A Framework for Policy Enforcement Management in Database-Defined Networking (DDN)

Ali Al-Haj and Benjamin Aziz
School of Computing, University of Portsmouth
Portsmouth PO1 3HE, United Kingdom
{ali.alhaj, benjamin.aziz}@port.ac.uk

***Abstract*—Software Defined Networking (SDN) make it easier for administrators to manage security policies on a network system. At the same time, it is still challenging to map high-level security policies defined by users into low-level security policies that can be enforced into network devices. Database-Defined Networking (DDN) is a new concept which use relational databases as an abstraction for managing an SDN. DDNs simplify the network management since the interface to its current state becomes purely database defined. We aiming to introduce a framework for policy enforcement management to effectively managing how DDN is securely configured.**

***Index Terms*—Software-Defined Networking, Database-Defined Networking, Security Policies**

I. Introduction

Complexity and fragility remains one of the main challenges in networking for administrators [8]. In legacy traditional networks, administrators have to convert their high-level polices such as firewall policy and routing policy into low-level vendor-specific configuration rules for each devise within their network and adopt them to match the network updates. However, this process is too complicated because of a large number of connected network devices. According to [10], automatic reconfiguration can't be reached in the traditional networks. Moreover, administrators need more flexibility to control and customize network devices. For that reason, Software-Defined Networking (SDN) emerged as novel paradigm that facilitates network management and enables programmatically towards efficient network configuration in order to improve network performance and monitoring of networks. SDN make it easier for manage security policies on a network system. However, it is still hard to map high-level security policies defined by users into low-level security policies that can be enforced on switches. The idea of insertion software to manage network with centralized controller open the door to a new abstraction to be builds on the top of SDN, one of these are the Database-Defined Network (DDN) controller Ravel [15]. One of the main attractive advantages of Ravel, the using of SQL language to control network nodes. These familiar database notions will allow us to port the rich literature of database techniques to Software-Defined Networks.

We summarize our motivation in the following four points:

1) Networks near-constant policy configuration changes is needed [5].
2) The need of automated method to vet changes at a high level abstraction [12].
3) DDN offers a great abstraction with a huge database features.
4) The need of security by design SDNs [3]

II. A Brief Background on DDNs

Database-Defined Networking (DDN) is a concept of using relational databases as an abstraction for managing an SDN. The concept of a DDN has recently been implemented in a system called Ravel [15], which is a controller that represents a network using a standard relational database. The architecture of Ravel is shown in Figure 1. In Ravel, the network can

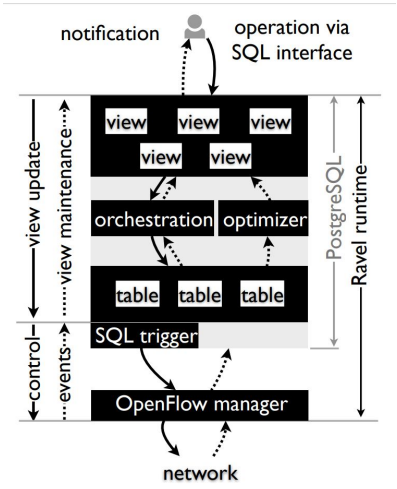

Fig. 1. Ravel Architecture [15]

be *queried* and its configuration *updated* using standard data languages, such as Standard Query Language (SQL). Interestingly, it becomes straightforward to divide the network into multiple zones and enforce access rules on those zones using access control lists [7]. Ravel defines the network in a flat manner exposing the topology and forwarding information in terms of three main tables. These are called `tp` (network topology), `cf` (configuration) and `tm` (reachability matrix).

Additionally, there are other node tables, which include a host and a switch table as well as a generic node table containing the identities and names of all the nodes in the network.

III. Our Approach

The main aim of this research is to improve DDN security and addressing research questions related to security policies enforcement. Moreover, we aim to contribute a novel and efficient high-level policies enforcement framework in the database layer, and take advantage of the database abstraction in order to provide secure network functionality (e.g. Routing, forwarding and access control). The framework would includes algorithms, analyses and implementation. Our initial aim is to achieve a automated transformation model from any network management policy languages to enforceable relational database rules.

IV. Related Work

In recent years, policy-based network management has developed and becomes more vibrant, with the SDN paradigm emerging. We outline below a few related works in this area that have been proposed and implemented recently.

