PAPER

Assessing Lightweight Virtualization for Security-as-a-Service at the Network Edge

Abderrahmane BOUDI[†*a)], **Ivan FARRIS**[†b)], **Miloud BAGAA**[†c)], *and* **Tarik TALEB**[†d)],

SUMMARY Accounting for the exponential increase in security threats, the development of new defense strategies for pervasive environments is acquiring an ever-growing importance. The expected avalanche of heterogeneous IoT devices which will populate our industrial factories and smart houses will increase the complexity of managing security requirements in a comprehensive way. To this aim, cloud-based security services are gaining notable impetus to provide security mechanisms according to Security-as-a-Service (SECaaS) model. However, the deployment of security applications in remote cloud data-centers can introduce several drawbacks in terms of traffic overhead and latency increase. To cope with this, edge computing can provide remarkable advantages avoiding long routing detours. On the other hand, the limited capabilities of edge node introduce potential constraints in the overall management. This paper focuses on the provisioning of virtualized security services in resource-constrained edge nodes by leveraging lightweight virtualization technologies. Our analysis aims at shedding light on the feasibility of container-based security solutions, thus providing useful guidelines towards the orchestration of security at the edge. Our experiments show that the overhead introduced by the containerization is very light.

key words: *NFV, Security, Cloud/Edge Computing, and IoT.*

1. Introduction

The interest towards cybersecurity is fast growing over the last years accounting for the tremendous effects and damages which can be carried out in our hyper-connected world. The potential attack surfaces are increasing at fast pace leveraging the widespread adoption of Internet of Things (IoT) devices. Furthermore, the heterogeneity of IoT devices, ranging from smart industrial appliances to simple domestic sensors, can even increase the complexity to provide the desired protection [1]. Novel security strategies are required to meet security policies in both industrial and domestic environments.

Accounting for the success of cloud solutions, the provisioning of on-demand security services according to the Security-as-a-Service model [2] is gaining notable attention from both industrial and research communities. In this way, organizations and users can be assisted by cloud-hosted components providing security and privacy protection [3], [4]. On the other hand, the deployment of security instances in remote data centers present several drawbacks, such as long routing detours and delay increase. To face these issues, Edge Computing [5] offers the opportunity to efficiently host services at the network edge, thus introducing remarkable benefits in terms of shortening latency and traffic reduction.

In this paper, we aim at investigating the provisioning of security services in resource-constrained edge nodes, such as network access points and IoT gateways. In this vein, we will evaluate Docker containers as promising lightweight virtualization technology [6]. We strongly believe that performance analysis of security defense systems is of utmost importance, since security mechanisms can notably influence the overall Quality of Service (QoS) [7]. Our analysis aims at shedding light on the feasibility of container-based security services in resource-constrained devices, assessing relevant resource consumption in a realistic testbed environment for a broad range of possible workloads.

The paper is organized as follows. In Section 2, we present a background on cloud-based security functions and edge computing features. Two promising case studies are discussed in Section 3. In Section 4, we present Security-as-a-Service features in edge environment, accounting for the constraints and challenges introduced by resource-constrained edge nodes. Section 5 reports the performance evaluation of container-based security functions. While we list some promising open research challenges in Section 6, concluding remarks are drawn in Section 7.

2. Background

2.1 Cloud-based Security Functions

Accounting for the remarkable benefits introduced by cloud service provisioning, an increasing number of security vendors are exploiting cloud ecosystems to provide their security solutions. This approach, referred

Manuscript received July 7, 2018.
Manuscript revised July 7, 2018.

[†]Dep. of Communications and Networking School of Electrical Engineering, Aalto University, Espoo, Finland. T. Taleb is also with Oulu University, Oulu, Finland, and with Sejong University, Seoul, Korea.

