Intrinsic Security and Self-Adaptive Cooperative Protection Enabling Cloud Native Network Slicing

WU Qiang*, WU Chunming*, YAN Xincheng, and CHENG Qiumei

***Abstract*—With the emergence of cloud native technology, the network slicing enables automatic service orchestration, flexible network scheduling and scalable network resource allocation, which profoundly affects the traditional security solution. Security is regarded as a technology independent of the cloud native architecture in the initial design, traditional passive defense such as "reinforced" and "stacked" is relied on to achieve system security protection. The lack of intrinsic security mechanisms makes the system capability insufficient when faces the uncertain threat brought by vulnerabilities and backdoors under the ecosystem of opening-up and sharing. The static nature of existing networks and computing systems makes them easy to be compromised and hard to defend, and thus it is urgent to provide intrinsic security and proactive protection against the unpredictable attacks. To this end, this paper proposes a novel paradigm named intrinsic cloud security (iCS) from the perspective of dynamic defense. The dynamic defense provides component-level security, and has complementary and consistency with the cloud native environment. In particular, iCS introduces mimic defense and moving target defense (MTD), and makes full use of the new features introduced by cloud native to implement an intrinsic and proactive defense mechanism with acceptable costs and efficiency. The iCS paradigm achieves seamless integration and symbiosis evolution between security and cloud native. We implement a trial of iCS based on 5GC commercial system and evaluate its performance on costs, efficiency and attack success. The result shows that the iCS enhanced mode always can provide a better and more stable defense effects.**

***Index Terms*—Cyber security, MTD, mimic defense, cloud native, Network Functions Virtualization (NFV), Software Defined Network (SDN), network slicing.**

I. Introduction

THE fifth generation (5G) is expected to become an end-to-end flexible, scalable and service-oriented system to meet the requirements in diversity scenarios, and network slicing is considered as a key driving force to achieve this challenging goal. As the next generation virtualized core network architecture [1], cloud native technologies enable loosely coupled systems that are resilient, manageable, and observable. When facing the deployment of various cloud-based and 5G on-demand slicing solutions, using cloud native thinking and models to build a cloud-based 5G core network is bound to accelerate the digital transformation of the entire communications industry [2]. Precisely, the notable features cover the following four aspects:

1) Microservice: Service chains are constituted in the granularity of stateless microservices.

2) Automation: The applications should be highly automated through embedded features in the NFVI (NFV Infrastructure) layer at all stages, including blueprint design, resource scheduling and orchestration, lifecycle management, status monitoring, and policy control updates. The uniform deployment of cloud network resources is achieved [3][4].

3) Lightweight virtualization: In actual deployment, NFCs (Network Function Conponent) are decoupled from underlying virtualization technologies. They can be deployed in a mixed environment of containers, bare-metal and VMs (Virtual Machine). This improves resource utilization and enables fast service delivery and agile application maintenance [5].

4) DevOps (Development & Operations): Programmability based on telecom network capabilities is critical to service innovation and ecosystem richness, and the network paradigm evolves from network sharing to multi-tenancy [6].

Combined with robust automation, above features allow network slicing to make frequent and predictable changes with high impact and with minimal toil. As a specific network function, the intrinsic security mechanism of networks and computing systems evolves with the changes of the application scenario and the service mode [7]. Under SDN/NFV paradigms, cloud native technology has become an essential component of cloud infrastructure, it has profoundly affected and changed the traditional security assurance system, implementation methodology and management system, posing challenges to the existing security protection system [8]. On one hand, the availability and security of centralized control planes are particularly acute. On the other hand, the relationship among multi-tier applications software, hardware, networks, and operating systems under the microservice framework becomes more complex [9]. This complexity also introduces more uncertainties to the security system in the cloud native scenario for network slicing, and incurs a larger attack surface [10]. In addition, traditional security technologies and solutions focus on the host system and network boundaries, and are isolated from each other. It is difficult to achieve consistency of security policies and rapid coordination of various defense methods [11]. In the Devops mode of multi-tenant collaboration, the security risks of 3rd components are also difficult to be

WU Qiang is with College of Computer Science and Technology, Nanjing University of Aeronautics & Astronautics, Jiangsu Nanjing 210016, China, and with College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China, corresponding author*, e-mail: wu.qiang@zte.com.cn.

