Chinese Journal of Electronics
Vol.30, No.3, May 2021

Secure Cloud Architecture for 5G Core Network

LI Lingshu[1], WU Jiangxing[1], HU Hongchao[1], LIU Wenyan[1] and GUO Zehua[2]

(1. *PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China*)

(2. *Beijing Institute of Technology, Beijing 100000, China*)

Abstract — Service-based architecture (SBA) is a profound advancement in the novel 5G Core network (5GC). Existing studies show that SBA can benefit from cloud computing to achieve extensibility, modularity, reusability, and openness. It also brings security problems (*e.g.*, hypervisor hijacking, and malware injection). To provide secure 5G services, we propose a service-based cloud architecture called Mimicloud for 5GC based on dynamic and heterogeneous techniques. Mimicloud provides flexible reconfiguration mechanisms to protect containers and eliminate all attack knowledge obtained from adversaries. We use multiple containers to execute crucial services and ensure security with crosscheck. Mimicloud employs heterogeneous components to prevent multiple containers from being breached through the same vulnerabilities. Experimental results show that Mimicloud can effectively strengthen the security of the 5GC. The performance overhead is analyzed in order to demonstrate its scalability.

Key words — 5G, Service-based architecture, Cloud security, Cyber mimic defense.

I. Introduction

The Fifth-generation (5G) of the mobile communication system is built based on Service-based architecture (SBA) and enables granular and flexible design for Virtual network functions (VNFs) of 5GC. In 3GPP Release 15 specification[1], each Network function (NF) is split into a set of self-contained, reusable, independent, and scalable services called NF services. Each NF service exhibits standardized interfaces, and VNF is organized by a series of service invocations. One essential feature of 5GC is to provide on-demand orchestration and distributed deployment, which can be realized by adaptively deploying SBA on cloud infrastructure and implementing NF services with containers. Customers can access NF services directly via a pay-per-use model. To automatically manage and orchestrate containers, many clouds start to adopt Kubernetes, a popular open-source container orchestration system[2].

Deploying VNF in clouds is an effective and popular way to unlock the full potential of 5G networks. However, it should be used securely since 5G will connect every aspect of life. For specific industry scenarios such as governments and hospitals, existing works are still far from adequately providing high-level Service level agreement (SLA) services. The damages are incalculable if the service executor (container) is compromised or tampered by the adversary without being noticed and rectified. The security issue is the main obstacle to prevent many companies and government agencies from working on the cloud. For 5G application scenarios such as telemedicine and autonomous driving, mobile operators are willing to pay extra resources for high security[3].

Recent research reveals two major potential security challenges of 5GC. The first is caused by virtualization technology. Virtualization technology is adopted by 5GC SBA to provide flexible resource allocation, whereas it is vulnerable to Denial of service (DoS), side-channel attacks, hypervisor hijacking, and attacks from underpinned architecture. If the hypervisor vulnerabilities are exploited by the adversary to acquire the root privilege, all the containers allocated on this hypervisor can be compromised easily. Some experienced adversaries can launch a Basic input/output system (BIOS) attack and directly threaten physical servers[4]. Furthermore, in the 5GC multi-tenant coexistence network, adversaries may bypass logical isolation between Dockers via side channels to steal sensitive information (*e.g.*, a password or secret key)[5].

The second is caused by cloud computing. Combining cloud computing with the 5G ecosystem brings security problems like penetration attacks, resource theft, Man-

Manuscript Received Aug. 21, 2020; Accepted Dec. 29, 2020. This work is supported by the National Key Research and Development Program of China (No.2018YFB0804004), the Beijing Institute of Technology Research Fund Program for Young Scholars, and the Natural Science Foundation of China under Grant (No.62002019).

 DOI:10.1049/cje.2021.04.005

in-the-middle (MitM) attack, *etc.* Application processes in traditional Information and communication technology (ICT) systems usually exchange information via internal share storage or internal communication (*e.g.*, socket or remote method invocation). However, cloud-based 5GC adopts many exposed APIs that make it easier for attackers to exploit loopholes and backdoors, stealthy access unauthorized resources, and consume more resources. An attacker can launch various attacks to compromise containers via Operations, administration, and management (OA&M) interfaces[6].

