать пакеты несколькими правилами маршрутизации. Также имеются выделенная групповая таблица для групповых правил маршрутизации, и опциональные групповые правила маршрутизации.

Заголовки пакетов поступивших на вход коммутатора сравниваются, с шаблонами правил в соответствии с приоритетами, и в случае совпадения выполняется соответствующая инструкция правила. К таким правилам относятся: пересылка пакета на другой коммутатор, сброс пакета, изменение заголовка пакета, переадресация пакета на сервер для анализа и др. Формат правил маршрутизации/модификации содержит, помимо прочих полей, поле счетчика, которое хранит число пакетов, обработанных данным правилом маршрутизации с начала добавления его в коммутатор.

Таким образом сосредоточение администрирования всех сетевых коммутаторов, на одном управляющем сервере или контроллере ПКС, позволяет осуществлять эффективное управление потоками данных путем равномерного распределения нагрузки на коммутаторы и дает возможность оперативно реагировать на изменения в управляемой сети, например такие как: подключение/отключение коммутатора; обнаружение/обрыв линии связи между коммутаторами [5].

**Анализ угроз безопасности в ПКС**

Использование ПКС исключает ряд типичных атак, связанных с исключением из использования рядя классических протоколов маршрутизации, но так же добавляет и новые угрозы безопасности. Такие угрозы безопасности, можно разделить на 4 группы:

1. Канала данных - злоумышленник может, например, как анализировать трафик с целью нахождения персональных данных, так и искажать его.
2. Коммутаторов - контроль над одним коммутатором ПКС может предоставить злоумышленнику широкий спектр возможных атак: на другие коммутаторы, на трафик, на управляющий сервер/контроллер, на канал управления.
3. Канала управления - злоумышленник может нарушать взаимодействие управляющего сервера/контроллера с коммутаторами. Для предотвращения этого имеется встроенный протокол защиты управляющего трафика TLS, который так же может иметь уязвимости.
4. Управляющего сервера/контроллера – злоумышленник, получив доступ к серверу, может тем самым получить и полный доступ у управляющему ПО ПКС. Также на рынке представлено большое количество контроллеров ПКС, которые могут иметь свои уязвимости.

Анализ и практика показывает, что наибольшей угрозой является наличие в сети ПКС скомпрометированного коммутатора, т.е. коммутатора который находится под контролем злоумышленника. Процесс компрометации хорошо изучен, и имеются эффективные методы противодействия этому. Поэтому далее остановимся на угрозе наличия скомпрометированного коммутатора в сети ПКС.

Коммутатор в сети, находящийся под контролем злоумышленников, может осуществлять следующие, наиболее вероятные атаки [6, 7]:

1. Атаки на управляющий сервер/контроллер:

* все пакеты скомпрометированным коммутатором перенаправляются на управляющий сервер/контроллер, якобы для анализа, тем самым перегружая его;
* скомпрометированный коммутатор может передавать на управляющий сервер/контроллер ложную информацию о физическом состоянии сети, например об отсутствии существующей физической линии, либо о наличии несуществующей физической линии, тем самым искажая представление о топологии сети;
* если в управляющей сети не используется, протокол защиты TLS, то у злоумышленников появляется возможность создавать фиктивные коммутаторы, тем самым влияя на процесс выбора маршрутов для потоков в сети.

2. Атаки на передаваемые пакеты/данные [8, 9]:

* перенаправление копий всех пакетов на контролируемый злоумышленниками хост, с целью получения из них конфиденциальной информации;
* перенаправление пакетов по ложным маршрутам, с целью создать перегрузку на некотором участке сети. В этом случае может произойти переполнение входного буфера коммутатора, и пакеты для анализа будут целиком переправляться управляющему серверу/контроллеру;
* сброс пакетов;
* изменение заголовков пакетов, с целью их последующего сброса другими коммутаторами, тем самым компрометируя их;
* искажение показания счетчиков правил маршрутизации, с целью сокрытия фактов атак;
* переполнение таблицы потоков;
* задержка, модификация содержимого пакетов, нарушение их порядка следования.

