Security as a Service Platform Leveraging Multi-Access Edge Computing Infrastructure Provisions

Pasika Ranaweera*, Vashish N. Imrith†, Madhusanka Liyanage‡, Anca Delia Jurcut§
*‡§School of Computer Science, University College Dublin, Ireland
†Department of Electrical and Electronics, University of Mauritius, Mauritius
‡Centre for Wireless Communications, University of Oulu, Finland
Email: *pasika.ranaweera@ucdconnect.ie,†vashish33@gmail.com,‡madhusanka@ucd.ie, §anca.jurcut@ucd.ie,
‡madhusanka.liyanage@oulu.fi

***Abstract*—The mobile service platform envisaged by emerging IoT and 5G is guaranteeing gigabit-level bandwidth, ultra-low latency and ultra-high storage capacity for their subscribers. In spite of the variety of applications plausible with the envisaged technologies, security is a demanding objective that should be applied beyond the design stages. Thus, Security as a Service (SECaaS) is an initiative for a service model that enable mobile and IoT consumers with diverse security functions such as Intrusion Detection and Prevention (IDPaaS), Authentication (AaaS), and Secure Transmission Channel (STCaaS) as a Service. A well-equipped edge computing infrastructure is intrinsic to achieve this goal. The emerging Multi-Access Edge Computing (MEC) paradigm standardized by the ETSI is excelling among other edge computing flavours due to its well-defined structure and protocols. Thus, in our directive, we intend to utilize MEC as the edge computing platform to launch the SECaaS functions. Though, the actual development of a MEC infrastructure is highly dependent on the integration of virtualization technologies to enable dynamic creation, the deployment, and the detachment of virtualized entities that should feature interoperability to cater the heterogeneous IoT devices and services. To that extent, this work is proposing a security service architecture that offers these SECaaS services. Further, we validate our proposed architecture through the development of a virtualized infrastructure that integrates lightweight and hypervisor-based virtualization technologies. Our experiments prove the plausibility of launching multiple security instances on the developed prototype edge platform.**

***Index Terms*—5G, Security as a Service, Virtualization, Multi-Access Edge Computing (MEC), Edge Computing, Docker, Security, Internet of Things (IoT)**

I. Introduction

Novel telecommunication and information systems are highly reliant on a storage and processing infrastructure that attributes ambient intelligence for effectuating autonomous operations in a swift manner. Thus, capacity and processing power are proliferated at information storage facilities to cater these novel requirements. Cloud computing was one such paradigm introduced for overcoming the pitfalls associated with the data centres that accommodate dedicated servers. Heterogeneous cloud infrastructure deployments served as private, public, and hybrid clouds offered the services Infrastructure as a Service models (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and Serverless computing that featured the benefits: outsourced maintenance, minimal management, elasticity, pay per usage, ubiquity, and convenient access mechanism [1]. Thus, the industries and general public opted for employing cloud services to host web services, store industrial and personal data for its convenience. Though, the current advances in technological fronts manifests that cloud computing is no longer a panacea. The centralized and geographically distributed placement restricts the ubiquity that is required for the emerging technologies such as enhanced Mobile Broadband (eMBB), massive Machine-type Communication (mMTC), and Ultra-reliable Low-latency Communication (URLLC) [2]. From the security perspective, the subscriber unawareness of their data location and the nature of manipulations that their data is subjected to is raising privacy concerns [3].

Edge Computing (EC) overcome these limitations and significant research is directed for investigating solutions to latency and bandwidth issues that are associated with the existing cloud computing paradigm. In spite of its origin in 2009 (i.e.Cloudlet introduction), even after a decade, EC is still at its early design stages, apart from small scale projects [4]. Various edge computing paradigms as in Multi-Access Edge Computing (MEC), Fog, Mobile Cloud Computing (MCC), and cloudlets are envisaging the deployment of applications such as Internet of Things (IoT), Ultra High Definition (UHD) streaming, Augmented Reality (AR), Tactile internet, Machine Type Communication (MTC), and Vehicle-to-Everything (V2E) [2], [5]–[7].

