АНАЛІЗ SDN-СПЕЦИФІЧНИХ МЕТОДІВ СТВОРЕННЯ ПРИХОВАНИХ КАНАЛІВ

Фокін Д.Г., Євдокименко М.О.

Кафедра інфокомунікаційної інженерії ім. В.В. Поповського,
Харківський національний університет радіоелектроніки,
Україна

E-mail: denys.fokin@nure.ua
marina.ievdokymenko@nure.ua

Abstract

Software-Defined Networking introduces significant flexibility but also novel security threats, notably covert channels that exploit its core mechanisms. This work provides an overview and systematization of existing SDN-specific steganography methods. We analyze five distinct techniques, including unprivileged host-to-host timing channels, privileged controller-to-host channels, and switch-to-switch channels using control-plane teleportation. A preliminary comparison, based on data from the original papers, reveals significant disparities in throughput and reliability. However, these methods were evaluated on disparate testbeds under different conditions, making a direct assessment of their relative threat impossible. This analysis substantiates the need for a unified comparative study, implementing and evaluating all methods on a single, standardized testbed to objectively measure their practical effectiveness and stealth.

Програмно-конфігуровані мережі (Software-Defined Networks, SDN) завдяки своїй централізації та гнучкості програмування здобули широке розповсюдження і активно використовуються у хмарних мережах, центрах обробки даних та у мережах великих корпорацій, таких як Microsoft, Google та Amazon. Однак ці ж самі нові функції, що надають переваги, одночасно створюють і нові вектори загроз. Централізований контролер стає критичною точкою відмови та привабливою ціллю для атак. Однією з таких значних, але малодосліджених загроз є мережна стеганографія, або приховані канали. Такі канали в SDN можуть використовувати специфічні механізми протоколів, наприклад, OpenFlow, для прихованої передачі інформації, оминаючи традиційні засоби моніторингу та безпеки, такі як брандмауери та системи виявлення вторгнень.

Головна загроза полягає у можливості непомітного витоку критично важливих даних з мережі: це може включати приватні ключі TLS та RSA, конфіденційні метадані, інтелектуальну власність або системну інформацію, таку як топологія мережі та чинні політики безпеки. Окрім витоку даних, приховані канали можуть використовуватися для координації атак, обходу фізичної чи логічної ізоляції мережі, та у випадку компрометації ключів — для атаки з видаванням себе за легітимний контролер. Таким чином зловмисники можуть не лише викрадати інформацію, але й готувати плацдарм для більш потужних та прихованих атак. Актуальність дослідження цих каналів полягає у необхідності розуміння повного ландшафту загроз, які виникають із впровадженням SDN.

У роботі [1] пропонується метод створення прихованого каналу між двома хостами, що знаходяться в одній SDN-мережі. Тут експлуатується обмежена обчислювальна потужність SDN контролера та використання ним механізмів проксі-запитів, зокрема ARP Proxy. Оскільки всі проксі-запити (SDN Proxy Request, SPR) вимагають централізованої обробки на контролері, він стає вузьким місцем, яке можна використовувати для модуляції затримки. Передумовою для атаки є наявність двох скомпрометованих хостів у мережі, які виступають як відправник та отримувач.

Послідовність передачі одного біта повідомлення виглядає наступним чином. Хост-відправник генерує короткий, але інтенсивний сплеск SPR-пакетів, наприклад, ARP-запитів, зі швидкістю, що тимчасово перевантажує контролер. Це створює чергу і спричиняє значне збільшення часу затримки (RTT) для всіх проксі-запитів у мережі. Для передачі біта іншого значення хост-відправник утримується від дій. Хост-отримувач, у свою чергу, постійно надсилає власні SPR-пакети та вимірює їх RTT. Якщо він виявляє затримку, що перевищує визначений поріг, він реєструє одне значення, в

іншому випадку – інше. Автори протестували метод на тестовому стенді з апаратними коммутатором EdgeCore AS4610-54T та контролером RYU.

Головною перевагою є висока скритність: при швидкостях до 5 біт/с навантаження на CPU контролера та кількість control-plane пакетів майже не відрізнялися від фонових показників. Іншою перевагою є те, що він не вимагає жодних привілеїв чи компрометації контролера або комутаторів. Недоліком є низька пропускна здатність, обмежена необхідністю "охолодження" каналу, яка в тестах не перевищувала 20 біт/с.

У статті [2] пропонується метод, який також реалізується між двома хостами, що можуть бути логічно ізольовані, наприклад, у різних VLAN, але керовані одним SDN контролером. Ключовий механізм, що експлуатується – це стандартна функція контролера MAC learning (або mobility application), яка відповідає за оновлення правил потоків при переміщенні хоста. Передумовою, як і в попередньому методі, є наявність двох скомпрометованих хостів, відправник та отримувач, які мають привілейований доступ до свого мережного стека для можливості підміни MAC-адреса.