There are two categories of defense approaches currently being adopted in the field of 5GC security. One is conventional cyberspace defense technologies, which test, discover and eliminate potential threats (*e.g.*, the distributed firewalls and Distributed intrusion detection system (DIDS)[7], remote verification and integrity checking[8], and Intel software guard extensions[9]). However, these static defense methods are challenging to defend against increasingly sophisticated and intelligent network intrusion based on unknown hardware and software vulnerabilities. Furthermore, Capital expenditures (CapEx) and Operational expenses (OpEx) are significant barriers for large-scale use in 5G.

The other is the new defense approach, which aims to increase the system's uncertainty and complexity (*e.g.*, Moving target defense (MTD), trusted computing). MTD is widely used in cloud computing to make the system more dynamic, random, and diverse[2]. However, MTD exposes a series of problems such as blindness, high cost, and inability to verify the status. In recent years, Cyber mimic defense (CMD) has emerged as a promising active network defense technology in the cybersecurity landscape[10]. Compare to MTD, CMD is more targeted and presents a more unpredictable defensive behavior in cyberspace[11]. Leveraging the advances in cloud computing to reduce CapEx and OpEx, CMD is suitable for building a high-security 5GC SBA.

In this paper, we propose a CMD-based high-security 5G core network with service-based architecture (Mimicloud), a closed-loop negative feedback control architecture. More precisely, flexible reconfiguration mechanisms can clean the containers and eliminate all the attack knowledge obtained by adversaries. Multiple containers are used to execute the same work to ensure security with crosschecking. Heterogenous components are employed to prevent multiple containers from being breached through the same vulnerabilities. The continuous-time Markov chain is adopted to evaluate the effectiveness of CMD strategies.

The contributions of this study are summarized as follows:

1) We devise a high-security service-based architecture for 5G core networks based on Kubernetes, which covers design, implementation, and monitoring procedures for container development and service acquisition.

2) We present a container reconfiguration mechanism to enhance the security of 5GC, whose availability is evaluated for automatically and economically adjusting reconfiguration strategies.

3) We adopt multiple containers with cross-checking for risk discovery and control, whose effectiveness and performance overhead are evaluated.

4) We use heterogeneous containers cluster to process tenants' requests in parallel and analyze its security effectiveness.

5) We built a container cloud experimental environment based on Kubernetes to analyze proposed approaches. The evaluation results confirm that Mimicloud can achieve high security while paying a 28% extra delay cost.

II. Overview of Mimicloud

To offer high-security 5G NF service, Mimicloud develops Dynamic heterogeneous redundancy (DHR) architecture and corresponding cyber mimic defense strategies in the cloud, as illustrated in Fig.1. DHR architecture, the core connotation of CMD, can significantly reduce uncertain disturbance and various malicious attacks that disrupt containers. All services running on containers are insulated and hidden by the input/output proxy to avoid attackers' direct access. The user's requests will be duplicated and distributed by the input proxy to multiple heterogeneous containers that belong to the same mimic high-security service suites. Each 5G NF service is hosted by heterogeneous containers located on different physical servers. After receiving the request, each container will process it independently and send the result to the adjudicator termed as the Mimic decision (MD) module. MD contrasts and analyzes multiple results based on preset strategy and outputs a result. Container substitution and reconfiguration are conducted by the task schedular and feedback controller. The task scheduler performs proactive substitution, which replaces the running containers periodically; The feedback controller collects status information of containers in real-time and performs reactive substitution, which is triggered by the container's error state. The uncertainty of DHR discourages the adversary from trial and error.

Based on Kubernetes' master-slave work mode, the proposed DHR-based 5GC cloud architecture (Mimicloud) consists of a control node (Master) and multiple compute nodes (Nodes). The task scheduler and the feedback controller in Master communicate with containers in Node through the API server. The container which is responsible for NF service execution is called the executor.