A. Motivation

The heterogeneity of IoT devices and compatibility concerns they are raising restricts the service providers to embed security measures in the same level. Therefore, provisioning security services externally as a third-party solution is demanding due to its flexibility and dynamic nature; to cater

the services considering the requirements and the capabilities (financial) of the consumer. Offering security services to the EC subscribers however, has a lesser representation in the current literature compared to other impending services. The concept of Security as a Service (SECaaS) was considered to be implemented with cloud environments as presented in [8]–[12]. In fact, SECaaS was introduced to provision virtual security applications of intrusion detection, network security monitoring, and authentication strategies; leveraging the cloud service platforms to offer flexibility and scalability to the consumers [11]. The limitations that were pointed out regarding cloud computing fades the feasibility on a practical SECaaS implementation. The EC infrastructures however, improves the realization of SECaaS potential with the attributed ultra-low latency, higher serving bandwidth, locational awareness, and real-time responsiveness [13]. The feasibility of implementing an effective edge computing platform is highly reliant on the virtualization technologies intended to be deployed. Moreover, the standardization and stability of the EC flavour to be adopted is imperative to practically achieve the SECaaS directive. To that front, we are confident on utilizing MEC among other EC paradigms due to its standardized architecture and wide adaptability as discussed in [14]. However, prevailing literature do not state a solid method for deploying virtualization at the edge platform of MEC. The aim of this research is to seek the feasibility of launching SECaaS with MEC EC provisions.

B. Contribution

In this paper, we are proposing a novel SECaaS architecture that leverages the MEC edge platform for providing consumers with various security services in order to ensure their security and privacy. Additionally, the security service providers could gain valuable insights from this approach and integrate their services with MEC, which is an inevitable deployment in the future. Our goal is to develop an edge computing environment that is accustomed to MEC standardization, by employing virtualization technologies. We have launched multiple security services simultaneously to verify the proposing edge platform. Moreover, each security service was published via a Transmission Control Protocol (TCP) port number, to be accessible throughout the edge infrastructure.

Rest of the paper is organized as follows: Section II summarizes the state-of-the-art research conducted in relation to this papers' direction. Section III introduces the novel proposed SECaaS architecture. Section IV describes the testing environment, while Section V presents the results and discusses their significance. Finally, Section VI concludes the paper.

II. Related Work

Khettab et al. in [15] proposed an architecture which amalgamated Network Function Virtualization (NFV) and Software Defined Networks (SDN) to ensure 5G network slice security through security functions provisioned as Virtual Network Functions (VNFs). The goal of the proposed model was to perform Optimal Resource Provisioning to Reduce the Operational Expenditure (OPEX) by launching VNFs within different slices to leverage the elasticity and flexibility of NFV. The security tools of Snort, Suricata, and Ntopng are launched as VNFs to form the SECaaS model that attribute dynamic deployment, performance tracking, and predictive auto-scaling. A performance evaluation was conducted to determine the scalability of the SECaaS tools. However, this paper does not explicitly specify the possibility to launch the proposed architecture in edge computing scenarios.

Boudi et al. in [16] conducted an assessment of container based technologies to select the best approach for furnishing security mechanisms to resource constrained edge nodes. Two case studies of Factory 4.0 and Smart Home was stated by the paper to realize the applicability of SECaaS based edge services. Docker containerization was utilized for evaluating the performance with a Raspberry Pi 3 edge node tested for various scenarios.

Sforzin et al. in [17] proposed a robust and scalable security solution for IoT environments to defend against cyber attacks. In this research an intrusion detection architecture was presented utilizing Raspberry Pi as the core commodity for simulating a resource constrained IoT node. Snort was used to evaluate the performance of the simulated node on the intention of determining the optimal configuration for sustainable operation.

Tripathi et al. in [18] explored the possibility of utilizing Raspberry Pi as an Intrusion Detection System (IDS) against cyber attacks. In addition, the implemented security functions included a honeypot, packet analyzer, and a firewall. The proposed system was tested in a home network with Snort as the respective IDS. The current literature fail to consider standardized MEC architecture for realizing their goals. Even-though the stated related researches conducted experiments on Snort, Suricata, and firewall functions, their performance in simultaneous operation sharing the same virtual resources was not considered. Thus, we have conducted the relevant experiments in Section IV to validate our proposing SECaaS architecture.

