5G ExPerimentation Infrastructure hosting CloudnativE Netapps for public proTection and disaster RElief

Henriques, J., Rosa, L., Gomes, A., Cordeiro, L., Apostolakis, K. C., Margetis, G., Stephanidis, C., Anastasi, M.-A. R., Skoufis, C., Siokis, A. & Ramantas, K. (2021, September). The 5G-EPICENTRE Approach for Decreasing Attack Surface on Cross-Testbeds Cloud-native 5G Scenarios. In *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)* (pp. 7-12). IEEE.

DOI: 10.1109/MeditCom49071.2021.9647599

The 5G-EPICENTRE Approach for Decreasing Attack Surface on Cross-Testbeds Cloud-native 5G Scenarios

Joao Henriques*, Luis Rosa*, Andre Gomes*, Luis Cordeiro*
Konstantinos C. Apostolakis †, George Margetis †, Constantine Stephanidis †‡
Maria-Andrea R.Anastasi §, Christos Skoufis §
Apostolos Siokis ¶, Kostas Ramantas ¶
*Onesource, Coimbra, Portugal, {joao.henriques,luis.rosa,gomes,cordeiro}@onesource.pt
†Institute of Computer Science, Foundation for Research and Technology, Hellas, Heraklion, Greece,
{kapostol,gmarget,cs}@ics.forth.gr
‡Department of Computer Science, University of Crete, Heraklion, Greece
§eBOS Technologies Ltd., Nicosia, Cyprus,{mariaa,christoss}@ebos.com.cy
¶Iquadrat Informatica S.L, Barcelona, Spain, {a.siokis, kramantas}@iquadrat.com

***Abstract*—The 5G-EPICENTRE EU-funded project proposes mission-critical service and application experimentation in federation, adopting a "testbed of testbeds" approach in which different 5G-based platforms are intelligently combined and calibrated from a single control point. This cross-testbed concept embraced in the 5G-EPICENTRE project, together with the transition of 5G technologies into a Cloud-native environment pose numerous challenges, including an increased attack surface and various security concerns such as how to enforce security policies at multiple levels across the entire infrastructure. In that sense, first, this paper provides an overview of such security challenges and a review of the methodologies discussed in the literature to decrease the attack surface in those complex scenarios. Later, this paper presents the 5G-EPICENTRE security approach and an early version of a security framework which considers the usage of security by design techniques, network and container-level isolation strategies and the usage of the service mesh design pattern, all of them key elements to allow to secure the overall infrastructure and monitor, mitigate and respond to security incidents.**

***Index Terms*—Cybersecurity, Heterogeneous Cross-Testbeds, Cloud-native, 5G**

I. Introduction

Toward realizing the 5G vision, future network applications will be empowered to reshape the network by taking full advantage of lightweight virtualization technologies. In such a landscape, security vulnerabilities for the entire network will inevitably increase, since utilization of the network edge by means of orchestrating and placing containerized network functions significantly increases the size of the network, and with it the overall attack surface. This not only compromises application security, but further exposes the whole network to potential threats, leaving the architecture vulnerable to malicious attacks. This is especially true in the case of containers, which are appealing attack vectors for hackers due to the various sensitive data contained within. For these reasons, security must be applied throughout the whole lifecycle of the network functions and should be seen as a shared responsibility (e.g. in the same way network administrators must protect 5G network core elements, developers must also be aware and protect their containers from security vulnerabilities).

The 5G-EPICENTRE EU-funded project [35] proposes mission-critical service and application experimentation in federation, adopting a "testbed of testbeds" approach in which different 5G-based platforms are intelligently combined and calibrated from a single control point. Such architecture, and especially the fact that it targets the Protection and Disaster Relief (PPDR) vertical, needs to address security at multiple levels. Hence, the project aims to configure secure network policies to deal with the increased attack surface resulting from the shift toward edge network function containerization, and the inevitably larger network size. The project plans to achieve this by imposing per-program restricted access profiles at container-level to reinforce isolated execution, while employing a service mesh concept to more efficiently address security considerations. Moreover, 5G-EPICENTRE will integrate security design into its component architecture through a horizontal, cross-layer privacy and security framework, featuring mitigation mechanisms at each architectural layer for dealing with the significantly larger attack surface.