[*]Laboratoire de la Communication dans les Systèmes Informatiques,École nationale Supérieure d'Informatique, Algiers, Algeria
a) E-mail: a_boudi@esi.dz
b) E-mail: ivan.farris@aalto.fi
c) E-mail: miloud.bagaa@aalto.fi
d) E-mail: tarik.taleb@aalto.fi
DOI: 10.1587/trans.E0.??.1

to as SECurity-as-a-Service (SECaaS) [2], is based on the provisioning of virtual security applications via the cloud, thus leveraging greater flexibility and economies of scale. In this vein, the Cloud Security Alliance (CSA) has defined guidelines for cloud-delivered defense solutions, to assist enterprises and end-users to widely adopt this security paradigm shift [8].

In this landscape, specific research efforts aim at developing schemes to appropriately model virtualized security services and to provide guidelines for efficiently integrating security services within standard cloud delivery solutions [9]. In [10], an approach towards the adoption of security policies management with dynamic network virtualization is presented. In particular, three different policy abstraction layers are defined and an iterative refinement process is proposed to determine the resources necessary to enforce specific security features through the provisioning of selected virtualized security functions. To meet the desired objectives and to avoid deviation from the expected policies' goals, an accurate estimation of the requirements for virtualized functions becomes crucial, as well as the management of the overall lifecycle.

Accounting for the significant advantages introduced by replacing dedicated network hardware with software instances, Network Function Virtualization (NFV) is gaining high momentum to enhance the scalability and flexibility of softwarized networks [11]. In [12], a framework for characterizing performance of virtual network functions has been developed, to determine optimal resource configuration for a given workload and useful insights to scale up or down relevant instances. Among the analyzed functions, the analysis of Intrusion Detection System (IDS) executed in virtual machines have been tested for cloud environments. Indeed, the performance of virtualized components can have a great impact on the overall service chaining, accounting for the hardware settings and virtualization technologies overhead [13]. The objective of this paper is to consider the evaluation of container-based technologies for providing security mechanisms in resource-constrained edge nodes.

2.2 Lightweight Virtualization for Edge Computing

Over the last years, Edge Computing has received more attention, accounting for the opportunity to extend the successful cloud model towards the edge of the network. In this way, great advantages can be introduced in terms of reduced latency, traffic reduction, and context-awareness. Not by chance, edge computing is considered as a pillar of next-generation 5G networks [14], [15], able to support demanding verticals such as massive IoT, virtual reality, and Tactile Internet [16], [17]. Also, standardization bodies and industrial consortia are promoting its widespread adoption by creating specific study groups, thus leading to ETSI Multiple-access Edge Computing (MEC) Working Group [18] and Open Fog Consortium [19].

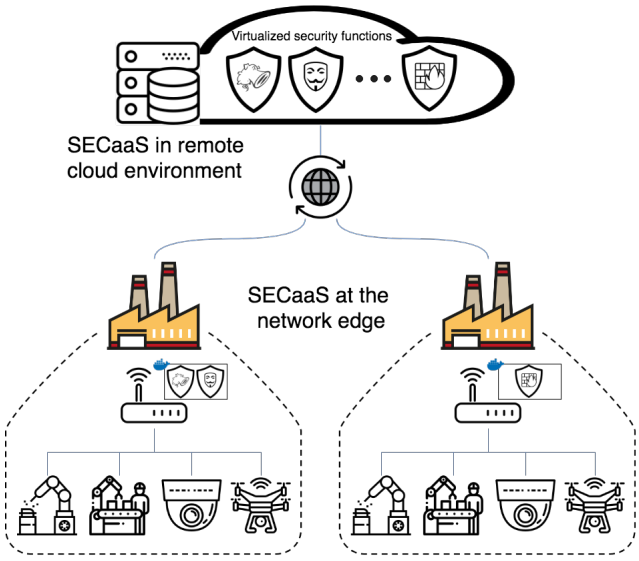


Fig. 1 Security-as-a-Service in industrial edge scenarios.

However, new challenges are introduced in the deployment of service instances at the network edge. Especially when considering resource-constrained edge nodes, lightweight virtualization technologies are strictly required. In this vein, container-based virtualization is able to offer several benefits with respect to classic hypervisor-based virtual machine environments: (i) fast creation and initialization of virtualized instances; (ii) high density of applications, thanks to the small container images; (iii) reduced overhead, while enabling isolation between different instances running in the same host [6], [16].