III. SECaaS Architecture

A. MEC Edge Platform

As depicted in Fig. 1, our focus is capitalized on the MEC edge level for this paper. The MEC edge level is bound to serve several Mobile Edge Services (MESs) such as AR, video streaming, V2E, in addition to the SECaaS services. Thus, we propose that each MES could be provisioned under a Mobile Edge Host (MEH); which is the main operating entity in the MEC edge level [7]. MEHs are launching Mobile Edge Applications (ME Apps) related to a particular service that interface with a User Equipment Application (UE App) in the UE [19]. As these MEHs might be commissioned to handle considerable amount of mobile or IoT devices (UE Apps), a function of a MEH could be managed by a VM to cater the required resources. Though, ME Apps should attribute light-weight virtualization characteristics due to flexibility, less resource consumption, dynamic creation and deletion requirements.

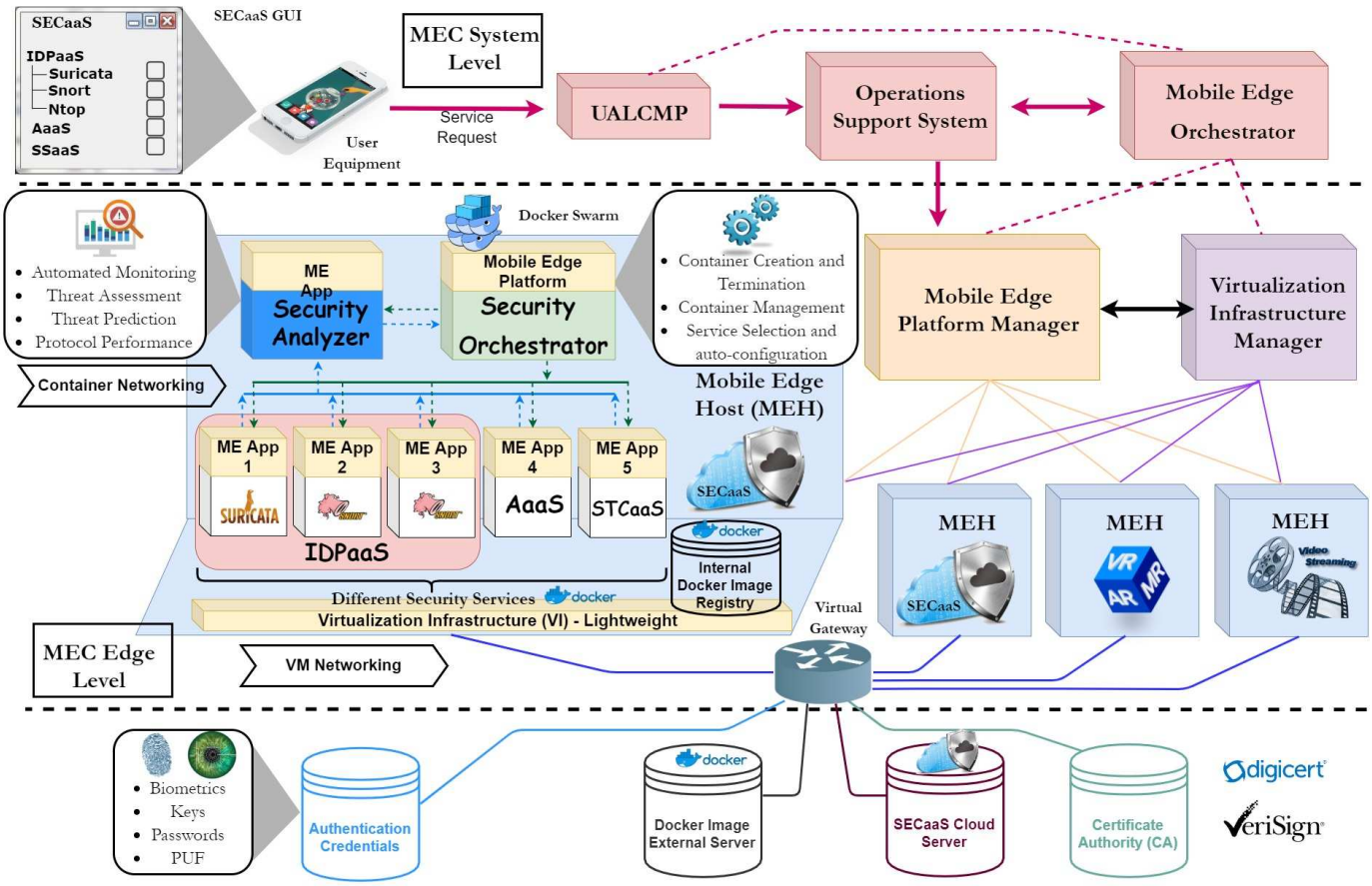


Fig. 1: SECaaS Architecture

Docker is the ideal technology that suits these criteria [16]. Our principal proposal is to employ docker containerization to launch multiple ME Apps as containers which are operating inside a VM. Mobile Edge Platform (MEP), the management entity within the MEH is to be launched as a container manager, or the manager node in the Docker swarm mode [20]. In that aspect, MEP should perform the functions of; i) creation and termination of containers according to UE App requests, ii) selection of services for created containers and auto-configuration, iii) Networking among containers, and iv) managing the container resources. The containerized network that connects all the containers and MEP together is outbound by a virtual gateway. This virtual gateway is connecting the internal container network to other VMs at the edge level and to the Internet with VM network adaptation.

The MEPM is developed as a VM while each MES offered by the MEC system is launched as a VM. The VIM functionality is to be launched using the current hypervisor technologies (i.e.: Microsoft Hyper-V or VMware ESXi). The entire MEC edge level is launched as a singular virtual platform governed by the selected VIM technology. Though, cluster of servers are required to host the virtual platform that should be dynamic. The most suited hypervisor technology should be evaluated through performance valuation considering their flexibility, resource utilization for creating and maintaining VMs, compatibility, interoperability, and support for heterogeneous mobile services. A bare metal hypervisor is an ideal launching platform for this proposed environment. This objective however, is exceeding the scope of this paper.