The remainder of this paper is organized as follows. Section II provides an overview of the 5G security and the attack surface. Section III presents the related work. Section IV introduces the 5G-EPICENTRE project and presents the key security components and strategies proposed in the project. Section V concludes the paper.

II. 5G Security and Attack Surface

The latest advances of 5G technologies and concepts, such as the "softwarisation" and virtualisation network functions, present a more challenging scenario from an orchestration standpoint. Inevitably, from a security perspective, this also

means an increased attack surface and new challenges on how to secure the entire infrastructure. As new 5G capabilities are introduced, new types of threats emerge demanding new approaches to security [19]. For instance, Network Slicing (NS), one of the key characteristics discussed in 5G, presents additional complexity and security concerns on how to properly ensure their isolation due to the number of involved components, legacy interworking, and configuration risks. Three major attack scenarios for 5G network slicing were uncovered by AdaptiveMobile [9] including user data extraction, Denial of Service (DoS) against another network function, and access to a network function and related information of another vertical. These attack scenarios specifically focused on network slicing, describe how to gain access to resources of another slice, and how to perform a DoS attack on another slice. They also explain how to extract user-specific information like a location from another slice. Current approaches and technologies are not mitigating such attacks. Ordonez et al. [2] also highlighted the lack of in-built auditability in SOL011 and SOL005 5G architecture interfaces, which makes the corresponding NFV orchestrator (NFVO) exposed interfaces sensitive points in terms of security. Moreover, the attack surface is not only limited to the network functions and a single deployment host but also extends to a large number of nodes. Multiple and heterogeneous domains bring additional complexity and a wider attack surface [21]. On the other hand, the increasing adoption of Cloud-native architectures and the microservice paradigm in the telecommunication sector has allowed decoupling classical monolithic Network Functions Virtualization (NFV), previously deployed in purpose-built hardware, into multiple smaller services running on the top of Virtual Machine (VM)s and containers. Such Cloud-native oriented approach allows to better fulfil the requirements of different 5G service types such as eMBB (enhanced Mobile Broadband), mMTC (massive Machine Type Communications) or URLLC (Ultra-Reliable Low-Latency Communication). Nevertheless, despite the numerous benefits (e.g. increased modularity and flexibility), a Cloud-native and microservice approach results in a larger and more complex infrastructure, which inevitably increases the attack surface at multiple levels. The increased number of connections between micro-services raises new risks of man-in-the-middle attacks spread over the infrastructure, so traffic authentication and authorization between services are vital concepts. The increased number of components might also led to misconfigured, and thus, vulnerable assets. For instance, a recently disclosed backdoor [32] leverages misconfigured Docker API ports to infiltrate Docker servers and later execute malware on the victim's infrastructure. Nowadays, those services might extend far from the traditional on-premises deployments. As we continue to scale up the number of microservices, it is paramount to build strategies to cope with increasingly Cloud-native environments and have the means to monitor the complex mesh resulting from all the microservice communications. Breaking up those traditional monolithic network functions into microservices, often deployed in different nodes and composed of multiple operating systems, programming languages and third-party libraries is by itself an open challenge, not only from an architectural standpoint but from a security point of view. For instance, how to know what is running, how to roll out new service versions and how to monitor and secure all those microservices [18]. Moreover, the application of security patches to containers presents additional challenges - they are usually considered as immutable, meaning that any reconfiguration or update involves rebuild and redeploy the container [37]. Also, the dynamic nature of container-based solutions increases the difficulties in the detection and managing application security vulnerabilities. Vulnerability scanners and manual processes are not suitable to be used in the context of cloud-native environments [36]. Security teams are overwhelmed by alerts, making it difficult to prioritize vulnerability remediation. Therefore, fully automated run-time security is essential to the future of vulnerability management and DevSecOps.