As discussed in [20], Docker containers represent a promising platform for Edge Computing. In this work, Docker has been evaluated in terms of deployment and termination, resource and service management. Different application fields for container-based virtualization have been demonstrated. Container technologies are used in a Capillary Network scenario [21], where Docker containers allow to package, deploy, and execute different functionalities at the capillary gateway. In [22], lightweight virtualization technologies are used to deploy on-demand gateway features for the Cloud of Things. However, an analysis of container technologies for security services is still missing.

3. Case studies

In this section, we present two promising use cases in both industrial and domestic environments which strongly push the need for provisioning security functions at the edge with advanced flexibility compared to classic dedicated hardware solutions.

3.1 Factory 4.0

The fourth industrial revolution is next-to-come and will be boosted by a progressive digitalization of industrial production processes. In this fervent ecosystem, sensors and actuators will play a fundamental role to bridge the physical and virtual domains by providing the necessary capabilities to monitor the industrial environment and to promptly react. Furthermore, automated robots are expected to provide real-time information about operational behavior, for enabling both remote quality of product and maintenance analysis [23]. The increased connectivity of industrial systems will thus be the key factor for next-generation Factory 4.0.

The dark side of the medal of this increased openness will be represented by the new potential security vulnerabilities which can be exploited by malicious attackers [24], [25]. Indeed, security threats can cause catastrophic effects in industrial environments leading to process interruption, product adulteration, and even health risk for workers operating in strict synergy with robots. These accidents can provoke huge losses in revenues and brand reputation, thus undermining the overall digitalization of industrial revolution.

Further challenges of industrial environments deal with the confidentiality of information gathered during production processes. Data leakages can also advantage potential competitors, and consequently companies are reluctant to have their data processed outside their boundaries. In this complex scenario, the increased abstraction capabilities of edge node can provide the appropriate environment to execute virtualized secure functions, as sketched in Fig. 1. For instance, enhanced gateway can forward data to/from industrial sensors and analyze relevant traffic flows to identify potential security vectors. Only the verified data can be admitted and security alerts are logged. Key aspects deal with the analysis of performance ensured by virtualized security functions in resource-constrained edge nodes. In this way, the interplay of virtualized security functions between cloud and edge can be further improved and novel offloading strategies can be developed, specifically tailored to the constraints of virtualized edge nodes.

3.2 Smart Home

A myriads of IoT devices will transform our houses in smart pervasive environments, ranging from smart kitchen appliances to tiny light sensors. A key factor is their enhanced interworking to exchange and cooperate with neighboring devices, as well as back-end applications hosted by cloud platforms. The dark side of this connectivity relates to the new potential security vectors which attackers can leverage to lead their malicious activities. Indeed, in October 2016, cybercriminals launched a Distributed Denial of Service (DDoS) attack[†] against an Internet Service Provider Dyn, thus disrupting access to several popular websites. To carry out this attack, a large number of internet-connected devices (mostly DVRs and cameras) were maliciously exploited by leveraging some firmware security flaws. The heterogeneity of devices make extremely complex to guarantee the desired security requirements for end-users.

To enhance defense mechanisms, the security-as-a-service paradigm can be promoted by Telco operators, which can provide routers/gateways with enhanced virtualization capabilities to their subscribers. A broad range of services can be deployed on-demand within the home environments, while enabling the creation of local edge clouds able to secure and verify the communications from/into domestic environments. In this way, potential sensitive information included in the traffic flows can be processed locally, thus preserving relevant confidentiality. For instance, IDS can be deployed to verify malicious traffic between personal IoT devices and remote cybercriminals. When potential threats are detected, security alerts are launched to inform the end-users and to trigger the adoption of appropriate countermeasures.

4. Container-oriented Edge Management of Virtual Security Functions

Container-based virtualization can drastically reduce the overhead with respect to classic hypervisor-based virtualization. Instead of executing a full operating system in each virtual machine instance, containers can run on top of the same kernel provided by the underlying host machines. Indeed, containers leverage two key features of the operating system kernel, i.e., namespaces and control groups (cgroups). Linux namespaces allow to isolate processes (i.e. containers) from each other, whereas cgroups can be used to allocate specific amount of resources, such as CPU, memory, and block device I/O, to each container instance. Compared to full and para-virtualized approaches, container virtualization is directly done in the kernel, thus guaranteeing better performance [6].