B. SECaaS Mobile Edge Service

As illustrated in Fig. 1, each security service (i.e. ME App) offered to the consumer is executed as a Docker container. Operating a single ME App to cater a specific UE App would raise issues in terms of scalability when multitude of UEs are connected to the SECaaS MEH and requesting services simultaneously. Thus, our approach is to launch ME Apps to serve more than one UE App and managing the service operations following the Service Function Chaining (SFC) concept.

1) Security Orchestrator (SO)

The service requests are handle by the SO acting as the MEP for this MEH. Once a service request is approved by the SO, it will create and configure a container with the approved service or utilize the services of an existing container. SO is monitoring the resource utilization of the Virtualization Infrastructure (VI) in the prospect of optimizing the efficiency.

2) SECaaS Services

Our SECaaS concept offers three distinct services to the MEC subscribers. They are; Intrusion Detection and Prevention as a Service (IDPaaS), Authentication as a Service (AaaS), and Secure Transmission Channel as a Service (STCaaS). Under IDPaaS, different well-known Intrusion Detection and Prevention Systems (IDPSs) are operated and offered to the consumer with their strengths and weaknesses, so that the user is capable of selecting the best service suited for their requirement. Currently, we are experimenting with the IDPSs of Suricata and Snort [21], [22]. In the AaaS directive, MEC edge level SECaaS MES is handling the authentication of any application or service desired by the subscriber as a Trusted Third Party (TTP). Thus, cloud based services are validated for the subscriber while user credentials are conveyed and verified to the cloud service by the SECaaS MES. In STCaaS, SECaaS is creating a secure tunnel between the UE and the third party service provider facility at the edge or at a distant location. In this initiative, an entire security protocol is engaged in securing the communication channel.