Three CMD-based principal security mechanisms are adopted in the Mimicloud, as illustrated in Fig.2. ① Executor reconfiguration. The task scheduler and feedback controller are deployed in Kubernetes Master to conduct MTD strategies through the API server. They re-select appropriate containers to deploy NF service for satisfying high-security degrees while keeping total performance costs acceptable. ② Multiple executors with crosschecking. The attackers can compromise the container to interrupt the NF service execution and use evading techniques to hide. Therefore, multiple executors with crosschecking are promoted to reduce the risk that a single executor is compromised or broken down. When the Master receives a tenant's request, the task scheduler selects multiple suitable containers to run the NF service. All the execution results are checked by MD, so the malicious container can be found if the execution result is different. ③ Heterogeneous executor pool. Since homogeneous components are vulnerable to the same vulnerabilities, we attempt to exploit the heterogeneity executor pool to enhance security. The heterogeneous containers contain Docker and Ckt; Heterogeneous OSs contain Ubuntu, CoreOs, Fedora; Heterogeneous physical servers contain Intel server and ARM server.

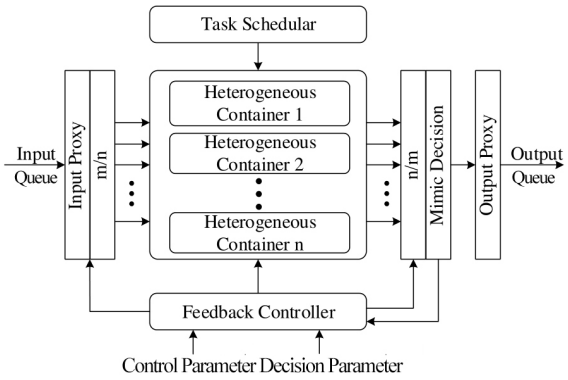

Fig. 1. DHR Architecture

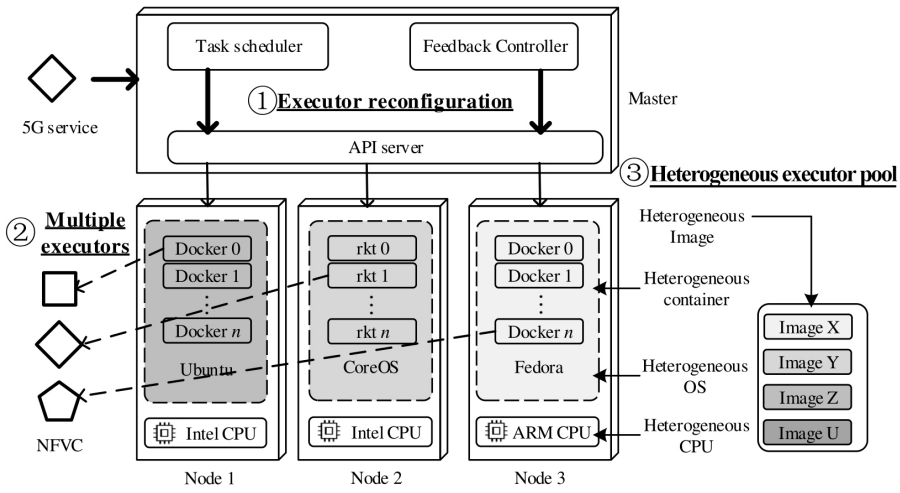

Fig. 2. Mimicloud architecture based on Kubernetes

III. Principal Mechanisms

This section shows three principal mechanisms of Mimicloud in detail: executor reconfiguration, multiple executors witch crosschecking, and heterogenous executor pool.

1. Executor reconfiguration

Executor reconfiguration is a shuffling technique rearranging or randomizing system configurations. It is efficient to improve security by shortening the containers' lifecycle in 5GC. The reconfiguration rate has a positive correlation with the security rate. However, it also brings more cost because containers in reconfiguration cannot provide service. Thus, it is an interesting question to find a reasonable trade-off. Compared with existing work[12], CMD service can share containers with no-CMD public cloud services. The idle resources in off-peak time can be used by regular service. In a nutshell, it is practical and economical.