In this paper, we use application-oriented Docker containers for executing security functions. Docker introduces an underlying container engine, the so-called Docker Engine, together with functional APIs that allow for easy building, management, and removal of a virtualized service. With respect to system-level containers, e.g., OpenVZ and LXC, application-oriented containers better cope with the microservice paradigm, which is considered the next big revolution for cloud-

[†]http://www.zdnet.com/article/dyn-confirms-mirai-botnet-involved-in-distributed-denial-of-service-attack/

based service provisioning [26].

Indeed, by enabling a virtualized environment even in resource-constrained edge nodes several advantages can be introduced:

- Flexibility: High levels of flexibility are required to dynamically launch different security functions to face new protection requirements, as well as scaling up/down instances according to the varying workload and resources' availability.
- Portability: This represents one of the most appealing features since by packing security software along with its dependencies into a single image container enables image-based deployment process, thus offering the freedom of *develop once, deploy everywhere*.
- Manageability: This feature allows a manager to facilitate the provisioning of security applications on the same infrastructure composed of edge nodes, by leveraging the virtualization provided by Linux containers.
- Reliability: To achieve fault tolerance, a continuous monitoring of devices and containerized applications can provide the ability of fast adaptation and reconfiguration, e.g., including specific high-availability mechanisms for security functions.

Another core aspect to boost container adoption in production environment concerns the development of orchestration systems to facilitate the deployment and management of multiple containerized applications across a number of either physical or virtual hosts [27]. The most popular solutions are Kubernetes, Docker Swarm, and Apache Mesos. In particular, Docker Swarm is gaining momentum since it is natively integrated with Docker distribution. A Docker Swarm is a cluster of running Docker Engines, which leverages the management features provided by the SwarmKit. Two different logical entities are defined: (i) the *manager node*, which performs the orchestration functions required to maintain the desired state of the swarm and dispatches units of work called "tasks"; (ii) the *worker nodes*, that receive and execute tasks scheduled by the manager. By default, manager nodes are also worker nodes, but it is possible to configure managers to be manager-only nodes. The agent notifies the current state of its assigned tasks to the manager node, so that the manager can continuously monitor the state of the cluster. Each node in the swarm enforces Transport Level Security (TLS) mutual authentication and encryption to secure its communications with all other nodes. The standard Docker API can be used to implement swarm management procedures, so to deploy security services to the swarm and carry out service orchestration. While legacy Docker Engine issues container commands, Docker Swarm mode orchestrates "services", which are the definitions of the tasks to be executed on the worker nodes. The service definition allows to specify the container image to use and the commands to execute inside the relevant container when the service is created. To enable coexistence of different security services over constrained edge nodes, the specifications of resource constraints must be considered in the overall orchestration to ensure the desired performance level. This aspect can be particularly difficult accounting that requirements of different virtualized security functions can vary significantly. To this aim, in the following, we carry out an exemplary characterization of container-based protection solution according to different workloads.

5. Performance evaluation

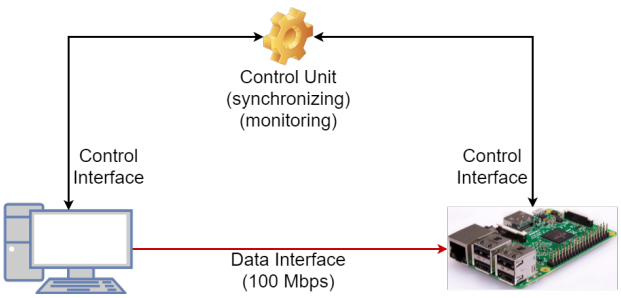


Fig. 2 Simulation setup.

In this section, we aim at comprehensively assessing the performance of virtualized security functions in resource-constrained edge nodes in a real testbed setup. Our objectives are also to demonstrate the feasibility of efficiently adopting container-based virtualization, by comparing the native execution of security functions and their respective containerized counterparts. In our analysis, we focus on: i) the number of processed packets; ii) network utilization; iii) the number of alerts; iv) RAM utilization; v) CPU load; and vi) the number of dropped packets.