3) Security Analyzer (SA)

SA, operated as a ME App is acquiring and storing security related statistics and credentials within the system. All the red flags drawn from the IDPaaS instances are gathered and conveyed to the SECaaS centralized server for updating their signature profiles and defence strategies. In addition, threat assessment and prediction constructs are executed at the SA with the gained insights from SECaaS centralized servers. The links to verify the user credentials (i.e. from an external database of authentication credentials) are contained in the SA. These links are viable for AaaS and STCaaS services. Moreover, performance statistics of all security service are recorded in SA.

4) Dockerized Environment

Each security service has its own dockerized image contained in the internal registry of the MEH docker environment. Updated images are conveyed to the internal registry from the external docker server (i.e. Docker hub). Pulling, running, and building docker images are automated within the SO function. Security services are granted with a distinct TCP port number for identifying the service throughout the entire MEC platform. This was attained by performing port forwarding feature of docker containers to the host VM.

IV. Prototype Testing Environment

Fig. 2 illustrates the testing environment emulated for a functioning MEH as a Ubuntu VM under VirtualBox 6.0 hypervisor. The host machine inherits the specifications of core i7 2.50 GHz CPU with 12 GB RAM as specified in TABLE Ia. Both Snort and Suricata docker containers were tested for their performance with the network traffic streams emulated via tcpreplay 4.3.1. Mainly there are two parameters that are vital for the security functions. They are, percentage of dropped packets without processed by the IDS (denoted as d) and percentage of alerts notified by the specific tool (denoted as a). In addition, CPU utilization (denoted as p) and RAM usage (denoted as r) were recorded to measure the performance of each container. The experiments were conducted in different scenarios. In order to emulate the tests, a pcap file called malware_exec.pcap with more than 800,000 packets were employed [23]. In this file 50% of the packets included malware content.

TABLE I: Specifications and Configurations of the Prototype Testing Environment

(a)

Host PC Specifications	
CPU	i7 2.50 GHz
RAM	12 GB
OS	Windows

(b)

VM Configurations	
CPU Cores	2 GB RAMs
1	2
2	2
3	2
4	4

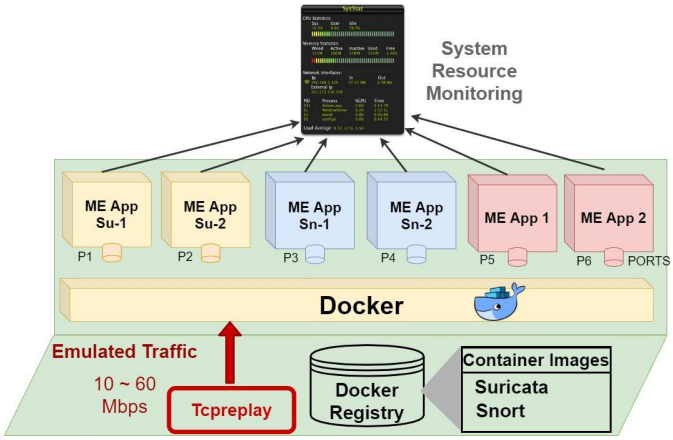


Fig. 2: Prototype MEH Platform

A. Parallel / Simultaneous Operation of Different Services

TABLE II: Comparison of Suricata and Snort Performance in Simultaneous Operation

Factor	Suricata	Snort
CPU Utilization	9.48	81.10
RAM Usage	19.23	33.21
Packet Drops %	33.1	0
Alerts per Packets %	32.9	0

In order to validate our argument with regard to parallel operation of docker container based security instances, testing the simultaneous operation of Suricata and Snort is conspicuous. Thus, TABLE II tabulates the varied parameters of Suricata and Sort executed on the same environment. The observations suggest Suricata is performing efficiently than Snort in terms of alerts. Thus, for out next testing scenarios, Suricata is being considered.

