Computers and Electrical Engineering 000 (2018) 1–16

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A framework for enabling security services collaboration across multiple domains☆

Daniel Migault[a], Marcos A. Simplicio Jr.[b,*], Bruno M. Barros[c,b], Makan Pourzandi[a], Thiago R. Almeida[b], Ewerton R. Andrade[b], Tereza C. Carvalho[b]

[a] Ericsson Security Research, Montreal, Canada
[b] Escola Politécnica, Universidade de São Paulo, Brazil
[c] École normale supérieure, Paris, France

ARTICLE INFO

Article history:
Received 25 April 2017
Revised 16 February 2018
Accepted 19 February 2018
Available online xxx

Keywords:
Cloud computing
Multi-domain networks
Collaborative security
Service Function Chaining
Network function virtualization

ABSTRACT

Network function virtualization opens a new era for security, allowing on-demand instantiation of defense appliances via technologies such as SDN (Software Defined Networking) and Service Function Chaining (SFC). Taking full advantage of such capabilities, however, requires collaboration among Security Service Functions (SSFs) distributed throughout the network. Indeed, collaboration among SSFs is expected to become as essential to SECaaS (SECurity as a Service) as elasticity is to IaaS (Infrastructure as a Service), enabling the efficient allocation of resources for handling large scale attacks. In this paper, we propose a framework leveraging SDN and SFC to improve collaboration among SSFs, allowing SSFs from different domains to negotiate and dynamically control the amount of resources dedicated to collaboration (called a "best-effort" mode). The feasibility, efficiency and scalability of the solution is experimentally assessed, showing that it incurs low overhead, increases the amount of traffic treated by SSFs before packets start being dropped.

1. Introduction

In the context of Network Function Virtualization (NFV), Security Service Functions (SSFs) are a special type of Service Functions (SFs) that usually take the form of on-path services for detecting and mitigating security threats (e.g., a firewall for filtering traffic) [1]. SSFs are instantiated across different administrative domains and must be able to dynamically scale in response to attacks (e.g., Distributed Denial of Service – DDoS). Such large scale deployment of service functions leads to interesting challenges, such as their efficient allocation inside one or (especially) across multiple administrative domains [2]. After all, to ensure its own system is protected, each domain is likely to instantiate SSFs intended to address similar threats, considering only locally-available information, and also to overprovision those SSFs so they can handle eventual traffic fluctuations. Since multi-domain environments are becoming more common as computation is moved to the networks' edge, a trend observed in many areas (e.g., mobile networks in general [3]), underused SSFs end up proliferating.

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. L. Bittencourt.

* Corresponding author.
E-mail address: mjunior@larc.usp.br (M.A. Simplicio Jr.).

https://doi.org/10.1016/j.compeleceng.2018.02.026
0045-7906/

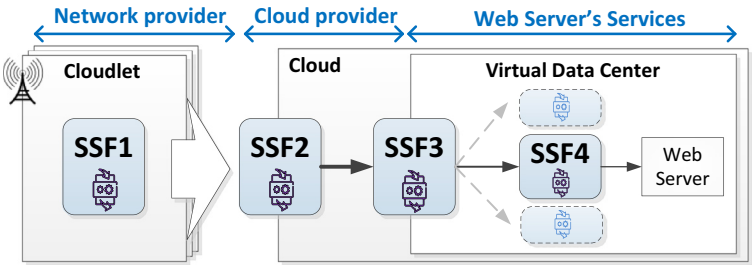

Fig. 1. A multi-domain scenario for SSF collaboration.