The testbed setup consist of Suricata running on a Raspberry Pi3 edge node (Fig.2). The experimental results show the difference between Suricata running on bare metal (SoBM) and running inside a Docker container (SoDC). In order to detect attacks, Suricata needs a rule set that characterizes these attacks. In this set of simulations, the emerging threats rules set is used. The traffic is generated from pcap files that contain excerpts of legitimate and malicious traffic. Our experimental setup is similar to the one used in [28]. We consider two pcap files: in the first file the number of small packets outnumbers the number of large packets; in the second file, the number of packets are quite similar, which means that the traffic generated by the small packets is tiny compared to the traffic generated by the large packets.

Finally, the rate of the traffic is varied from $10Mbps$ up to $90Mbps$ for large packets and from

$10Mbps$ to $50Mbps$ for small packets. The reason why we cannot go beyond $50Mbps$ is that the Raspberry begins to heat up and crashes, therefore the results are not reliable in those conditions. Finally, each simulation setup is run 10 times and the average and standard deviation are plotted for each experiment.

5.1 Processed packets

Fig.3(a) and Fig.4(a) show the number of processed packets in relation to the traffic rate. It represents the number of packets that were sent to the detection engine of Suricata. It contains both legitimate and malicious packets. Overall, SoDC and SoBM process roughly the same number of packets. It should be noted that when the packets are small, there are four times more traffic than there is in large packet simulations.

5.2 Network utilization

Fig.3(b) and Fig.4(b) depict the performance of network utilization. From the obtained results, we can conclude that SoBM slightly outperforms SoDC. As before, running Suricata on bare metal or on a Docker container does not show a clear difference in the performance. Unlike previously, the network utilization is greater when the packets are large.

5.3 Alerts

The alerts are the number of successfully detected threats. Given the fact that, in large packets simulations, it is the same pcap file that is played over and over again, it is clear then that the number of alerts will increase with the rate at which the traffic is played. In all simulations, SoBM and SoDC have detected the same number of alerts (Fig.3(c) and Fig.4(c)). From one simulation to the next, the packets that were dropped and their number varies greatly. That is why there is a high variability in the number of detected attacks.

5.4 RAM utilization

In large packets simulations, the RAM usage ranges between 26% and 28%. While in small packets runs, the RAM reaches 50% of utilization, in both scenarios, SoBM and SoDC have shown similar RAM usage (Fig.3(d) and Fig.4(d)).

5.5 CPU utilization

The size of packets has a big impact on the load exerted on the CPU. Even when the traffic rate is only $50Mbps$, the average utilization is beyond 80% if the packets are small. SoDC shows slightly better CPU usage than SoBM. The difference is around 2% when the packets are large and reaches 6% when they are small. Investigating this situation shows that SoDC is taking more time running on kernel space, while SoBM is taking more time on user space. It should be noted that roughly the same number of packets is received by Suricata when the rate is $90Mbps$ for large packets and $20Mbps$ for small packets, that is why the CPU load is high when the traffic mainly consists of small packets.

5.6 Number of drops

Fig.3(f) and Fig.4(f) show the percentage of drops occurred during the performance evaluation. In Fig.4(f), the dropping began at $20Mbps$, and the percentage of dropped packets increases with the bandwidth. One fifth to one forth of the packets were dropped when the rate was at 50%. This is the reason of the variability of the number of alerts in Fig.4(c). When there are large packets, the number of drops is less than 2%, even when the rate is 90%. The reason why SoDC shows less drops is due to two main reasons. As shown in Fig.3(e) and Fig.4(e), SoDC has less impact on the CPU on average. Therefore, it is less prone than SoBM to drop packets due to bursts. The second reason is that SoBM generally receives more packets than SoDC, thus SoBM has to drop more packets. Fig.5 shows the ratio between the number of successfully processed packets by SoBM and SoDC (Eq.1).