Furthermore, Kubernetes, a widely adopted solution to orchestrate containers, which is expected to be increasingly relevant in 5G deployments in the near future, introduces itself additional complexity and security risks. Understanding and managing all the Kubernetes configurations and networking policies is critical from a security perspective. Finally, 5G-based scenarios composed of several heterogeneous cloud-native domains, such as the ones considered in the 5G-EPICENTRE project also creates new security challenges due to the underlying increase in size and complexity.

III. Related Work

This section brings up the literature review, similar works and approaches aiming to reduce the security risks and attack surface in consequence of the complexity raised by the Cloud-native 5G scenarios.

Vale et al. [33] provided a systematic review of the adopted security mechanisms for microservice-based systems by examining 26 papers published from November 2018 until March 2019. Yu et al. [34] presented a survey related to security risks of microservices-based fog applications and discusses the security concerns of containers, data, permissions and network security. The Cloud Native Computing Foundation (CNCF) [4] has been focused on the complexities around security covering all Cloud-native landscape from the full lifecycle of development to compliance. CNCF states that Cloud-native development must be modelled into distinct lifecycle phases, including development, distribution, deployment, and runtime. Such an approach identifies and secures workloads to meet the scale needs of Cloud-native applications while accommodating constant flux. Nassif et al. [3] provided a systematic literature review of Machine Learning (ML) and Cloud security methodologies and techniques which were categorized into three main research areas including types of Cloud security threats, ML techniques, and performance outcomes. Service mesh architectures may provide additional capabilities to orchestration for handling service-to-service communications in order to cope with the inherent complexity topology of services

without imposing changes on the workload software itself [16]. Some of available service mesh platforms include Istio [10], Linkerd [12], Amazon App Mesh [13], and Airbnb Synapse [11]. Istio has a very active community and the CNCF-accepted project Linkerd provides support for the fundamental features. Additional alternatives exist, including OpenShift Service Mesh by Red Hat, Consul Connect, Kuma, Maesh, ServiceComb-mesher and Network Service Mesh. Li et. al. [20] summarized service mesh approaches to overcome the complexity related to microservices applications by introducing a dedicated infrastructure layer without imposing modification on the service implementations. Design supporting high performance, adaptability and high availability are presented as challenges to achieve the vision of a service mesh. They also have identified the research opportunities and a comparison between the available service mesh technologies. Dab et. al. [17] proposed a service mesh traffic steering solution of cloud-native functions for 5G, while considering the network state of the underlying NFV infrastructure providing the missing Kubernetes networking capabilities. An optimized network-aware load balancing strategy was proposed to reduce end-to-end latency and deployment time. Kang et. al. [28] explored a protected coordination scheme for service mesh, by encrypting all the traffic among application tenants. The authors split the monitor and control traffic the data traffic, followed by encrypting this control traffic. Miller et. al. [30] enforced network workflows to prevent data exposure. They developed an infrastructure using the isolation provided by a microservice architecture, to enforce owner policy. Hussain et. al. [31] proposed an automated intelligent association model of new APIs to service mesh using ML.

A. Security Standards/ Best Practices

Embracing containers, as one of the central transformations in the Cloud-native environment, requires new security best practices. Security standards provide the best practices and mechanisms enhancing the chances to stop attackers and defend against their threats. The need for standardization of the high number of technologies of the 5G puzzle required the involvement of specialized groups beyond the traditional big players. NIST Application Security Container Guide [5], Center for Internet Security (CIS) [7], NIST Security Strategies for microservices [8], and OpenSCAP [7] explain the security concerned with container technologies and make practical recommendations when planning, implementing and maintaining containers. Chandramouli et. al. [29] contributed to guidance on security strategies using service mesh architecture for implementing core features of microservices, as well as countermeasures for microservices-specific threats.