Послідовність передачі одного біта повідомлення наступна. Хост-відправник надсилає будь-який пакет, вказуючи у якості MAC-адреси джерела MAC-адресу хоста-отримувача. Комутатор відправника, отримавши пакет з відомою MAC-адресою на новому порту, надсилає повідомлення контролеру. Контролер, вважаючи, що хост-отримувач перемістився, запускає процес реконфігурації потоку: він видаляє старі правила потоку, пов'язані з цією MAC-адресою на комутаторі отримувача, та готується встановити нові. У цей короткий момент легітимний трафік хоста-отримувача зазнає вимірюваної затримки RTT, оскільки старі правила видалені. Для передачі біта іншого значення хост-відправник не робить нічого, і хост-отримувач не фіксує аномалій RTT. Автори протестували свій метод, використовуючи Mininet для симуляції топології з 20 комутаторів та хостів, та окремий сервер для контролера. Тести проводились з OpenFlow та з P4Runtime.

Перевагами методу є його незалежність від конкретного SDN-протоколу (OpenFlow, P4Runtime тощо), оскільки експлуатується загальна логіка роботи контролера, та відсутність необхідності компрометації інфраструктури. Як і попередній метод, він не вимагає компрометації контролера або комутаторів. Недоліком є відносно низька пропускна здатність порівняно з іншими методами, а також чутливість до загального навантаження на контролер, яке може збільшувати рівень помилок.

Дослідження [3] демонструє канал, призначений для витоку інформації від контролера до хоста поза мережею. Відправником у цьому методі виступає шкідливий додаток, встановлений безпосередньо на SDN-контролері. Отримувачем є будь-який зовнішній хост, який має легітимний доступ до публічного сервісу, наприклад, веб-сервера, що знаходиться всередині цільової SDN-мережі. Канал експлуатує механізм закінчення терміну дії правил потоку (rule expiry). Передумовою для атаки є компрометація control-plane, а саме встановлення шкідливого додатку на контролер, який має повноваження керувати правилами потоків.

Для передачі одного біта даних автори використовують Манчестерське кодування. Хост-отримувач ініціює зв'язок, відправляючи два пакети до внутрішнього сервера з інтервалом, достатньо великим, щоб правило потоку встигло спливти, наприклад, >1 секунди. Кожен з цих пакетів генерує packet-in запит до контролера на встановлення нового правила. Якщо код, що потрібно передати це '10', для передачі '1', контролер навмисно затримує відправку flow-mod повідомлення для першого пакета, створюючи високий RTT, і відправляє flow-mod негайно для другого створюючи низький RTT. Для передачі '0' (код '01') дії виконуються навпаки. Отримувач порівнює RTT цих двох послідовних запитів і декодує біт. Автори протестували метод на стенді з апаратними комутаторами EdgeCore AS4610-54T, контролером Floodlight та внутрішнім веб-сервером Apache. У ролі отримувачів виступали 6 хостів, орендованих у різних географічних локаціях (США, Нідерланди, Китай).

Ключовою перевагою є унікальна здатність виводити дані за межі мережі, навіть з фізично ізольованим control-plane, будь-якому хосту в Інтернеті, що робить його ідеальним для витоку даних. Однак метод має два суттєві недоліки: по-перше, він вимагає компрометації самого контролера, і по-друге, він має надзвичайно низьку пропускну здатність, теоретично обмежену 0.5 біт/с, через мінімальний час життя правила в 1 секунду та 2 пакети на біт), що робить його придатним лише для передачі дуже малих об'ємів даних, як-от ключів шифрування.

Крім того, в [3] описується єдиний запропонований SDN-специфічний storage-based канал, що працює від контролера до внутрішнього хоста. Цей метод експлуатує SDN proxy механізми, наприклад, ARP Proxy, де контролер зобов'язаний генерувати та надсилати відповіді на запити хостів. Передумовою, аналогічно до попереднього методу, є наявність шкідливого додатку-відправника на контролері. Отримувачем виступає хост-спільник всередині SDN-мережі.

Послідовність передачі даних наступна. Хост-отримувач надсилає звичайний проксі-запит, наприклад, ARP-запит. Відправник перехоплює цей запит, генерує відповідь, але перед її відправкою вбудовує біти прихованого повідомлення у невикористовувані або зарезервовані поля пакета. Наприклад, автори використали для цього 144 біти поля padding в ARP-відповіді. Після модифікації пакет надсилається хосту-отримувачу, який зчитує звичайну відповідь, а також вилучує приховані дані з поля padding. Тестування проводилося на тому ж стенді (EdgeCore, Floodlight).

Головною перевагою методу є висока, у порівнянні з іншими, пропускна здатність та ідеальна точність передачі. До недоліків відноситься компрометація контролера, а також відносно низька скритність: модифікація зарезервованих полів, є прямим порушенням протоколу, яке легко може бути виявлено системою виявлення вторгнень (IDS), що виконує глибоку інспекцію пакетів.

Робота [4] пропонує прихований канал, що функціонує між двома комутаторами, використовуючи для цього control-plane. Канал базується на концепції "SDN Teleportation" та експлуатує OpenFlow handshake. Зокрема, використовується той факт, що контролер очікує унікальний ідентифікатор Datapath ID (DPID) від кожного комутатора і примусово розриває з'єднання з другим пристроєм, що намагається підключитися з DPID, який вже використовується. Передумовою для атаки є компрометація двох комутаторів (відправника та отримувача), які, однак, можуть бути повністю ізольовані один від одного у data-plane, наприклад, знаходитись у різних фізичних мережах чи бути розділеними брандмауером. Єдиною вимогою є те, що обидва комутатори мають доступ до одного й того ж контролера, причому сам контролер залишається нескомпрометованим.