$$ratio = \frac{ptks_{bm} - drop_{bm}}{ptks_{dc} - drop_{dc}} \quad (1)$$

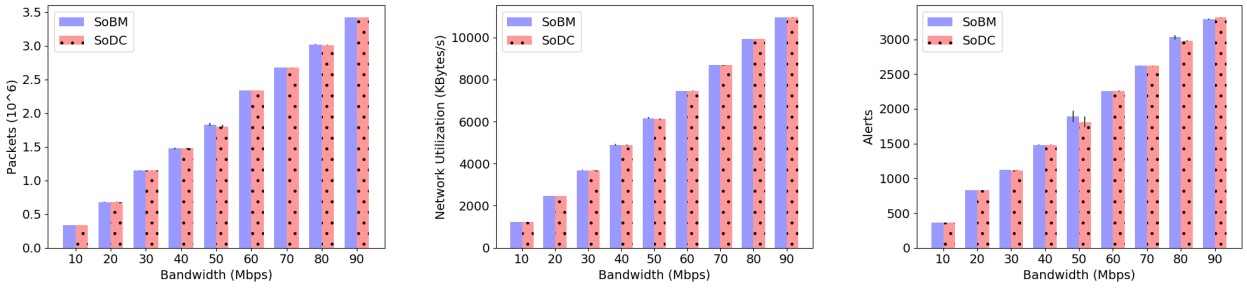
where $ptks_{bm}$ and $ptks_{oc}$ denote the number of packets received by SoBM and SoDC, respectively. $drop_{bm}$ and $drop_{dc}$ are the number of drops performed by SoBM and SoDC, respectively. As it can be seen in Fig.5, SoBM processes slightly more packets than SoDC.

5.7 Discussion

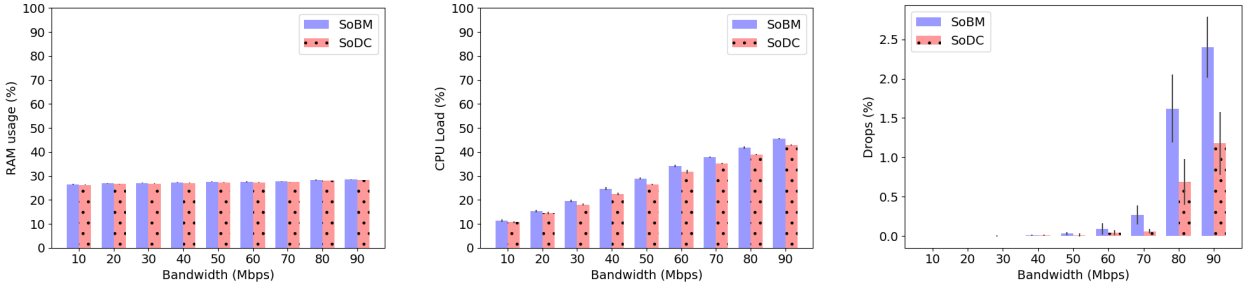
The aforementioned performance results have shown that the overhead introduced by the containerization of security functions is very light. It should be noted, though, that the Docker container had full control over the network interface and only one Docker container was running during the performance evaluation. Using a bridged network would have negatively impacted the performances. Running two containers can bring a notable impact on the performance of the Raspberry Pi. Even then, given the obtained results, it is clear that lightweight virtualization, even in small devices such as a Raspberry Pi, is quite efficient. Therefore, the dynamic deployment of containerized security functions in the network's edge is a very interesting prospect.

6. Open research challenges

The joint use of lightweight virtualization and edge computing represents a promising environment to provide SECaaS, considering the multiple envisioned benefits reported in the previous sections. Furthermore, this study opens up several research challenges to be further