Security-by-Design (SBD) is a security approach aiming to incorporate security and privacy into the implementation process of a software solution. It integrates security into every step of a project implementation lifecycle, seeking to minimize vulnerabilities and reduce the attack surface by designing [22]. The approach aims to minimize risks of threats and vulnerabilities starting from the start of the development lifecycle, all the way to the end. The lifecycle embodies the different phases in the development beginning with analysis, moving forward to the planning, design, development, testing and ending at deployment and maintenance of a software solution. Most organisations often acknowledge that security should be an important consideration when developing systems however, business performance and cost often precede security. It is satisfying to see that awareness is raised on security issues but instead of security considerations throughout the lifecycle, companies focus on applying security practices only at the initiation of the development or in the final design. This practice often affects the efficient application of security on the final product. Evidently, what is required is integrating security at every step from initiation to design and development and then to deployment and maintenance, as an effective way to protect against cyber threats. This sequential approach is what is called the SBD approach.

A common approach to address security challenges that may appear in the development lifecycle, developers and the rest involved parties (Project Managers, Security Officers and System Administrators) make use of model-based procedures following specific security guidelines. Unified Modeling Language (UML) security stereotypes aim to guide developers by annotating vulnerable model parts and allow the automatic security test case generation [23]. Specifically, security modelling is used to define mechanisms that satisfy security criteria such as confidentiality and integrity [24]. The use of UML models essentially represents “misuse” cases or “what can go wrong” during the SDLC and what to do to fix it. UML is a modelling language that offers model-based security engineering and while UML helps engineers in designing software, it generally lacks security features. UML extensions such as UMLsec and SecureUML address security features and allow consideration of security in the seven parts of the software development lifecycle.

SecureUML is a modelling language extension used to integrate information relevant to access control into application models [26]. SecureUML meta-model uses tags like User, Role, and Permission and the relationships between them. Protected resources are expressed using standard UML elements (concept of Model Element). UMLsec is a security specification language or, otherwise, a UML profile extension using stereotypes, tagged values and constraints allowing secure system development by evaluating UML specifications for weaknesses in design [27]. Threat scenarios are specified based on adversary strengths, while specifications of a threat are related to the adversary’s actions. Moreover, constraints specify security requirements while the actors, only perform actions to which they were assigned appropriate rights to. Comparing the two models,

DevOps is the approach encompassing a set of cultural values together with the necessary tools and practices that move the step of continuous development of software to the production environment [25], linking the development team to the operations team. SecDevOps implements the SBD principle by using automated security review of code and automated

application security testing. It is the process of integrating secure development best practices and methodologies into development and deployment processes proving to be highly relatable to the SBD approach as described above.

B. Other EU-5G Related Projects

5G-PPP Phase 3 called for proposals on 5G-Innovations for verticals with third party services and smart connectivity beyond 5G with two calls under the Horizon2020 Framework Programme. ICT-41-2020 projects investigate innovative solutions to facilitate operations in the above-mentioned verticals by providing secure services. The 9 selected projects (5GASP, 5G-EPICENTRE, 5G-ERA, 5G-IANA, 5GINDUCE, 5GMediaHUB, evolved5G, Smart5Grif and VITAL5G) touch on verticals such as PPDR, Industry 4.0, Transport & Logistics, Automotive mobility/industry, eHealth, Smart Energy and Media. They aim to exploit software functions from experimental facilities, to be used openly by SMEs and developers wishing to test their applications in the context of specific vertical use cases. Additionally, the projects will create 5G open-source repositories for wide use while also provide input for the development of standards. Further EU-5G related projects investigate Smart Connectivity beyond the scope of 5G networks. Another 9 projects received funding under the "ICT-52-2020: Smart Connectivity Beyond 5G" aiming to surpass the challenge of going beyond 5G capabilities available in 2020. Moreover, the projects support the initiative to move from hardware to software with expected impact on latency, scalability connectivity, network/service management with the provision of advanced solutions.