WU Chunming is with College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China, corresponding author*, e-mail: wuchunming@zju.edu.cn.

YAN Xincheng is with State Key Laboratory of Mobile Networks and Mobile Multimedia Technology, Nanjing 210012, China.

CHENG Qiumei is with College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China.

Manuscript received September 19, 2020.

detected effectively.

Many cloud computing management systems, such as Spring Cloud, are unable to provide a dedicated security solution. The security protection in cloud native for network slicing requires targeted and feasible solutions. The iCS as a public support platform is proposed to align cloud native technology with mimic defense and MTD paradigm, the purpose is to achieve the intrinsic security and self-adaptive cooperative protection of the cloud native environment. Under the constraints of cost and efficiency, the iCS paradigm implements seamless integration and symbiosis evolution between security and cloud naive environment. The main contributions of this study are summarized as follows:

- The security challenges that pose threats to the cloud naive environment and its pivotal characteristics are systematically reviewed, we found that there is good fitness and coherence between the dynamic defense ideas and cloud native for network slicing.
- The iCS framework is proposed to align the cloud naive environment with the paradigm of mimic defense and MTD, thus to achieve secure and reliable network slicing as well as allround defense against the uncertain threat brought by vulnerabilities and backdoors under the ecosystem of opening-up and sharing.
- The intrinsic security features are expanded at the ETSI (European telecommunications standards institute) NFVI architecture level such as the heterogeneous dynamic redundancy of the attack surface and the automated deployment of key security components.

The rest of this paper is organized as follows. The following section introduces related work and emerging requirement. Then the iCS system architecture are proposed in section 3. The theoretically modeling and solving of the defense ability of iCS are presented in section 4. Afterward the performance on costs, efficiency and attack success is evaluated in section 5, and some conclusions are given in the last section.

II. Related Work

The static nature of existing networks and computing systems makes them easy to be compromised and hard to defend. Attackers have an asymmetric advantage in that they are able to sniff a system, study its vulnerabilities, and choose the most appropriate opportunity to gain the maximum benefit of attack. To overcome the inherent defects, some dynamic defense ideas have been proposed to improve the security gains of networks and computing systems. The most representative theories include MTD idea and mimic defense theory.

A. Threat analysis and emerging requirement in the 5G cloud native environment

Cloud native yields numerous benefits, and some new features, such as multi-tenancy, capabilities openness, VNF provision by many different vendors, are provided to construct the ecosystem of opening-up and sharing. These features emerge at the cost of some security flaws and backdoors, 5G network slice becomes vulnerable to a number of security threats [12]. Traditional defense concept is passive response mechanisms in a fixed manner based on prior knowledge, usually viruses or malicious behaviors features. As the production relations in the 5G ecology tend to be complex and open, it is difficult to exhaust those malicious behaviors and virus features including unknown threats. The passive response mechanism based on prior knowledge is obviously not enough to keep up with the development of threats.

The security threats within the ecosystem of opening-up and sharing mainly come from two aspects within the ecosystem of opening-up and sharing: one is that there are the huge amount of code (some as many as tens of millions of lines) in the implementation stages, which determines that the vulnerabilities is numerous and unavoidable; the other is that the system has the characteristics of multi-tenants and openness, the flaws are unconsciously introduced or the backdoors are deliberately implanted while the use third-party software or open source code [13]. The static model of the existing cloud native environment follows the fixed predictable and consistent laws, and the attack experience against one sample can be popularized and applied to all products, making the scope of the same attack damage larger. In the scenario setting, attackers exploit and use victim system vulnerabilities or backdoors to breakthrough and control the target system, the attack strategies cover the current conventional attack methods, the poisoning attack [14] and advanced persistent threats (APT), etc., but do not include the attacks type of resource exhaustion against system availability. The uncertain threat brought by vulnerabilities and backdoors is the most serious security challenge for the cloud native environment. This paper focus on the most widespread threat scenarios, which are caused by known or unknown vulnerabilities in the system or protocol, as well as uncontrollable system backdoors, and the protection capability for boundary isolation is not involved.