Таким образом, исходя их представленного широкого диапазона возможных атак, требуются эффективные методы выявления таких скомпрометированных коммутаторов.

К явным признакам компрометации коммутатора можно отнести:

1. Не прохождение коммутатором процесса аутентификации устройства. Однако у злоумышленников имеется доступ к криптографическим ключам, и пройдя процедуру аутентификации, коммутатор будет считаться легитимным.
2. Наличие на коммутаторе правил маршрутизации, которые не установлены сервером/контроллером. Однако злоумышленники могут создавать поддельные таблицы правил маршрутизации, данные из которых могут быть переданы по запросу на управляющий сервер/контроллер, тем самым скрывая факт вмешательства.
3. Несоответствие значений счетчиков правил маршрутизации на коммутаторе с расчетными значениями. Сервер/контроллер может запросить значения счетчиков и проверить принцип сохранения потока. Но злоумышленники, так же могут подделывать значения счетчиков.

Как видно, использование методов основанных на выявлении простых признаках компрометации недостаточно.

**Анализ механизмов, используемых в современных системах обнаружения скомпрометированных коммутаторов**

Рассмотрим механизмы используемые в существующих, системах обнаружения скомпрометированных коммутаторов. Работа таких систем основана на наличии у сервера/контроллера полной информация о топологии сети, свойствах пакетов проходящих через сеть, и о правилах обработки сетевого трафика, установленных на коммутаторах. К таким механизмам можно отнести [10]:

* выявление некорректных выполнений существующих правил маршрутизации. Производится тестирование всех правил, установленных на всех коммутаторах, с помощью тестовых пакетов. Такой механизм, например используют в системе обнаружения ATPG [11];
* проверка маршрутов пакетов. Основана на сравнении теоретических и реальных маршрутов. Используется, например в PDMD [12];
* проверка корректности, целостность и неизменчивости передачи пакетов между входным и выходным коммутаторами сети. Такие пакеты пересылаются и сравниваются на сервере/контроллере. Используется в DYNAPFV [13];
* проверка работы новых правил маршрутизации. В коммутаторы добавляются новые правила, и через них пропускаются пакеты, которые обрабатываются этими правилами. Отслеживаются и проверяются пути этих пакетов. Механизм используется в FDWD [14];
* анализ и проверка согласованности сетевой статистики, путем проверки значений счетчиков правил маршрутизации коммутаторов. Анализируются только пути, покрывающие все вершины, для которых добавляются дополнительные правила маршрутизации, и анализируется только статистика этих правил. Используется в SPHINX [15].

Многие вышеописанные системы обнаружения скомпрометированных коммутаторов с успехом функционируют в сетях. Однако и у них встречаются существенные недостатки, например такие как:

* дополнительная нагрузка на сеть, вызванная действиями самих систем обнаружения, которая может приводить к перегрузкам в сети;
* не предусмотрен механизм анализа перегрузок в сети, которые могут встречаться довольно часто, в следствии чего сбросы пакетов связанные с перегрузкой, могут быть приняты за вредоносные сбросы;
* не возможность обнаружения кратковременных атак;
* влияние установки системами обнаружения дополнительных правил маршрутизации, на сам процесс обнаружения;
* стандартные изменения в топологии сети могут повлиять на процесс обнаружения атак;
* не возможность обнаружения сложных атак, производимых с использованием нескольких скомпрометированных коммутаторов;
* не учитываются многие известные способы сокрытия атак;
* проблемы с учетом легитимных сбросов пакетов;
* не всегда возможно точно установить скомпрометированные коммутаторы, например если легитимный окружен скомпрометированными; и др.

**Разработка системы обнаружения скомпрометированных коммутаторов**

Из выше приведенных данных, очевидна потребность в создании новых эффективных систем обнаружения скомпрометированных коммутаторов, так как в ряде случаев, существующие системы обнаружения не всегда являются эффективными.