The authors in [6] proposed *Frenetic*, an OpenFlow-based network programming language, which provides an interface to query traffic information and create policies to react to network events. Simplification of network event programming and retrieval of traffic information is the main focus of Frenetic, though it does not provide alternative mechanisms for handling events sent by network switches. *Procera*, another high-level language proposed in [9], allows administrators to define policies and deploy in SDN networks. A dynamic network reconfiguration is required for this framework since it focuses on event-driven networks. According to [1], in order to validate the Procera framework, the scalability of the number of rules and the performance related to the time of translation of these rules to OpenFlow rules remains to be evaluated.

Fresco [13] is another OpenFlow-based security framework, where the security modules are exposed to external users giving them the ability to define and enforce security policies. Definition of the types, input/output parameters, actions and events are all required information for using Fresco. Fresco can be compared to Procera and Frenetic, in terms of allowing network events to be manipulated and in handling them through predefined modules.

Ponderflow [2] uses the Ponder language [4] for managing an OpenFlow network. The main drawback of *Ponderflow*, however, is that it lacks policy conflict resolution mechanisms. In addition to that, no experiments were made by the authors, for translating the proposed Ponderflow language to OpenFlow rules, to validate their approach within a real-world scenario.

OpenSec [11] is another policy-based network security management system, in which the authors focused on simplifying how network security policies are implemented and how they can respond to system alerts. OpenSec implements network policies in a simple language, which is then automatically converted into a set of rules that are set up into the network devices' level. OpenSec allows administrators to define a flow in terms of OpenFlow matching fields and identify which security properties should apply to that flow.

Recently, the authors in [14] proposed a network policy chain criteria based on the Database-Defined Networks approach, they employ the database integrity constraints to provide a logical framework to describe network policies. Moreover, the core idea behind thier work is the semantic modelling of network policies as integrity constraints that is managed by relational database.

V. Conclusion

Since the emerging SDN technology attracts more and more attention and applications with its benefits in flexibility and programmability, controlling the network correctly and efficiently with the new architecture is challenging. The concept of a DDN, as abstraction of an SDN, offers a great opportunity to simplify the network management. This research would create an enforcement framework for Policy-Based Management in Database-Defined Network (DDN). We believe this is a fruitful research area, and are excited about the future research on designing powerful network control platforms with next generation programmable switches to make networks architecture not only more secure but more effective, stable and reliable.

References

[1] Aschoff, R., Rosendo, D., Machado, M., Santos, A., Sadok, D.: A network access control solution combining orbac and sdn (2017)

[2] Batista, B.L.A., Fernandez, M.P.: Ponderflow: A new policy specification language to sdn openflow-based networks (2014)

[3] Dacier, M.C., König, H., Cwalinski, R., Kargl, F., Dietrich, S.: Security challenges and opportunities of software-defined networking. IEEE Security & Privacy **15**(2), 96–100 (2017)

[4] Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The ponder policy specification language. Policy **1**, 18–38 (2001)

[5] Feamster, N., Rexford, J.: Why (and how) networks should run themselves. arXiv preprint arXiv:1710.11583 (2017)

[6] Foster, N., Harrison, R., Freedman, M.J., Monsanto, C., Rexford, J., Story, A., Walker, D.: Frenetic: A network programming language (Sep 2011). https://doi.org/10.1145/2034574.2034812, http://doi.acm.org/10.1145/2034574.2034812

[7] Glaeser, N., Wang, A.: Access control for a database-defined network (2016)

[8] Jammal, M., Singh, T., Shami, A., Asal, R., Li, Y.: Software defined networking: State of the art and research challenges. Computer Networks **72**, 74–98 (2014)

[9] Kim, H., Feamster, N.: Improving network management with software defined networking. IEEE Communications Magazine **51**(2), 114–119 (2013)

[10] Kreutz, D., Ramos, F.M., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: A comprehensive survey. Proceedings of the IEEE **103**(1), 14–76 (2015)

[11] Lara, A., Ramamurthy, B.: Opensec: Policy-based security using software-defined networking. IEEE Transactions on Network and Service Management **13**(1), 30–42 (2016)

[12] Saha, S., Prabhu, S., Madhusudan, P.: Netgen: Synthesizing data-plane configurations for network policies p. 17 (2015)

[13] Shin, S., Porras, P., Yegneswaran, V., Fong, M., Gu, G., Tyson, M.: Fresco: Modular composable security services for software-defined networks (2013)

[14] Wang, A.: Database criteria for network policy chain. In: Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization. pp. 49–54. SDN-NFV Sec'18, ACM, New York, NY, USA (2018)

[15] Wang, A., Mei, X., Croft, J., Caesar, M., Godfrey, B.: Ravel: A database-defined network. In: Proceedings of the Symposium on SDN Research. pp. 5:1–5:7. SOSR '16, ACM, New York, NY, USA (2016)