B. Varying Data Rate of the Traffic Stream for a Single Suricata Instance

Fig. 3 depicts the variation of drop percentage and alerts per total packets percentage. As expected d increases while a

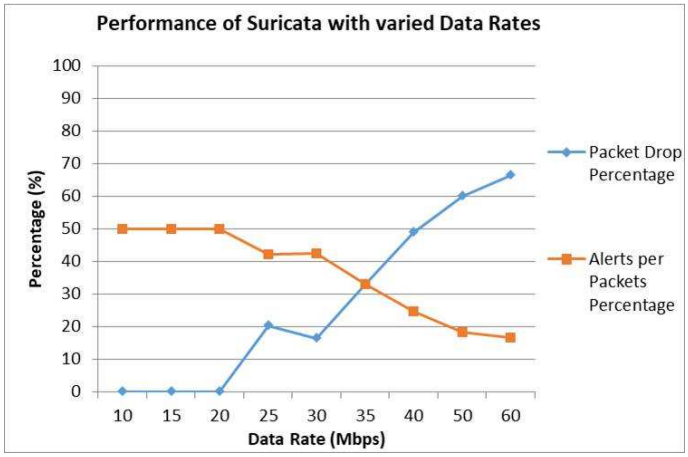


Fig. 3: Suricata performance when traffic flow data rates are varying

degrades with improving data rates due to the fact that Suricata instance fails to read and process the traffic flow. According to the graph, 35 Mbps observed to be the moderate value that could be considered for the experiments followed.

C. Variation of VM Resources for a Single Suricata Instance

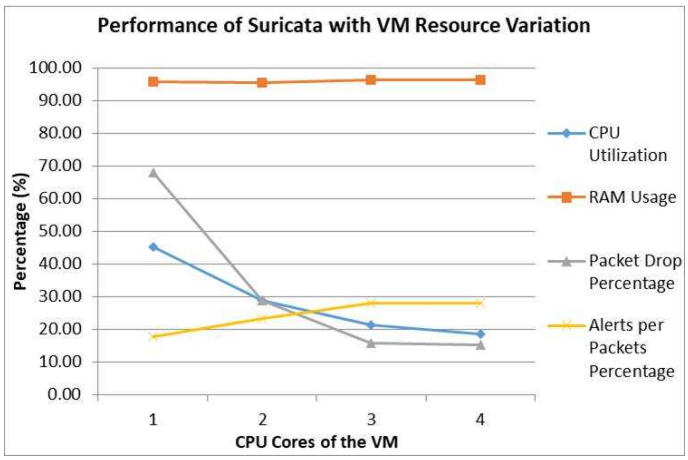


Fig. 4: Suricata performance with varied VM resources

Resources availability of the VM is vital for the outcome of IDS tools been used. Thus, number of CPU cores and RAM allocated for the VM are tested for the configurations depicted in TABLE Ib. p and d values are alleviating with higher resources while a shows a minor increment. r values however, doesn't vary significantly.

D. Multiple Container Processing

In Fig. 5, multiple Suricata containers were tested for determining its performance. The outputting alerts or a are increasing with each Suricata instance operated in parallel. Though, the packet dropping percentage or d is stable between 25%-30%. Only drawback however is the accumulating CPU usage with each Suricata process.

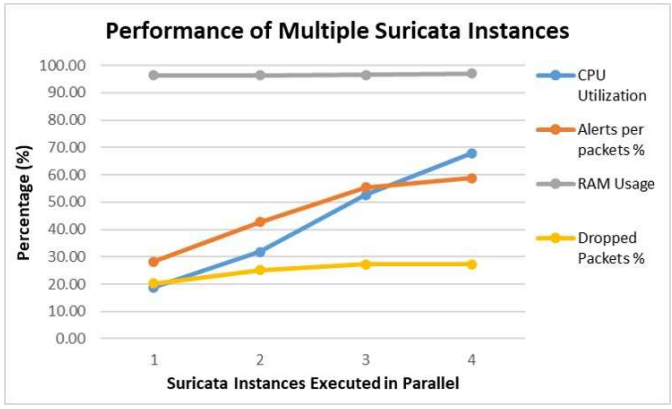
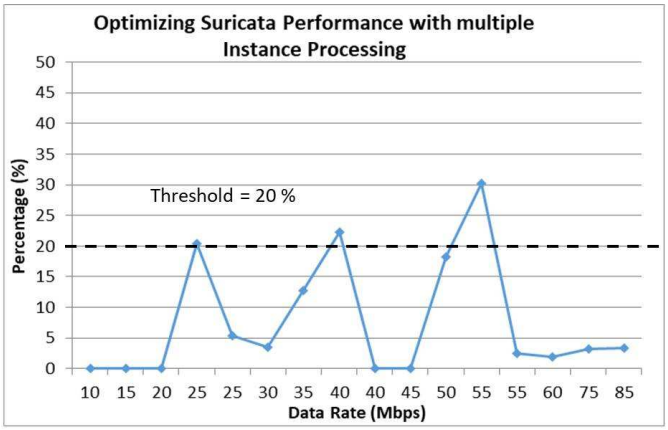