Such security threats raise the concern about the technical requirement gaps between 5G services with higher security requirements such as industrial control and the main enabling technologies with certain security risks [11][15]. For a cloud native in based on COTS (Commercial Off-The-Shelf) involving multiple parties to be considered complete, it needs to comply with the same functional and security requirements as NFs in dedicated hardware:

- The solution needs to provide security gains at the architectural level for all parties involved. The addition of multi-tenancys and outsourcers makes the relationship between network operators and vendors no longer stick to the “one-to-one” fixed relationship, and the fluid and overlapping outsourcing relationship established by the network platform is continuously expanding its scale. Which brings an issue about the imbalance and mismatch of the security capabilities among the service providers, and a universal architectural-level security compatible with all parties is required.
- The solution needs to provide the capabilities for seamless integration and symbiotic evolution between security and cloud native. The lightweight VNF design ideas can better satisfy the requirements of flexible network slicing than a traditional heavy monolith VNF. There are more interface relationships among components [5], which

introduces a greater attack surface. For the solution, it is important to make full use of the above-mentioned features introduced by cloud native to offset the increase in attack surface and improve the intrusion tolerance of the system, and as little as possible to influence on the level of performance and cost.

- The solution needs to provide a simple and effective judgement way for network abnormalities in case of multi-scenario and diversified application. The passive defense effectiveness often depends on the scale of the prior knowledge about malicious behavior features and so on. It is difficult to guarantee the maturity, comprehensiveness and timeliness of the existing prior knowledge. There is a theoretical challenge for the judgement criteria of malicious behavior and its real-timing, when new types of attacks and zero-day vulnerabilities emerge in endlessly [16].

Morever, traditional "reinforced" and "stacked" defense solutions also face the issues of cost, seamless integration and symbiosis evolution in the cloud naive environment, whereas the intrinsic security issue is not considered in the NFVI reference architecture as defined by ETSI [17].

B. Moving Target Defense

The idea of MTD tries to build self-defending systems by making systems dynamic and therefore harder to be sniffed and predicted [18][19]. A constantly changing system makes the resources that an attacker can exploit uncertain in the space-time dimension, and a macroscopic is that the attack surface moves irregularly. According to the attack surface theory, the attack accessibility of target object cannot be guaranteed, which increases the attackers workload. Some works presented a large existing range of MTD techniques to constantly change network configurations or parameters, firewall settings, operating systems, memory addresses, instruction sets, or application execution environments. However, due to the conditions of infrastructure at that time, the implementation of these MTD paradigms is highly coupled with its service system, a dedicated MTD solution is required for each specific service system. MTD security scheme and service system nested with each other, coupling and interface compatibility issues limit the scope and effectiveness of the solution. In order to implement them, an application developer has to be proficient both service logic and security skills. With the openness of network capabilities and the improvement of the ecological chain, multi-vendor component integration, multi-tenancy, and DevOps become the main application mode in the cloud native environment. It becomes more difficult to embed these dedicated MTD security solutions into all aspects of service design, network topology and network element dynamic deployment.

The capabilities developed by SDN/NFV for network slicing, such as the dynamic resource scheduling, resilient scale in and out, and software definition, provide an important guidance and convenience for the dynamic and flexible deployment of MTD security policies [20]. On the basis of analyzing the characteristics and security challenges of cloud native technology, it is necessary to consider how to introduce MTD from the top-level design of the overall security architecture.