IV. 5G-EPICENTRE

The previous section provided an overview of the related work proposed in the literature to address the security challenges of Cloud-native 5G scenarios. This section presents the 5G-EPICENTRE main objectives, use cases and the proposed security approach that will be further researched in the context of the project. The 5G-EPICENTRE project proposes to federate multiple 5G platforms using a Cloud-native and microservices-oriented approach.

A. Objectives

In order to substantiate the 5G-EPICENTRE vision, overall activities will target the following objectives:

- To build an end-to-end 5G experimentation platform specifically tailored to the needs of the public safety and emergency response market players.
- To pilot 5G systems in PPDR-based trials, successfully demonstrating 5G-EPICENTRE onboarded apps as a crucial communications accompaniment to public safety mission-critical communication technologies.
- To cultivate a '5G Experiments as a Service' model, which will enable developers and SMEs to experiment with PPDR applications in parameterized, easily repeatable, and shareable environments.
- To facilitate automation, continuous deployment and multi-access edge computing supported by containerized network functions so as to reduce service creation time and time-to-market for 5G solutions.
- To leverage Artificial Intelligence (AI) for achieving cognitive experiment coordination and lifecycle management, including dynamic 5G slicing, application awareness and insightful ML-driven analytics.
- To implement impact-driven dissemination, standardisation and exploitation.

B. Use Cases and Challenges

The 5G-EPICENTRE project oversees the platform's secure interoperability capabilities beyond vendor-specific implementation. It includes the following set of public security and disaster management use cases:

- Multimedia Mission Critical Communication and Collaboration Platform.
- Multi-agency and multi-deployment mission-critical communications and dynamic service scaling.
- Ultra-reliable drone navigation and remote control.
- IoT for improving first responders' situational awareness and safety.
- Wearable, mobile, point-of-view, wireless video service delivery.
- Fast situational awareness and near real-time disaster mapping.
- Augmented Reality (AR) and Artificial Intelligence (AI) wearable electronics for PPDR.
- AR-assisted emergency surgical care.

C. Proposed Approach

The overall 5G-EPICENTRE architecture is segmented into a multi-layered approach (front-end, back-end, federation and infrastructure layer). The front-end layer includes the processes supporting the interaction between the platform and PPDR solution providers. The back-end layer comprises the functional components of the platform. The federation layer comprehends the cross-testbed orchestration of network services and resources to ensure an optimal experiment environment. Finally, the 5G testbed infrastructural elements of each of the federated testbeds compose the infrastructure layer. Orthogonal to those layers, and to address the security challenges of the envisioned and highly complex 5G scenarios, an Holistic Security and Privacy Framework (HSPF) was also proposed.

Figure 1 depicts an early design of the proposed security framework, being composed of three key elements: a policy engine, a security engine and an AI engine. The policy engine centralizes the configuration of the policies at the network and container levels. For instance, applying policy profiles at the container level, can be used for multi-tenant environments, manage the resource access or enforce process kernel restrictions enhancing its isolation. The security engine comprises the protection to the underlying host OS, where

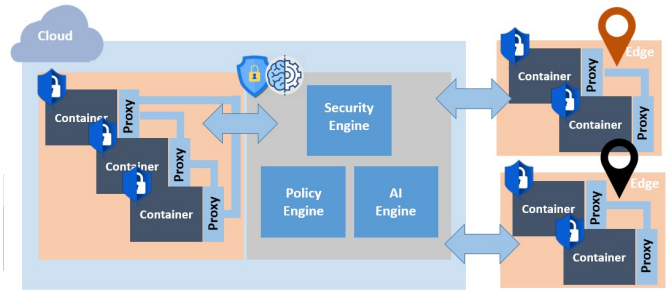

Fig. 1. 5G-EPICENTRE Security Framework Proposal

the containers run, access control and authentication mechanisms such as single sign-on, network traffic encryption and container isolation techniques. Finally, an AI engine will be further researched to assist security and policy enforcement. For instance, the AI engine might help to identify anomalous streams based on the observability of the network traffic and support the enforcement of automated response policies and actions. Figure 2 summarizes the list of features that will be further considered in the proposed security framework.