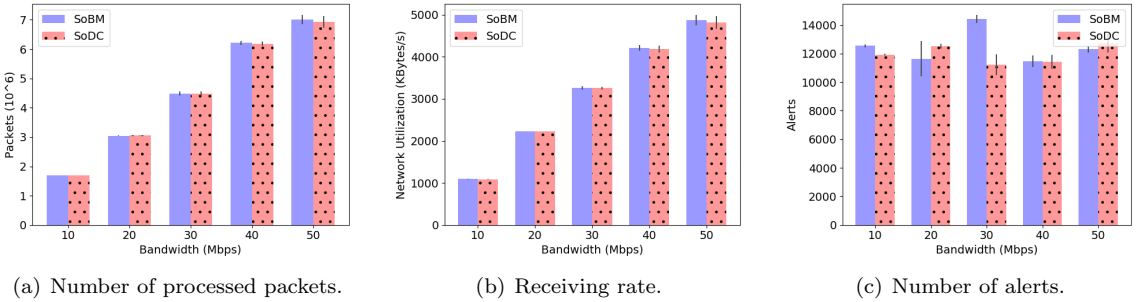


(a) Number of processed packets. (b) Receiving rate. (c) Number of alerts.

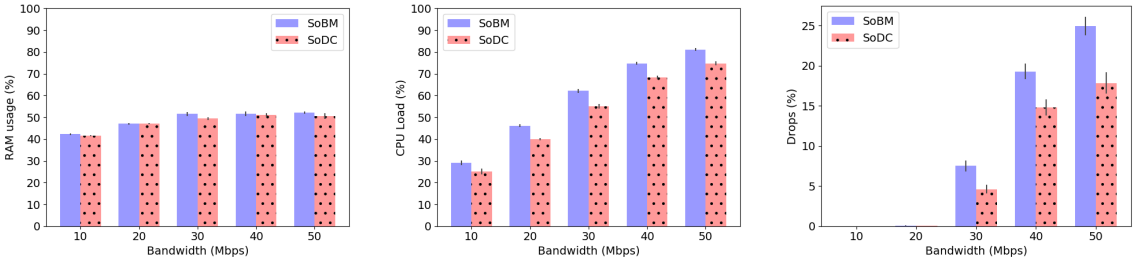


(d) RAM usage. (e) CPU load. (f) Number of drops.

Fig. 3 SoBM vs SoDC results (Large packets).



(a) Number of processed packets. (b) Receiving rate. (c) Number of alerts.



(d) RAM usage. (e) CPU load. (f) Number of drops.

Fig. 4 SoBM vs SoDC results (Small packets).

investigated for an efficient provisioning of security features at the network edge.

- *Security services orchestration*: A key feature of edge computing concerns the opportunity to spread and coordinate service provisioning among distributed edge nodes to efficiently balance workload. However, as discussed in Section 4, current orchestration solutions have been mainly designed for data center environments and further efforts are required to cope with challenges of resource-constrained edges. Also, multiple devices can collaboratively perform security functions, providing value-added service benefits. For instance, in the case of intrusion detection scenarios, each containerized IDS instance can share contextual information with neighboring nodes, so it can dy-

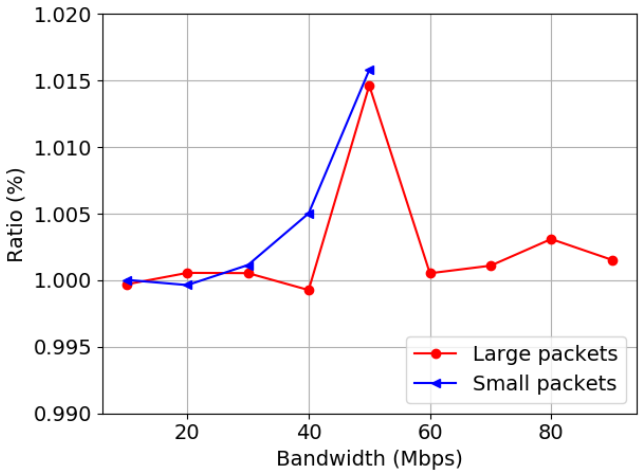


Fig. 5 Ratio of successfully treated packets by SoBM over SoDC.

namically refine the detection process.

- *Security of container virtualization*: Container virtualization heavily relies on underlying kernel features to provide the necessary isolation for virtualized services [29], [30]. Therefore, specific efforts should address the relevant security concerns, accounting also for misleading configurations of relevant container options. Furthermore, a complex ecosystem has been developed around the Docker virtualization technologies, including container image repositories and orchestration platforms. These complementary tools introduce new security challenges which go beyond the classical host domain, involving for instance the integrity of container images during transfer over insecure Internet connections, as well as the interactions with potentially untrusted management modules.