Послідовність передачі біта повідомлення відбувається у заздалегідь узгоджений часовий інтервал. Комутатор-відправник підключається до контролера, використовуючи узгоджений DPID, і підтримує це з'єднання; для передачі іншого значення відправник не підключається. Комутатор-отримувач, через короткий проміжок часу, завжди намагається підключитися до контролера, використовуючи той самий узгоджений DPID. Якщо його з'єднання миттєво розривається контролером (TCP FIN), отримувач фіксує це як отриманий біт. Якщо ж з'єднання встановлюється успішно, отримувач фіксує інше значення. Автори протестували прототип каналу на базі комутаторів Open vSwitch та контролера ONOS.

Основною перевагою методу є його здатність повністю обходити будь-які засоби безпеки у control-plane, оскільки комунікація відбувається виключно через control-plane. Недоліком є необхідність компрометації двох комутаторів мереженої інфраструктури, що є значно складнішим завданням, ніж компрометація хостів. Також недоліком є низька точність передачі в порівнянні з іншими розглянутими методами.

У таблиці 1 наведено зведені характеристики та результати експериментальних досліджень для рогзлянутих вище методів. Важливо зазначити, що всі наведені дані щодо пропускної здатності та рівня помилок взяті безпосередньо з оригінальних наукових праць [1] – [4], в яких ці методи були вперше запропоновані та протестовані.

Таблиця 1. SDN-специфічні приховані канали

Метод	[1] ARP Burst	[2] Macchiato	[3] SDN Rule Expiry	[4] SDN Teleportation	[3] ARP Proxy
Тип каналу	Timing	Timing	Timing	Timing	Storage
Механізм що експлуатується	SDN Proxy - proxy request processing overload	MAC learning	Rule Expiry	OpenFlow Handshake	SDN Proxy - packet-in / packet-out
Відправник	Internal Host 1	Internal Host 1	Controller	Switch 1	Controller
Отримувач	Internal Host 2	Internal Host 2	External Host	Switch 2	Internal Host
Пропускна здатність	<20 біт/с	~5.9 біт/с	~0.5 біт/с	~20 біт/с	~200 біт/с
Рівень помилок	~1.5%	~0.4%	~0%	~10%	~0%

Початкове порівняння, виявляє значні розбіжності у характеристиках каналів. Однак, ці дані отримані авторами оригінальних робіт [1] – [4] на абсолютно різних програмних та апаратних тестових стендах (Mininet, OvS, EdgeCore) та з різними контролерами (RYU, ONOS, Floodlight). Це унеможливлює об'єктивне порівняння методів між собою та адекватну оцінку їх реальної загрози або користі для розвитку використання прихованих каналів для захисту важливої control-plane інформації як це продемонстровано у [5].

Таким чином обґрунтовується необхідність подальшого дослідження, що є метою майбутньої наукової роботи. Планується реалізація всіх розглянутих методів на єдиному уніфікованому тестовому стенді. Це дозволить провести комплексну оцінку ефективності та прихованості кожного каналу в однакових, контрольованих умовах — при однаковій топології, контролері, рівні фонового навантаження.

Література

1. Y. Ji, J. Cao, Q. Li, Y. Liu, T. Wei, K. Xu, and J. Wu, “Constructing SDN Covert Timing Channels Between Hosts With Unprivileged Attackers,” IEEE Transactions on Networking, vol. 33, no. 2, pp. 612-626, Apr. 2025, doi: 10.1109/TNET.2024.3496997

2. A. Sabzi, L. Schiff, K. Thimmaraju, A. Blenk, and S. Schmid, “Macchiato: Importing Cache Side Channels to SDNs,” in Proc. Symp. on Architectures for Networking and Communications Systems (ANCS ’21), Lafayette, IN, USA, Dec. 13–16, 2021, pp. 1-7, doi: 10.1145/3493425.3502758

3. J. Cao, K. Sun, Q. Li, M. Xu, Z. Yang, K. J. Kwak, and J. Li, “Covert Channels in SDN: Leaking Out Information from Controllers to End Hosts,” in Proc. SecureComm 2019, LNICST vol. 304, S. Chen et al., Eds., Cham, Switzerland: Springer, 2019, pp. 429-449, doi: 10.1007/978-3-030-37228-6_21

4. R. Krösche, K. Thimmaraju, L. Schiff, and S. Schmid, “I DPID It My Way! A Covert Timing Channel in Software-Defined Networks,” in Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS 2018), 2018, pp. 218-225, ISBN 978-3-903176-08-9

5. Y. Hu, X. Li, and X. Mountrouidou, “Improving Covert Storage Channel Analysis with SDN and Experimentation on GENI,” in Proc. National Cyber Summit (NCS ’16), Huntsville, AL, USA, Jun. 7-9, 2016.