Fig. 5: Performance of Suricata with multiple instances

E. Optimizing Security Service Provisioning

According to the observations from Fig. 3, when the data rate is increasing, d is rapidly accumulating. One of the benefits of launching multiple security instances is its capability to handle higher data rates than the isolated instances. Thus, Fig. 6 is formed by concatenating four different data sets extracted to measure d for different data raters when n number of Suricata instances are operating within a VM bearing n CPU cores. For this simulation, we have considered a threshold of 20% packet drop percentage. Once the data rate goes beyond the 25 Mbps level, threshold on d would be exceeded. Thus, we consider at such an instance, another Suricata container would be launched by the SO for balancing the load. Moreover, with dynamic resource allocation capabilities inherited by modern hypervisors, allocation of additional virtual CPU cores or expanding the virtual limits of the host CPU is plausible [24], [25]. Thus, this simulation is a feasible insight gained through this research that could be achievable via the SO functionality.



Suricata Instances	1	2	3	4
CPU Cores	1	2	3	4

Fig. 6: Simulating packet drop optimization with multiple Suricata instances

V. Discussion

According to the experiments conducted, parallel operation of multiple IDPSs as dockerized instances is plausible. Though, each IDPaaS tool is attributing different performance characteristics. Even the rules and signatures of each tool are differentiated in regards to robustness to various attacks. Thus, number of alerts prompted by the tools are inconsistent for different pcap files. Though, suricata detection accuracy is excelling than its counterparts. Moreover, suricata is capable of load distribution when multiple instances are operating. The simulation presented with the conducted experiments confirms the requirement for an orchestrating entity within each MEH. With the availale technologies, it is possible to balance the IDPaaS based network load among multiple security instances.

VI. Conclusion

The main goal of this paper was to prove the feasibility of Security as a Service (SECaaS) model in MEC (Multi-access Edge Computing) paradigm. It allows to launch multiple security services simultaneously at the egde of the network by employing virtualization technologies. We have followed MEC architecture to form the edge platform and launched multiple security services at the edge successfully. The proposed architecture has the ability to dynamically optimized security services by adapting resources and also adding or removing additional instances of security services to accommodate the dynamic traffic profiles. The methodologies and techniques adapted would be valuable for telecommunication, cloud, and security service providers to enhance their service models in order to cater an extended consumer base with improved and guaranteed quality. In the future, we intend to develop the AaaS and STCaaS services with proper orchestration functioning to the proposed MEC based edge platform.

Acknowledgement

This work is party supported by European Union in RESPONSE 5G (Grant No: 789658) and Academy of Finland in 6Genesis Flagship (grant no. 318927) projects.

References

[1] R. Roman, J. Lopez, and M. Mambo, "Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security Threats and Challenges," Future Generation Computer Systems, vol. 78, pp. 680–698, 2018.

[2] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on Multi-Access Edge Computing for Internet of Things Realization," IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 2961–2991, 2018.

[3] M. Liyanage, I. Ahmad, A. B. Abro, A. Gurtov, and M. Ylianttila, A Comprehensive Guide to 5G Security. John Wiley & Sons, 2018.

[4] J. Pan and J. McElhannon, "Future edge cloud and edge computing for internet of things applications," IEEE Internet of Things Journal, vol. 5, no. 1, pp. 439–449, 2017.