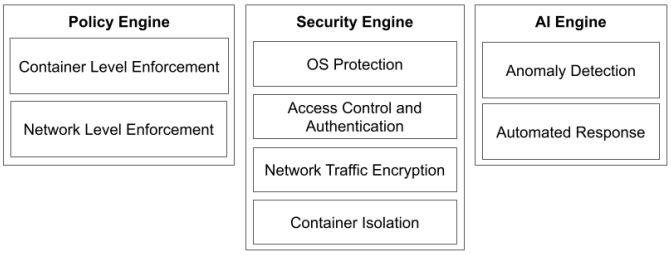

Fig. 2. Security Framework Features

As discussed before, the security of Cloud-native scenarios highly rely on the enforcement of security mechanisms at the container-level such as i) preventing containers from having root access, ii) controlling communications between containers, iii) restricting permissions and access to only what's necessary for the applications to function, iv) ensuring containers are free of known vulnerabilities. In this sense, and amongst others, different container-level isolation mechanisms such as Docker seccomp, namespaces, cgroups, process restrictions, device, and file restrictions will be further investigated.

The classical in-depth network security approaches, such as perimeter firewalls cannot be easily applied to Cloud-native scenarios, such as the ones envisioned by 5G-EPICENTRE container-based approaches which require more fine-control regarding the network communications between all the different containers.

In the literature, beyond traffic management, circuit breaking and service discovery, the service mesh concept is also discussed as a Cloud-native approach to bring additional security features. It allows decoupling the inherent complexity in the implementation of security features from the existing applications to be put in a service proxy. These proxies, which have access to network traffic, are, therefore, strategic components for supporting the deployment of such security mechanisms.

This concept, which will be further investigated in the context of the proposed approach to support different security capabilities, including logging API traffic, observability tagging, network traffic encryption, authentication, and authorization. Beyond the centralized management of the policies, it can also be used to support policy enforcement between different edge/cloud network traffic in Cloud-native 5G scenarios. The service mesh concept mainly relies on the usage of network proxies (the sidecars), deployed together with each container instance, for intercepting and mediating the network traffic among microservices. This way, those sidecar proxies can be leveraged to implement a different kind of network-based policies, attaining the involved context (e.g. to evaluate whether a given request is authorized or not). On a 5G hybrid edge/cloud environment those concepts can also be used to independently apply policies on the different domains.

Finally, in addition to the proposed framework, the 5G-EPICENTRE project will foster a SBD philosophy which aims to incorporate security and privacy concepts from the early stages of the design and development lifecycle. Concepts and approaches such as UMLsec, SecureUML and SecDevOps will be considered during the design of the overall 5G-EPICENTRE architecture.

V. Conclusions

Increasing complex 5G scenarios, such as the ones brought by the 5G-EPICENTRE project designed on the top heterogeneous Cloud-native cross-testbeds, poses several challenges from a security standpoint. They demand specifically tailored security approaches for security orchestration, analytics and automation. This paper provides an overview of these security challenges and a review of the methodologies discussed in the literature to decrease the attack surface in such complex scenarios. Then, this paper presented the key research topics that will be further explored in the context of 5G-EPICENTRE and proposes an early version of a security framework to address those challenges. Our approach was conceived as a Cloud-native framework to support data observability and traffic filtering by enforcing different types of policies at the application and network levels. To accomplish that, and amongst others, we proposed the usage of security by design techniques, network and container-level isolation strategies and the usage of the service mesh design pattern at the core of the security framework, all of them key elements to allow to secure the overall infrastructure and monitor, mitigate and respond to security incidents.