The task scheduling system acts as a modified M/M/c queue. There are three status of containers: idle, working, and reconfiguration. A container pool consists of C containers, but only idle containers are available to serve incoming requests that arrive at an average rate λ. Under the control of a mimic scheduler, requests will be served by suitable idle containers with an average service time S.

Mimicloud conducts a generic MTD technique to rebuild containers occasionally and independently. Thus, each container handles service requests as well as reconfiguration requests. While a resource is under reconfiguring, it is not available to process service requests. Containers are reconfigured independently of one another at a rate of per time unit. Continuous

time markov chain (CTMC) is adopted to compute the probability distribution of the number of containers under reconfiguring. Subsequently, the distribution is used to determine resource availability and response time.

The container's life cycle consists of the free period and reconfiguration period. Thus, the container's average age is $1/\alpha$. Consider that there are c containers in the container pool. Let m be the average number of containers available for use (*i.e.*, not being reconfigured) and c_r be the average number of containers being reconfigured. Thus, we have $c = m + c_r$. According to Little's Law[13] we obtain $m = X \times (1/\alpha)$.

Where X is the system's reconfiguration throughput, *i.e.*, the aggregate rate at which containers complete their reconfiguration. Let S be the average time for a container to complete the reconfiguration process, including the time of turning off all running transactions, changing its configuration file, shutting down the container, and rebooting it.

2. Multiple executors with crosscheck

To offer high-security service, Mimicloud provides multiple replicas of containers to execute the same work and build a mimic voter middleware to compare the result. The feedback control mechanism works when the multiple results are not the same (reactive substitution), or the request rate exceeds the threshold (proactive substitution). It swaps out an online container with a clean one. As illustrated in Fig.3, Mimicloud builds a security component I/O proxy, which acts as a single queue. All incoming application requests are collected by the I/O proxy and served one after the other. For requests with the high-security requirement, the proxy duplicates the request and distributes these duplicates to different containers.

3. Heterogeneous executor pool

A heterogeneous executor pool is a diversity-based MTD technique employing the deployment of containers with different implementations. The cloud structure of 5GC has security automation control abilities like configuration modifying, vulnerability testing, security audits, and security patching. However, homogeneous redundant containers are competent for fault tolerance but useless for security reinforcement and risk propagation restraint. A single flaw may be manifested throughout the system. Thus, heterogeneous task executors are used in Mimicloud.

A heterogeneous executor can be represented by a multi-dimensional vector. Each element in the vector denotes an independent attribute of the container. For instance, if we choose CPU, OS, container type, and software as representational attributes, an executor A's attribute vector can be represented as (Intel i7, ubuntu 19.04, Docker 17.09, Apache 1.8). Likewise, executor B with different attributes can be represented as (Intel i7, ubuntu 18.04, Docker 17.09, Nginx 1.17). We use a natural number to represent different values under different attribute, then the attribute vectors of server A and B can be expressed as $(1,\ 1,\ 1,\ 1)^{\mathrm{T}}$ and $(1,\ 2,\ 1,\ 2)^{\mathrm{T}}$. For two vectors $X = (v_1, v_2, v_3, v_4)^{\mathrm{T}}$ and $Y = (u_1, u_2, u_3, u_4)^{\mathrm{T}}$, the distance is defined as $d = \sqrt{\sum_{i=1}^{|X|} \alpha_i |v_i - u_i|^2}$, α_i is the security weighting factor set according to the administrator's concern to adjust the weight of different attributes. Different numbers are only used as symbols to represent different types of the same attribute, without mathematical significance (*e.g.*, size and distance). The same number may appear in different rows, and no correlation is identified in different attributes. The difference of a single attribute $|v_i - u_i|$ is defined according to the actual situation. For instance, we can employ the number of same vulnerabilities to quantize $|v_i - u_i|$. A generic maximum heterogeneity selection algorithm randomly selects the first executor. Subsequently, all the latter executors will be selected based on heterogeneity with selected former executors.