На наш взгляд к таким системам должны предъявляться следующие, основные требования:

* высокая вероятность выявления наличия вредоносных действий в ПКС, среди которых обязательно должны быть:
  + обнаружение утечки данных на хосты злоумышленников;
  + выявление как постоянных, так и краткосрочных воздействий злоумышленников;
* обнаружение скомпрометированных коммутаторов;
* контроль перегрузок в сети, учитывающий и контролирующий дополнительную нагрузку на ПКС, оказываемую самой системы обнаружения.

Таким образом система обнаружения скомпрометированных коммутаторов, должна использовать следующие механизмы, основанные на добавлении новых правил маршрутизации, и анализе данных на управляющем сервере/контроллере:

* анализа изменений в топологии ПКС. Данный механизм отслеживает подключение/отключение как коммутаторов, так и связанных с ними физических линий. Использование его должно позволить обнаружить скомпрометированный коммутатор, который передает на управляющий сервер/контроллер ложную информацию о физическом состоянии сети, либо обнаружить проблемы с исправностью коммутатора;
* отслеживания перегрузок в ПКС, и в случае не превышения некоторого порога, задействуется механизм проверки корректности, неизменчивости, полноты и эффективности передачи пакетов. Данный механизм подразумевает пересылку всех входящих пакетов в ПКС, и исходящих пакетов из ПКС, на сервер/контроллер для анализа. Там производится сравнение реальных маршрутов прохождения пакетов с теоретическими, проверяется их целостность, при этом учитываются легитимные сбросы и анализируются задержки. Этот механизм должен позволить выявлять большой спектр атак, в том числе вредоносные задержки, краткосрочные атаки и утечку данных;
* в случае наличия перегрузок в ПКС, превышающих некоторый порог, или в случае выявления предыдущим механизмом наличия атак в сети, задействуется механизм проверки правил маршрутизации, последовательно проверяющий их на каждом коммутаторе ПКС. Для этого, в течении некоторого времени, пересылаются входящие и исходящие пакеты с такого коммутатора на сервер/контроллер, где происходит их анализ, при этом используются эталонные таблицы маршрутизации, хранящиеся на сервере/контроллере. Анализируются так же и задержки прохождения пакетов. Этот механизм предназначен для выявления скомпрометированных коммутаторов.

**Вывод**

На основе проведённого в статье анализа делается вывод о необходимости разработки новых, эффективных систем обнаружения скомпрометированных коммутаторов. Формируются требования, предъявляемые к таким системам. Отмечается, что акцент надо сделать на высокой вероятности выявления наличия вредоносных действий в ПКС, а именно: на обнаружении утечек данных, и на выявлении как постоянных, так и краткосрочных вредоносных воздействиях злоумышленников. Подчеркивается, что при этом надо учитывать и контролировать перегрузки в ПКС.

Для постарения эффективной системы обнаружения скомпрометированных коммутаторов предлагается использовать три механизма. Первый механизм анализирует изменение в топологии ПКС и выявляет скомпрометированный коммутатор, который передает на управляющий сервер/контроллер ложную информацию о физическом состоянии сети. Второй отслеживает перегрузки в ПКС, и в случае их отсутствия проверяет корректность передачи пакетов. Третий выявляет скомпрометированный коммутатор путем анализа его входящих и исходящих пакетов на сервере/контроллере.

В статье дается обоснование эффективности такой системы обнаружения скомпрометированных коммутаторов.