Acknowledgment

This project has received funding from the EU's Horizon2020 innovation action program under Grant agreement No 101016521 (5G-EPICENTRE project). This paper reflects only the authors'view and the Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] Yala, L., Iordache, M., Bousselmi, A., & Imadali, S. (2019, October). Testbed federation for 5g experimentation: Review and guidelines. In 2019 IEEE Conference on Standards for Communications and Networking (CSCN) (pp. 1-6). IEEE.

[2] Ordonez-Lucena, J., Tranoris, C., Rodrigues, J., & Contreras, L. M. (2020, June). Cross-domain slice orchestration for advanced vertical trials in a multi-vendor 5G facility. In 2020 European Conference on Networks and Communications (EuCNC) (pp. 40-45). IEEE.

[3] Nassif, A. B., Talib, M. A., Nasir, Q., Albadani, H., & Dakalbab, F. M. (2021). Machine Learning for Cloud Security: A Systematic Review. IEEE Access, Access, IEEE, 9, 20717–20735. https://doi.org/10.1109/ACCESS.2021.3054129

[4] Cloud native security whitepaper, [Online]. Available: https://github.com/cncf/sig-security/blob/master/security-whitepaper/CNCF_cloud-native-security-whitepaper-Nov2020.pdf", last accessed in March 2021

[5] NIST Application Container Security Guide, July 2017. [Online]. Available: https://csrc.nist.gov/csrc/media/publications/sp/800-190/draft/documents/sp800-190-draft2.pdf, last accessed in March 2021

[6] , Openscap, [Online]. Available: https://www.open-scap.org, last accessed in March 2021

[7] Center for Internet Security, [Online]. Available: https://www.cisecurity.org, last accessed in March 2021

[8] , Security Strategies for Microservices-based Application Systems, [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-204/final, last accessed in March 2021

[9] , AdaptiveMobile, (2021), A Slice in Time: Slicing Security in 5G Core Networks

[10] The Istio Team, "Istio: An Open Platform to Connect, Manage, and Secure Microservices." https://github.com/istio/istio, last accessed in March 2021

[11] Airbnb, "Synapse: A Transparent Service Discovery Framework for Connecting an SOA," https://github.com/airbnb/synapse, last accessed in March 2021.

[12] The Cloud Native Computing Foundation, "Linkerd: Production-grade Feature-rich Service Mesh for Any Platform." https://github.com/linkerd/linkerd, last accessed in March 2021.

[13] Amazon Web Services, Inc., "AWS App Mesh", https://aws.amazon.com/app-mesh/, last accessed in March 2021

[14] AliPay, "SOFAMesh: A Solution for Large-scale Service Mesh based on Istio,"https://github.com/alipay/sofa-mesh, last accessed in March 2021.

[15] The Cloud Native Computing Foundation, "Envoy," https://www.envoyproxy.io/, last accessed in March 2021

[16] Buoyant Inc., "The Service Mesh: What Every Software Engineer Needs to Know about the World's Most Over-Hyped Technology," last accessed in MArch, 2021. [Online]. Available: https://buoyant.io/service-mesh-manifesto/

[17] Dab, B., Fajjari, I., Rohon, M., Auboin, C., & Diquélou, A. (2020, June). Cloud-native service function chaining for 5G based on network service mesh. In ICC 2020-2020 IEEE International Conference on Communications (ICC) (pp. 1-7). IEEE.