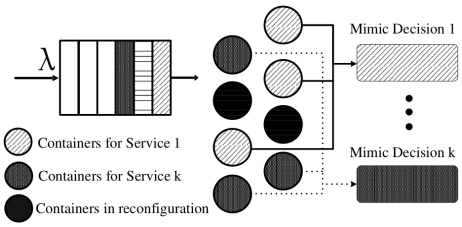

Fig. 3. Task scheduling and decision mechanism

IV. Evaluation

In this section, we first introduce the evaluation setup. Then, we analyze the availability of executor reconfiguration. Lastly, we deploy a capability evaluation of multiple executors and a security evaluation of the heterogeneous executor pool.

1. Evaluation setup

The proposed approaches are analyzed by simulation and experiment. For simulation, the proposed algorithms are simulated in MATLAB R2018a environment in a ThinkPad T40 with Intel Core i5-6500 3.2GHz CPU and 8GB RAM.

In the experiment, we set up a test network containing x86 and ARM architectures. The network consists of 6 servers equipped with a heterogeneous processor (*e.g.*, Intel Xeon CPU E5-2630 v3 processor, Dhyana Hygon C86 7151 processor, GM-FT2000 processor, Huawei Kunpeng 920) and multiple operating systems (*e.g.*, Ubuntu, Centos).

2. Executor reconfiguration

We consider the resource availability of mimic high-security application (application running on several

containers and cooperate based on DHR architecture) on three kinds of cloud system: the conventional system as the baseline (system G); the system allowed the service downgrade to low-security mode, suggesting that the application can work with no less than $\lceil K/2 \rceil$ executors (system D); our proposed system adopting scheduling strategy (system S). Fig.4 shows three systems' availability curves as a function of the reconfiguration rate α for values of the reconfiguration time S equal to 50 and 150 seconds, respectively. It is suggested that the longer time required for a container to rebuild, the lower availability of the system's container pool. Likewise, the higher reconfiguration rate of containers, the lower the availability. When the reconfiguration rate tends to zero, the availability tends to 1 since all resources are available for use. Furthermore, the task scheduling strategy can significantly increase resource availability. Note that under the relative low reconfiguration cost $\alpha \cdot S$, the performance of system S is even better than system D.

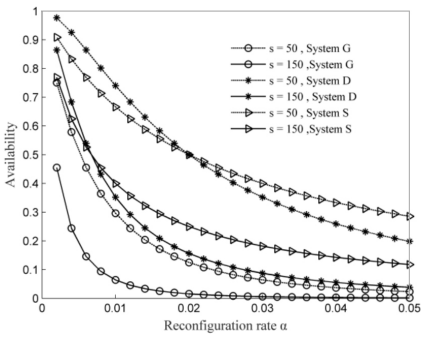

Fig. 4. Relationship between the reconfiguration rate α and availability of different system model

3. Multiple executors

The delay of a web service caused by I/O proxy is analyzed in the experiment network, as shown in Fig.5. Compared to the delay of normal service, the I/O proxy's time overhead is small. The service delays of Mimicloud and conventional systems are illustrated in Fig.6. The high security of Mimicloud is achieved by sacrificing some service performance. Compared to the conventional system, Mimicloud costs about 28% more in time delays. Besides, It can be seen from Figs.5 and 6 that the main overhead comes from multiple executors rather than the I/O proxy. The mimic decision component has to wait for all responses of multiple executors and conduct crosscheck.

4. Heterogeneous executor pool

The security impact of heterogeneous task executors is to be evaluated. A multi-element executor can be represented as $X = (v_1, \ldots, v_n)^{\mathrm{T}}$. $\boldsymbol{v}_i$ denotes an attribute of the container (*i.e.*, CPU type, OS type, Hypervisor type). There are n types of candidate attributes available for the executor. In other words, the executor's heterogeneity is n. For attributes i, there are $m_i (i \in n)$ values available to select. In theory, there are $\prod_{i \in n} m_i$ types of executors at most. Adversaries know that each container must belong to one of them but cannot accurately determine which one it is. Adversaries will randomly select an executor type $Y = (u_1, \ldots, u_n)^{\mathrm{T}}$ as the attack strategy each time.