**СПИСОК ЛИТЕРАТУРЫ**

1. Igor Sergeevich Konstantinov, Jurij Gennadievich Chashin and Sergej Aleksandrovich Lazarev, 2014. Simulation of the Software-Defined Network for a High-Performance Computing Cluster. Research Journal of Applied Sciences, 9: 704-706. <http://medwelljournals.com/abstract/?doi=rjasci.2014.704.706>
2. Jianmei Guo, Dingyu Yang, Norbert Siegmund, Sven Apel, Atrisha Sarkar, Pavel Valov, Krzysztof Czarnecki, Andrzej Wasowski, and Huiqun Yu. 2017. Data-efficient performance learning for configurable systems. Empirical Software Engineering (2017), 1–42.
3. Nunes, B., Mendonca, M., Nguyen, X., Obraczka, K., Turletti, T.: A survey of Software-Deﬁned Networking: past, present, and future of programmable networks. IEEE Commun. Surv. Tut. PP(99), 1–18
4. Foundation O. N. OpenFlow Switch Specification v1.5.1. — 2015. <https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>
5. Смелянский Р.Л. SDN&NFV - новые горизонты. 2015. [lvk.cs.msu.ru/~sveta/Статья\_для\_CNEWS\_2015\_ЦПИКС\_Смелянский.pdf](http://lvk.cs.msu.ru/~sveta/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F_%D0%B4%D0%BB%D1%8F_CNEWS_2015_%D0%A6%D0%9F%D0%98%D0%9A%D0%A1_%D0%A1%D0%BC%D0%B5%D0%BB%D1%8F%D0%BD%D1%81%D0%BA%D0%B8%D0%B9.pdf)
6. Towards fine-grained network security forensics and diagnosis in the sdn era / Haopei Wang, Guangliang Yang, Phakpoom Chinprutthiwong et al. // Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security / ACM. — 2018. — P. 3–16.
7. Pang C., Jiang Y., Li Q. Fade: Detecting forwarding anomaly in software-defined networks // Communications (ICC), 2016 IEEE International Conference on / IEEE. — 2016. — P. 1–6.
8. Securing data planes in software-defined networks / Tzu-Wei Chao, Yu- Ming Ke, Bo-Han Chen et al. // NetSoft Conference and Workshops (Net-Soft), 2016 IEEE / IEEE. — 2016. — P. 465–470.
9. Reins to the cloud: Compromising cloud systems via the data plane / Kashyap Thimmaraju, Bhargava Shastry, Tobias Fiebig et al. // arXiv preprint arXiv:1610.08717. — 2016.
10. Петров И.С. Системы обнаружения скомпроментированных коммутаторов в программно-конфигурируемых сетях. Журнал Информационные технологии, издательство Новые технологии, Москва, 2019. Том 25, №3, с.131-142.
11. Automatic test packet generation / Hongyi Zeng, Peyman Kazemian, George Varghese, Nick McKeown // Proceedings of the 8th interna- tional conference on Emerging networking experiments and technologies / ACM. — 2012. — P. 241–252.
12. Mohammadi A. A., Kazemian P., Pakravan M. R. Detecting malicious packet drops and misroutings using header space analysis // Telecommu- 142 nications (IST), 2016 8th International Symposium on / IEEE. — 2016. — P. 521–526.
13. Dynamic packet forwarding verification in sdn / Qi Li, Xiaoyue Zou, Qun Huang et al. // IEEE Transactions on Dependable and Secure Computing. — 2018.
14. How to detect a compromised sdn switch / Po-Wen Chi, Chien-Ting Kuo, Jing-Wei Guo, Chin-Laung Lei // Network Softwarization (NetSoft), 2015 1st IEEE Conference on / IEEE. — 2015. — P. 1–6.
15. Sphinx: Detecting security attacks in software-defined networks. / Mohan Dhawan, Rishabh Poddar, Kshiteej Mahajan, Vijay Mann // NDSS. — Vol. 15. — 2015. — P. 8–11.

**Чашин Юрий Геннадиевич**

Белгородский государственный университет, г. БелгородК.т.н., доцент кафедры «Математического и программного обеспечения информационных систем»  
Тел.: +7(4722) 30-12-11  
E-mail: chashin@bsu.edu.ru

**Лазарев Сергей Александрович**Белгородский государственный университет, г. БелгородК.э.н., заместитель директора по научной и международной деятельности института «Инженерных и цифровых технологий»

Тел.: +7(4722) 30-12-11  
E-mail: lazarev\_s@bsu.edu.ru