ИТМО

Безопасность в SDN

Фоминцев Денис К34212

ВВЕДЕНИЕ

SDN (Software-Defined Networking, программно-определяемые сети) — это подход к сетевой архитектуре, в котором управление сетью отделено от передачи данных, и осуществляется программно через централизованный контроллер. Это позволяет администраторам динамически управлять сетью, автоматически настраивать маршрутизацию, балансировать нагрузку и быстро адаптировать инфраструктуру под изменяющиеся требования.

SDN обеспечивает гибкую, адаптивную и экономичную архитектуру, способную эффективно адаптироваться под передачу больших потоков трафика.

Основные компоненты

Уровень управления (Control Plane)

Основная задача Control Plane – обеспечить условия того, чтобы все компоненты системы выполняли свои функции эффективно и безопасно. Он содержит логику управления сетью, обычно реализуемую в централизованном контроллере.

Отвечает за создание локального набора данных, который используется для формирования записей в таблицах пересылки. Эти записи затем используются плоскостью передачи данных для маршрутизации трафика.

Основные компоненты

Уровень передачи данных (Data Plane)

Отвечает за фактическую передачу данных между сетевыми устройствами. Плоскость передачи данных выполняет обработку входящих датаграмм, проходящих через ряд операций на канальном уровне. В этот процесс входят начальные проверки на целостность и корректность данных. Если датаграмма соответствует установленным требованиям, она передается в дальнейшую обработку.

На этом этапе плоскость передачи данных использует таблицы пересылки (FIB), заранее сформированные контрольной плоскостью.

Четыре основных принципа SDN

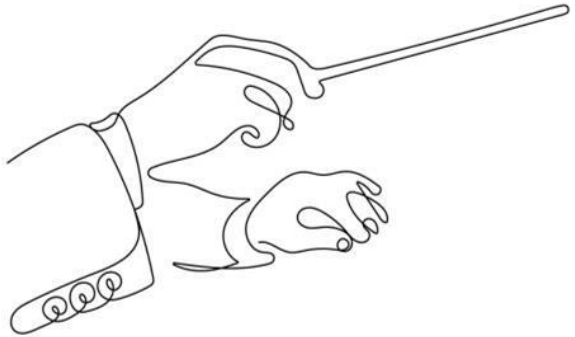
1. Разделение плоскостей управления и передачи данных.

Этот принцип подразумевает, что управление сетью отделяется от передачи данных, что позволяет централизованно управлять политиками и правилами маршрутизации, в то время как устройства сети занимаются только передачей данных.

Четыре основных принципа SDN

2. Логически централизованное управление.

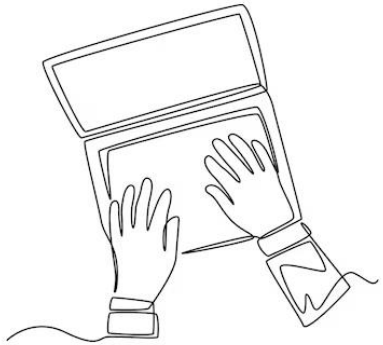
В этой модели контроллеры могут находиться в разных местах, но работают как единое целое, предоставляя централизованный интерфейс для управления. Это позволяет использовать преимущества централизованного управления, сохраняя при этом распределенные ресурсы для повышения надежности и масштабируемости.



Четыре основных принципа SDN

3. Программируемость сети.

SDN предоставляет возможность программирования сети через открытые интерфейсы и API, что позволяет разработчикам создавать и внедрять новые сетевые услуги и приложения без необходимости в сложных конфигурациях оборудования.