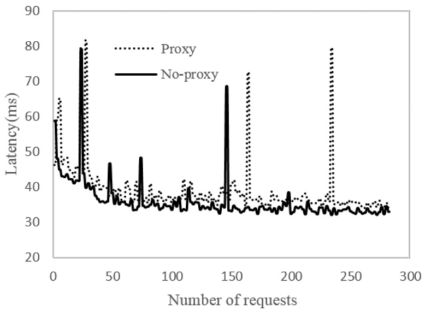

Fig. 5. The effect of I/O proxy on service delays

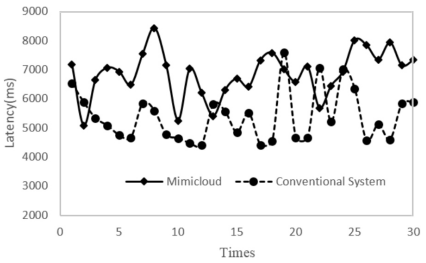

Fig. 6. The effect of multiple executors on service delays

First, we only consider that there one attribute (OS type) for executor X and attack Y. Similar to Ref.[12], we quantize the compromise probability of executor according to the number of common vulnerabilities. The number of the common vulnerabilities between $\boldsymbol{v}_i$ and u_i is denoted by $\rho(v_i, u_i)$. The probability of being compromised is assumed proportional to the ratio of common vulnerabilities. Thus, it is supposed that adversaries adopt a strategy u_i to attack the task executor cluster, and the executor whose OS is $\boldsymbol{v}_i$ will be compromised with probability p:

$$p \propto P(u_i \mid v_i) = \frac{\rho(v_i, u_i)}{\rho(v_i, v_i)} \tag{1}$$

Then we consider the multi-attribute situation. We consider the executor is compromised only on the condition that all attributes are compromised. For comparison, we assume that the undeclared element of the executors is all the same. Thus, all executors have the same number of attributes. We assume that $p = P(u_i \mid v_i)$. These assumptions are simplified for analyzing the general relationship between system attributes and system security conveniently. Based on majority voting, the adversary conducts a successful attack when at least more than half executors are compromised.

As shown in Fig.7, multi-executors with result comparison can reduce the attack success rate. It is much difficult for the adversary to find the vulnerability of all attributes. Moreover, it is shown that with the increase of the executor's attribute, the attack success rate shows a sharp drop. We can conclude that building a multi-attribute heterogeneous executor cluster can improve the system security competently. However, heterogeneous executor cluster will further increase the delay cost. As illustrated in Fig.8, the system performance of Mimicloud is determined by the worst executor. It is difficult to construct executors that vary from one another and all perform well. The diversified development of 5G services is an important task for Mimicloud.

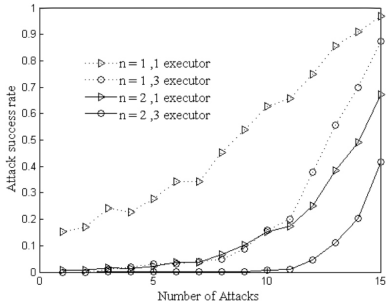

Fig. 7. Relationship between the heterogeneity and attack success rate

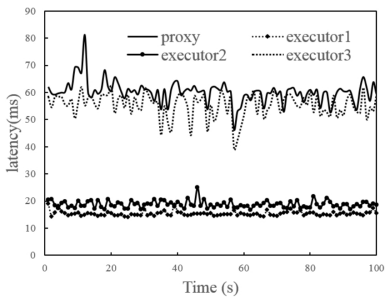

Fig. 8. Time latency of the proxy and executors

V. Related Works

Standardization organizations such as 3GPP, 5GPPP, IETF, ETSI, NGMN have promoted their security architectures and solutions[14]. There are two basic approaches in the current research on 5GC security: reactive cyberspace defense technologies and active defense technologies.