7. Concluding remarks

The community of academic and industrial researchers has paid remarkable attention towards the adoption of cloud-based security functions to provide on-demand defense mechanisms against the increasing malicious ICT attacks. To benefit from reduction in latency and network traffic overhead, edge environments are promising candidates to host virtualized security functions. However, the resource constraints of edge nodes can impact the overall performance of SECaaS solutions. In this paper, we shed light on the provisioning of security functions via lightweight virtualization technologies, by assessing the performance of Docker container-based IDS Suricata in a real testbed. Future works will explore the open challenges envisioned in Section 6 to boost SECaaS at the network edge. Furthermore, we will extend the characterization of containerized security functions to efficiently orchestrate security over distributed edge nodes.

Acknowledgment

An abridged version of this paper has been published in the proceedings of the 2018 edition of the IEEE CSCN [31]. This work was partially supported by the ANASTACIA project, that has received funding from the European Unions Horizon 2020 Research and Innovation Programme under Grant Agreement N. 731558 and from the Swiss State Secretariat for Education, Research and Innovation. This work was also supported in part by the Academy of Finland 6Genesis Flagship (Grant No. 318927).

References

[1] M.M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the Internet of Things," Services (SERVICES), 2015 IEEE World Congress on, pp.21–28, IEEE, 2015.

[2] V. Varadharajan and U. Tupakula, "Security as a service model for cloud environment," IEEE Transactions on Network and Service Management, vol.11, no.1, pp.60–75, 2014.

[3] I. Farris, J. Bernabe, N. Toumi, D. Garcia, T. Taleb, A. Skarmet, and B. Sahlin, "Towards provisioning of SDN/NFV-based security enablers for integrated protection of IoT systems," 2017 IEEE Conference on Standards for Communications and Networking (CSCN), Helsinki, pp.187–192, Sept 2017.

[4] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A survey on emerging sdn and nfv security mechanisms for iot systems," IEEE Communications Surveys Tutorials, (to appear).

[5] I. Farris, T. Taleb, H. Flinck, and A. Iera, "Providing ultra-short latency to user-centric 5G applications at the mobile network edge," Transactions on Emerging Telecomm. Technologies (ETT), Mar. 2017. DOI 10.1002/ett.3169.

[6] R. Morabito, J. Kjällman, and M. Komu, "Hypervisors vs. lightweight virtualization: a performance comparison," IEEE International Conference on Cloud Engineering (IC2E), IEEE, 2015.

[7] T. Taleb and Y. Hadjadj-Aoul, "QoS2: a framework for integrating quality of security with quality of service," Security and communication networks, vol.5, no.12, pp.1462–1470, 2012.

[8] "Defined Categories of Service 2011," tech. rep., Cloud Security Alliance - SecaaS WG, 2011.

[9] A. Furfaro, A. Garro, and A. Tundis, "Towards security as a service (SecaaS): On the modeling of security services for cloud computing," Security Technology (ICCST), 2014 International Carnahan Conference on, pp.1–6, IEEE, 2014.

[10] C. Basile, A. Lioy, C. Pitscheider, F. Valenza, and M. Vallini, "A novel approach for integrating security policy enforcement with dynamic network virtualization," Network Softwarization (NetSoft), 2015 1st IEEE Conference on, pp.1–5, IEEE, 2015.

[11] T. Taleb, A. Ksentini, and R. Jantti, ""Anything as a Service" for 5G mobile systems," IEEE Network, vol.30, no.6, pp.84–91, November 2016.

[12] L. Cao, P. Sharma, S. Fahmy, and V. Saxena, "NFV-vital: A framework for characterizing the performance of virtual network functions," Network Function Virtualization and Software Defined Network (NFV-SDN), 2015 IEEE Conference on, pp.93–99, IEEE, 2015.

[13] R. Bonafiglia, I. Cerrato, F. Ciaccia, M. Nemirovsky, and