[5] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration," IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1657–1681, 2017.

[6] M. Liyanage, A. Braeken, P. Kumar, and M. Ylianttila, IoT Security: Advances in Authentication. John Wiley & Sons, 2020.

[7] P. Ranaweera, A. D. Jurcut, and M. Liyanage, "Realizing multi-access edge computing feasibility: Security perspective," in 2019 IEEE Conference on Standards for Communications and Networking (CSCN). IEEE, 2019, pp. 1–7.

[8] W. Chen, S. Sharieh, and B. Blainey, "A security-as-a-service solution for applications in cloud computing environment," in Proceedings of the Communications and Networking Symposium. Society for Computer Simulation International, 2018, p. 4.

[9] M. Hawedi, C. Talhi, and H. Boucheneb, "Security as a service for public cloud tenants (saas)," Procedia computer science, vol. 130, pp. 1025–1030, 2018.

[10] 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.

[11] M. Hussain and H. Abdulsalam, "Secaas: security as a service for cloud-based applications," in Proceedings of the Second Kuwait Conference on e-Services and e-Systems. ACM, 2011, p. 8.

[12] H. Al-Aqrabi, L. Liu, J. Xu, R. Hill, N. Antonopoulos, and Y. Zhan, "Investigation of it security and compliance challenges in security-as-a-service for cloud computing," in 2012 IEEE 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops. IEEE, 2012, pp. 124–129.

[13] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," IEEE Internet of Things Journal, vol. 5, no. 1, pp. 450–465, 2018.

[14] S. Kekki et al., "MEC in 5G Networks," ETSI White Paper #28, vol. 28, no. 28, pp. 1–28, 2018, last accessed 16 June 2019. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf

[15] Y. Khettab, M. Bagaa, D. L. C. Dutra, T. Taleb, and N. Toumi, "Virtual security as a service for 5g verticals," in 2018 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2018, pp. 1–6.

[16] A. Boudi, I. Farris, M. Bagaa, and T. Taleb, "Assessing lightweight virtualization for security-as-a-service at the network edge," IEICE Transactions on Communications, vol. 102, no. 5, pp. 970–977, 2019.

[17] A. Sforzin, F. G. Mármol, M. Conti, and J.-M. Bohli, "Rpids: Raspberry pi ids—a fruitful intrusion detection system for iot," in 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld). IEEE, 2016, pp. 440–448.

[18] S. Tripathi and R. Kumar, "Raspberry pi as an intrusion detection system, a honeypot and a packet analyzer," in 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS). IEEE, 2018, pp. 80–85.

[19] ETSI, "Mobile Edge Computing (MEC) Framework and Reference Architecture," ETSI White Paper #3, 2016, last accessed 16 May 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf

[20] B. I. Ismail, E. M. Goortani, M. B. Ab Karim, W. M. Tat, S. Setapa, J. Y. Luke, and O. H. Hoe, "Evaluation of docker as edge computing platform," in 2015 IEEE Conference on Open Systems (ICOS). IEEE, 2015, pp. 130–135.

[21] K. Nam and K. Kim, "A study on sdn security enhancement using open source ids/ips suricata," in 2018 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2018, pp. 1124–1126.

[22] H. M. Elshafie, T. M. Mahmoud, and A. A. Ali, "Improving the performance of the snort intrusion detection using clonal selection," in 2019 International Conference on Innovative Trends in Computer Engineering (ITCE). IEEE, 2019, pp. 104–110.

[23] A. D. Pinto. (2019, Jan) Tricotools. [Online]. Available: https://github.com/NozomiNetworks/tricotools

[24] T. Miao and H. Chen, "Flexcore: Dynamic virtual machine scheduling using vcpu ballooning," Tsinghua Science and Technology, vol. 20, no. 1, pp. 7–16, 2015.

[25] J. Park, D. Lee, B. Kim, J. Huh, and S. Maeng, "Locality-aware dynamic vm reconfiguration on mapreduce clouds," in Proceedings of the 21st international symposium on High-Performance Parallel and Distributed Computing. ACM, 2012, pp. 27–36.