[18] Google (2019), Welcome to the service mesh era, https://cloud.google.com/blog/products/networking/welcome-to-the-service-mesh-era-introducing-a-new-istio-blog-post-series, last accessed in March 2021

[19] Nokia 5G security. A new approach to building digital trust, 2019, [Online]. Available: https://onestore.nokia.com/asset/206609, last accessed in March 2021

[20] Li, W., Lemieux, Y., Gao, J., Zhao, Z., & Han, Y. (2019). Service mesh: Challenges, state of the art, and future research opportunities. In 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE) (pp. 122-1225). IEEE.

[21] IETF, (2019), [Online]. Available: https://tools.ietf.org/html/draft-bernardos-nfvrg-multidomain-05, last accessed in March 2021

[22] McManus, J. (2018). Security by design: teaching secure software design and development techniques. Journal of Computing Sciences in Colleges, pp. 75-82

[23] Peralta, Karine & Orozco, Alex & Zorzo, Avelino & Oliveira, Flávio. (2009). Specifying Security Aspects in UML Models. CEUR Workshop Proceedings

[24] Mayer N. (2009) Model-based Management of Information System Security Risk. PhD Thesis, University of Namur

[25] Davis, J. and Daniels, K. (2016) Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale. " O'Reilly Media, Inc.", 2016

[26] Matulevicius, R.; Dumas, M. (2010). The 9th Conference on Databases and Information Systems, 2010. Databases and Information Systems: The 9th Conference on Databases and Information Systems; Riga, Latvia; July 57, 2010. Ed. Barzdins, J.; Kirikova, M. Latvia: University of Latvia Press, Riga, Latvia, 171185 https://courses.cs.ut.ee/2010/is/uploads/Main/RBACforUML.pdf

[27] Jurjens, J. (2002) UMLsec: Exending UML for Secure Systems Development. In: Proceedings of the 5th International Conference on The Unified Modeling Language, LNCS, vol. 2460, pp. 412—425

[28] Kang, M., Shin, J. S., & Kim, J. (2019, January). Protected coordination of service mesh for container-based 3-tier service traffic. In 2019 International Conference on Information Networking (ICOIN) (pp. 427-429). IEEE.

[29] Chandramouli, R., & Butcher, Z. (2020). Building secure microservices-based applications using service-mesh architecture. NIST Special Publication, 800, 204A.

[30] Miller, L., Mérindol, P., Gallais, A., & Pelsser, C. (2020). Towards Secure and Leak-Free Workflows Using Microservice Isolation. arXiv preprint arXiv:2012.06300.

[31] Hussain, F., Li, W., Noye, B., Sharieh, S., & Ferworn, A. (2019, October). Intelligent Service Mesh Framework for API Security and Management. In 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 0735-0742). IEEE.

[32] threatpost (2020), Doki Backdoor Infiltrates Docker Servers in the Cloud, [Online]. Available: https://threatpost.com/doki-backdoor-docker-servers-cloud/157871, last accessed in March 2021

[33] Pereira-Vale, A., Márquez, G., Astudillo, H., & Fernandez, E. B. (2019). Security mechanisms used in microservices-based systems: A systematic mapping. In 2019 XLV Latin American Computing Conference (CLEI) (pp. 01-10). IEEE.

[34] Yu, D., Jin, Y., Zhang, Y., & Zheng, X. (2019). A survey on security issues in services communication of Microservices-enabled fog applications. Concurrency and Computation: Practice and Experience, 31(22), e4436.

[35] 5GEPICENTRE (2021). 5G ExPerimentation Infrastructure hosting Cloud-nativE Netapps for public proTection and disaster RElief, [Online]. Available: https://www.5gepicentre.eu, last accessed in March 2021.

[36] Dynatrace (2021).Precise, automatic risk and impact assessment is key for DevSecOps, Global CISO Report, [Online]. Available: https://assets.dynatrace.com/en/docs/report/2021-global-ciso-report.pdf, last accessed in July 2021

[37] Adkins, H., Beyer, B., Blankinship, P., Lewandowski, P., Oprea, A., Stubblefield, A. (2020). Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems. O'Reilly Media.