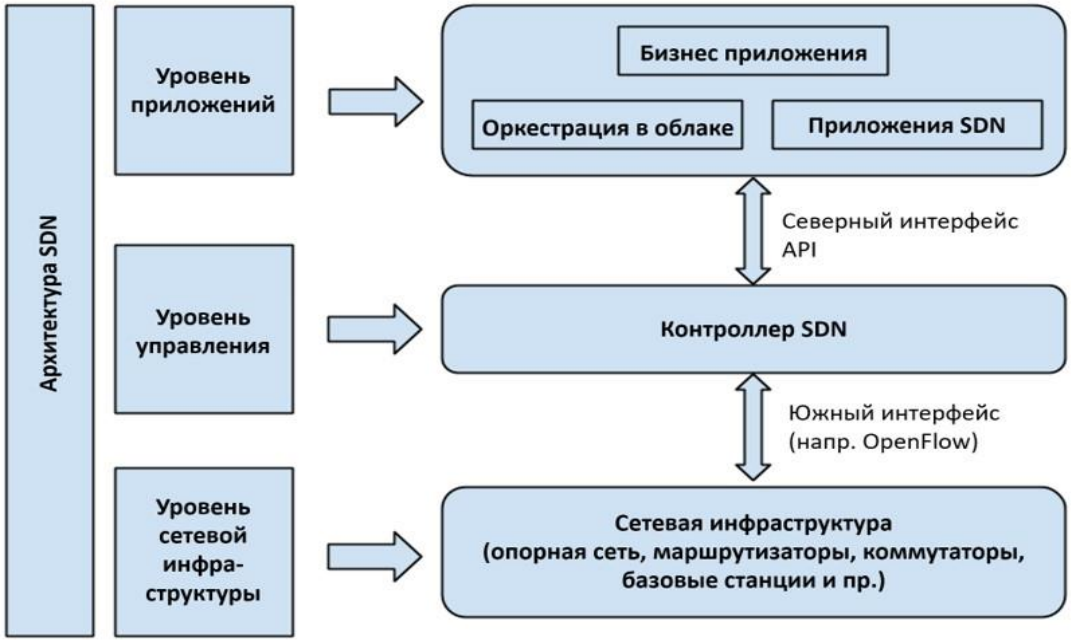
Четыре основных принципа SDN

4. Виртуализация.

Этот принцип позволяет создавать логические сети поверх физической инфраструктуры, что обеспечивает большую гибкость в управлении ресурсами и возможность быстрого масштабирования сети в ответ на изменяющиеся требования бизнеса.

virtualization

Архитектура SDN



Интерфейсы

Южный интерфейс связывает SDN-контроллер с сетевыми устройствами, передавая команды управления и получая данные о состоянии сети.
Для этого могут использоваться протоколы, такие как **OpenFlow**, **NETCONF**, **gNMI**, **BGP-LS**, **P4 Runtime**, **OF-CONFIG**, обеспечивающие гибкое управление маршрутизацией, конфигурацией и мониторингом сети.

Северный интерфейс соединяет SDN-контроллер с внешними приложениями и сервисами, предоставляя API для автоматизации и мониторинга сети.
Основными технологиями взаимодействия являются **REST API**, **gRPC**, **NETCONF/YANG** и **GraphQL**, позволяющие динамически управлять конфигурацией сети и интегрировать SDN с аналитическими системами.

Атаки на контроллер (Controller Attacks)

- DDoS-атаки (перегрузка запросами, отказ в обслуживании).
- Эксплойты уязвимостей (атаки на API, уязвимости в ПО контроллера).
- Перехват управления (злоумышленник получает контроль над контроллером).

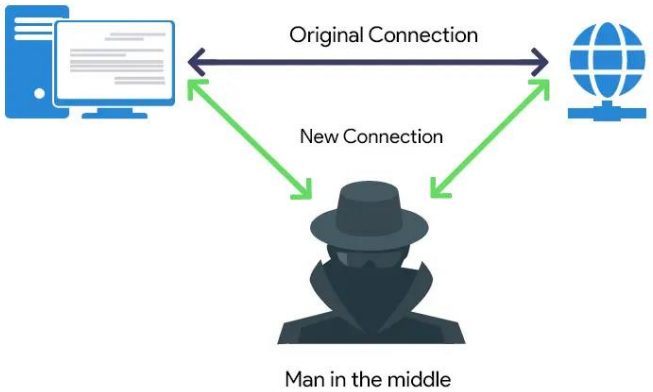
Защита контроллера

- Использование отказоустойчивых архитектур (кластеризация, резервные копии).
- Ограничение доступа к API с аутентификацией (OAuth, JWT, TLS).
- Мониторинг аномалий и система обнаружения вторжений (IDS/IPS).