Aiming to tap the full potential of deployed SSFs for handling large scale attacks, it is important to enable their collaboration. To give a concrete example, consider the scenario depicted in Fig. 1, in which a web-server is instantiated in a cloud computing environment, within a virtual data center (VDC). The traffic from end-users is conveyed by an Internet Service Provider (ISP) to the Cloud, then to the VDC by the Cloud Provider, and finally to the end server. In this scenario, since the VDC receives all traffic and runs the end server, it is better placed for detecting application- or service-specific attacks, as well as DDoS; the VDC can then inform other domains of the ongoing attack, leading to *better detection*. Then, for *better mitigation*, the VDC can define and implement a coordinated response strategy either locally or by (partially) outsourcing this task to other SSFs on the attack's path. If this strategy is conceived in such a manner that SSFs in each domain concentrate their efforts on tasks that they can perform better, avoiding redundancies, a natural result is *better resource usage*: for example, the Cloud could instantiate SSFs focused on attack detection, since it can inspect all incoming traffic (even if encrypted) and run application-specific tools; the ISP, in turn, can deploy SSFs for filtering malicious packets closer to their sources. Even when the collaboration's scope is a single domain, resource usage optimization is also possible, since an overloaded virtual machine (VM) that runs an SSF can offload some tasks to another VM with idle resources, avoiding instantiating new SSFs.

Another advantage of SSF collaboration is that it creates a *highly dynamic and scalable* architecture for attack mitigation, leading to a robust and cost-effective Security-as-a-Service (SecaaS) solution [4]. For example, in DDoS and similarly distributed attacks, the aggregation of resources on the attackers' side usually overcomes the resources available at any single point on the target's side. A common strategy for dealing with this issue is to redirect the traffic to a scrubbing center, which is supposed to have enough resources to handle the attack. With collaboration, however, the defense is also distributed, combining the capabilities of several SSFs to surpass the attacker's resources. Consequently, collaboration reduces the system's dependency on scrubbing centers.

The literature has many examples of how collaboration can improve security (for a survey, see [5]). These benefits led to several collaborative proposals, most of which focus on signaling the detection of DDoS attacks [6] or mitigating them by outsourcing tasks toward the attack's sources [7]. Unfortunately, however, collaboration has seen restricted deployment in practice. In part, this is due to the cost of adding such features in appliance routers, and to the limited capabilities made available by vendors. Nevertheless, today this issue can be addressed by Software Defined Networking (SDN) [8], a technology with potential to reduce management costs and to diversify the possible interactions among domains. Indeed, the interest of major industry players in enabling and orchestrating collaboration among domains motivated the creation of Internet working groups focused on this topic (e.g., [6,9]).

Despite those advances, there is still one important challenge when enabling collaboration: even though it is expected to benefit all participants, collaboration requires one domain to allocate resources for another domain, which may not seem advantageous at first sight. To encourage such practice, each domain must be able to dynamically control the amount of resources allocated for a collaboration, negotiating it in real time and in a flexible manner. For example, a domain should be allowed to handle only a fraction of the traffic, considering its current load, to avoid undesirable situations such as resource exhaustion or hijacking. Collaboration agreements established for this purpose can, however, be significantly simpler than the formal agreements usually established with scrubbing centers. After all, the collaboration may be temporary and consider only the current capabilities of each domain.

Aiming to create a flexible multi-domain collaboration environment, in [2] we proposed and discussed a collaboration framework that, when compared with previous work, brings the following contributions. First, it enables collaboration among SSFs within or among different administrative domains, as well as the provisioning of new SSF instances on demand if deemed necessary. Second, these instances may be associated: (1) to different SSF types, so the collaboration consists in requesting a functionality; or (2) to a same SSF type, in which case part of an SSF's load is outsourced. Nodes can even participate in what we call a "best-effort" collaboration mode: whenever the amount of resources required by the outsourced task reaches an agreed-upon threshold, the task is not performed by the SSF that engaged in the collaboration, and untreated packets are marked so they can be treated later. In this paper, we extend the aforementioned work by providing additional experimental results, including the efficiency assessment of packet filtering capabilities implemented in different

Please cite this article as: D. Migault et al., A framework for enabling security services collaboration across multiple domains, Computers and Electrical Engineering (2018), https://doi.org/10.1016/j.compeleceng.2018.02.026