Reactive cyberspace defense technologies try to test, discover, and eliminate potential threats. Anti-malware, installed on mobile terminals or hosted in the cloud, is a great way to improve overall resistance to malware attacks[15]. Encryption based practices are widely used in the mobile network; for instance, encryption technology is taken before sending to Location-based services (LBS) provider [16]. Moreover, identity and access management mechanisms (*e.g.*, zero trust architecture) can mitigate the spread of attacks when some cloud assets are compromised. Liu *et al.*[17] proposed an infrastructure to enable mobile users to share real-time videos on 5G enabled clouds by restricting access to only authorized customers. Nevertheless, the nature of reactive security mechanisms has limitations. As cyber-attacks become more intelligent and coordinated, they can breach the traditional defense mechanism. Therefore, developing game-changing defense approaches is necessary and inevitable.

Moving target defense, widely used in cloud computing, is a proactive security solution to defend against cyber-attacks in 5GC SBA. MTD has been extensively studied in IP randomization, virtual machine migration, virtual machine scheduling, *etc.* Seth *et al.*[18] promoted a Dynamic heterogeneous shortest job first (DHSJF) model and considered both dynamic heterogeneities of workload and dynamic heterogeneity of resources. Thompson *et al.*[19] developed a Multiple OS rotational environment (MORE), which consists of a set of virtual machines equipped with different Operate Systems and Web applications. MTD techniques have used various models and solution techniques (*e.g.*, game theory, genetic algorithms, and machine learning)[20]. Sengupta *et al.*[21] determined an optimal switching strategy for a Web application based on the Bayesian Stackelberg game, maximizing the security utility and minimizing the switching overhead. Zhu *et al.*[22] supposed that the defender has no or limited information about the attackers. Two iterative reinforcement learning algorithms were proposed based on Markov chains to obtain an ideal defense strategy.

VI. Conclusions

To offer highly available, reliable, and trusted 5G service, we present Mimicloud architecture, a high-security SBA 5G core network inspired by CMD. In this paper, we describe the implementation details of Mimicloud and introduce several principal security mechanisms, including executor reconfiguration, multiple executors with crosschecking, and heterogeneous executor pool. Moreover, we model the task scheduling as an M/M/c queue and analyze our mechanism's performance in theory. Finally, the security and availability of Mimicloud are analyzed in the experiment. The results show that the CMD method produces high security with an acceptable additional performance overhead.

References

[1] 3GPP TS 29.500 v1.0.0:2018, 5G System Technical Realization of Service Based Architecture.

[2] A. Kanellopoulos and K. Vamvoudakis, "A Moving Target Defense Control Framework for Cyber-Physical Systems",

IEEE Transactions on Automatic Control, Vol.65, No.3, pp.1029–1043, 2020.

[3] B. Spasic, A. Rath, P. Thiran, *et al.*, "Security pattern for cloud SaaS: from system and data security to privacy", *4th International Conference on Cloud Computing Technologies and Applications* (*Cloudtech*), Brussels, Belgium, pp.1–8, 2018.

[4] P. Stewin and I. Bystrov, "Understanding DMA malware", *9th Int Conf on Detection of Intrusions and Malware, and Vulnerability Assessment*, Heraklion, Crete, Greece, pp.21–41, 2012.

[5] A. Verma, M. Mittal and B. Chhabra, "The mutual authentication scheme to detect virtual side channel attack in cloud computing", *International Journal of Computer Science and Information Security* (*IJCSIS*), Vol.15, No.3, pp.83–98, 2017.

[6] I. Ahmad, T. Kumar, M. Liyanage, *el al.*, "Overview of 5g security challenges and solutions", *IEEE Communications Standards Magazine*, Vol.2, No.1, pp.36–43, 2018.

[7] W. Meng, Y. Wang, W. Li, *et al.*, "Enhancing intelligent alarm reduction for distributed intrusion detection systems via edge computing", *Australasian Conference on Information Security & Privacy*, Springer, Cham, pp.759–767, 2018.

[8] H. Lauer and K. Nicolai, "Hypervisor–based attestation of virtual environments", *Ubiquitous Intelligence and Computing*, Toulouse, France, pp.333–340, 2016.

[9] M. Schwarz, S. Weiser, D. Gruss, *et al.*, "Malware guard extension: using SGX to conceal cache attacks", *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Bonn, Germany, pp.3–24, 2017.