Атаки на южный интерфейс (Southbound Interface Attacks)

Воздействие на связь между контроллером и сетевыми устройствами

- Инъекции вредоносных потоков (Flow Rule Injection) – изменение правил в базе информации о пересылке (FIB) для перехвата трафика.
- Человек посередине (MITM) – подмена контроллера или сетевых устройств.



Защита южного интерфейса

- Шифрование коммуникаций (TLS, IPsec).
- Подписанные правила потоков для валидации изменений.
- Ограничение команд на сетевых устройствах (White-listing команд).

Атаки на северный интерфейс (Northbound Interface Attacks)

Компрометация API взаимодействия с внешними сервисами

- SQL/XSS-инъекции в REST API.
- Неавторизованный доступ через слабые механизмы аутентификации.

Защита северного интерфейса

- Внедрение различных моделей разделения доступа, например, ролевые RBAC (Role-Based Access Control), мандатные и дискреционные, а также комбинации данных моделей с учетом специфики защищаемой инфраструктуры и аутентификации.
- Анализ API-трафика на предмет аномального поведения.

Атаки на инфраструктуру передачи данных (Data Plane Attacks)

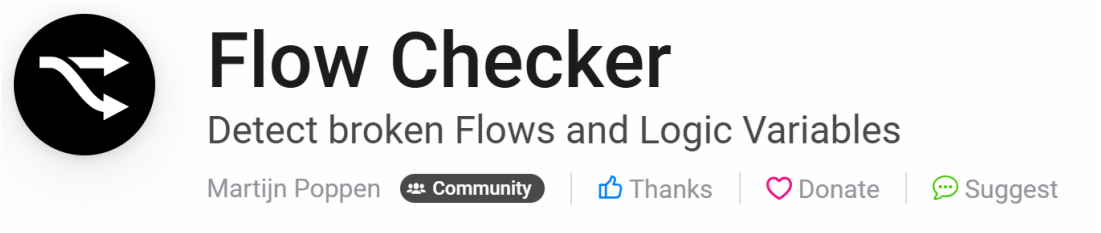
Эксплуатация уязвимостей сетевых устройств

- Подмена ARP/ND (Spoofing) – перенаправление трафика злоумышленнику.
- Переполнение таблиц потоков (Flow Table Overflow) – исчерпание ресурсов коммутаторов.

Защита плоскости передачи данных

- Ограничение количества потоков на коммутаторах.
- Использование фильтрации пакетов (ACL, DPI).
- Механизмы защиты от ARP/ND спуфинга (DAI, SAVI).

Инструменты-помощники. FlowChecker



FlowChecker — это инструмент, который используется для проверки и валидации потоков в сети.
Архитектура SDN позволяет динамически управлять потоками данных через установку правил маршрутизации и обработки трафика на сетевых устройствах. FlowChecker использует бинарные диаграммы решений для анализа и верификации конфигурации сети. Он разделяет сетевые ресурсы на слои, что позволяет эффективно выявлять конфликты и несоответствия в правилах потоков.

Инструменты-помощники. VeriFlow

Как и FlowChecker, Veriflow работает с контроллером SDN. Контроллер управляет всей сетью и устанавливает правила для маршрутизации трафика. Veriflow помогает контроллеру проверять, что эти потоки не противоречат друг другу, а также соответствуют заданным политикам безопасности и правилам маршрутизации. **Veriflow выполняет формальную верификацию потоков в режиме реального времени.**

Заключение

SDN дает огромные возможности для гибкого и эффективного управления сетью, как и любая технология, она требует серьезного подхода к безопасности. Комплексная защита контроллера, интерфейсов и инфраструктуры помогает минимизировать риски и повысить устойчивость сети к различным угрозам.

Спасибо
за внимание!

ITMO more than a UNIVERSITY