[10] J. Wu, *Cyberspace Mimic Defense: Generalized Robust Control and Endogenous Security*, Springer, Cham, pp.207–272, 2020.

[11] B. Zhang, X. Chang and J. Li, "A generalized information security model SOCMD for CMD Systems", *Chinese Journal of Electronics*, Vol.29, No.3, pp.417–426, 2020.

[12] Y. Wang, J. Wu, Y. Guo, *et al.*, "Scientific workflow execution system based on mimic defense in the cloud environment", *Frontiers of Information Technology & Electronic Engineering*, Vol.19, No.12, pp.1522–1536, 2018.

[13] L. Kleinrock, *Queueing systems, Volume I: Theory*, John Wiley & Sons, New Jersey, USA, pp.9–20, 1975.

[14] I. Ahmad, T. Kumar, M. Liyanage, *et al.*, "Overview of 5g security challenges and solutions", *IEEE Communications Standards Magazine*, Vol.2, No.1, pp.36–43, 2018.

[15] M. Polla, F. Martinelli and D. Sgandurra, "A survey on security for mobile devices", *IEEE Communications Surveys Tutorials*, Vol.15, No.1, pp.446–471, 2013.

[16] X. Pan and Z. Xiao, "Survey of location privacy–preserving", *Journal of Frontiers of Computer Science and Technology*, Vol.1, No.3, pp.268–281, 2007.

[17] J. Liu, M. Au, W. Susilo, *et al.*, "Secure sharing and searching for real-time video data in mobile cloud", *IEEE Network*, Vol.29, No.2, pp.46–50, 2015.

[18] S. Seth and N. Singh, "Dynamic heterogeneous shortest job first (DHSJF): A task scheduling approach for heterogeneous cloud computing systems", *International Journal of Information Technology*, vol. 11, pp.653–657, 2018.

[19] M. Thompson, N. Evans and V. Kisekka, "Multiple OS rotational environment an implemented moving target defense", *IEEE International Symposium on Resilient Control Systems* (*ISRCS*), Denver, CO, USA, pp.1–6, 2014.

[20] J. Cho, D. Sharma, H. Alavizadeh, *et al.*, "Toward proactive, adaptive defense: A survey on moving target defense", *IEEE Communications Surveys & Tutorials*, Vol.22, No.1, pp.709–745, 2020.

[21] S. Sengupta, S. Vadlamudi, S. Kambhampati, *et al.*, "A game theoretic approach to strategy generation for moving target defense in Web applications", *International Foundation for Autonomous Agents and Multiagent Systems*, São Paulo, Brazil, pp.178–186, 2017.

[22] M. Zhu, Z. Hu and P. Liu, "Reinforcement learning algorithms for adaptive cyber defense against heartbleed", *ACM Workshop Moving Target Defense*, Scottsdale, Arizona, USA, pp.51–58, 2014.

LI Lingshu is a Ph.D. candidate at National Digital Switching System Engineering and Technological R&D Center (NDSC), Zhengzhou, China. He received the B.S. degree in electrical engineering from Shanghai Jiao Tong University in 2014 and M.S. degree in mobile core network from NDSC in 2017. His research interests include cyber security and cloud computing. (Email: lls.ndsc@aliyun.com)

WU Jiangxing is a professor and the director of China National Digital Switching System Engineering and Technological R&D Center (NDSC). In 2003, he became an academician of China Academy of Engineering. His research interests include cybermimicsecurity and computer architecture.

HU Hongchao is currently an associate professor of the National Digital Switching System Engineering and Technological Research and Development Center (NDSC), Zhengzhou, China. His research interests include cloud computing security, SDN&NFV security and cyber security.

LIU Wenyan, born in 1986. Ph.D., lecturer in National Digital Switching System Engineering and Technological Research and Development Center (NDSC), Zhengzhou, China. His main research interests include cloud computing, software-defined network and cyber security.

GUO Zehua(corresponding author) received the B.S. degree from Northwestern Polytechnical University, the M.S. degree from Xidian University, and the Ph.D. degree from Northwestern Polytechnical University. His research interests include software-defined networking, cloud computing. (Email: guolizihao@hotmail.com)