C. Mimic Defense Theory

The mimic defense theory provides a new perspective for solving component-level security issues, that is, the system itself has a typical DHR (Dynamic Heterogeneous Redundancy) structural model and thus has the capability to proactively defend. The model is composed of the service dispatch agent (SDA), heterogeneous components sets with equivalent functionality, shuffling strategies algorithms, executor sets and mimic decision point (MDP) with majority selection ruling algorithms. The SDA forwards the input incentives to every one in the executor set according to the forwarding strategy, and the output results from different executors are submitted to the MDP voter for judgment, and the system output is obtained. It does not depend on known sample data, no post-maintenance [21]. Some previous work presented several software fault tolerance techniques, several diversity defense techniques in popular operating systems were also discussed such as address space randomization, instruction set randomization [22], and data randomization [23]. In that regard, the recent advances, including SDN and cloud native, as well as the software diversity technologies as described above, allow for acceptable cost and efficient deployment of the mimic defense idea, which makes it possible for a cloud native environment to achieve seamless integration and symbiosis evolution with intrinsic security, thus a proactive defense and high security gain are obtained.

Just as a triangle has natural stability, it is widely used in the design of highly stable structures. We use micmic defense theory and MTD idea to enhance the intrinsic security of the cloud environment. Table I provides a summary of the security challenges associated with cloud native as discussed above, and lists the security gains along with iCS techniques. Our iCS implementation approach include the basic mode and the enhanced mode. The diversity software technique is adopted to generate functionally equivalent heterogeneity components in the basic mode. The features of automatic deployment, elastic scalability, service chaining orchestration in existing cloud foundation service are extended to achieve the dynamic scheduling of the function executor and redundancy management of resources. And in the enhanced mode, the subscription publication (P/S) mechanism is provided to support common communication among heterogeneity components, SDA, and MDP.

III. A DHR Architecture for Cloud Native Environment

A. Intrinsic Security and Self-Adaptive Cooperative Protection

Intrinsic security is an innate security nature that is an indivisible part of a service system. It provides the surface defense instead of the point defense. According to the mimic defense theory, the Dynamic Heterogeneous Redundancy (DHR) model is an intrinsic defense method based on the architecture technology [29]. It does not rely on prior knowledge, and the genes of DHR model are highly complementary

TABLE I: The challenge and security gain in cloud native

Challenge in cloud native	Security features of related work	Security features of iCS
Stealth malware [12], malicious insider [17].	Volume/swap encryption, VNF image signing, strict operational practices [12][24].	Majority consistency judgement mechanism from heterogeneous replicas, component-level, architecture-level protect.
Vulnerability.	Point defense, passive protection based on prior knowledge, usually viruses or malicious behaviors features [25].	Surface defense, majority consistency judgement mechanism. Proactive defense without prior knowledge.
Wider spread of the virus&vulnerabilities because of infrastructure homogenization, shared components.	Regular hypervisor patching, disable all unused services [25].	Component diversity, smaller attack surface, detecting&cleaning mechanism for poisoning replicas.
Compromised hypervisor, security loopholes caused by interoperability issues of many different vendors [26].	VM separation and management traffic, hypervisor patching. Geo-tagging using remote attestation for regularity compliance [27].	Component diversity, majority consistency judgement mechanism with the heterogeneous redundant characteristics, detecting&cleaning mechanism for poisoning replicas.
Security symbiosis evolution with multi-tenancys, outsourcers, DevOps.	Crypto&Trusted hardware [13].	Generalized robustness, intrinsic security complied with ETSI NFV architecture.
DDos attack [17].	Flexible VNF strategic deployment [28].	Network topology mutation, dynamic&random VNF deployment, dynamic migration.
Side-channel attack.	Trusted execution environments [13].	Shuffling.
Boundary isolation failure.	Hypervisor introspection [17], security zoning.	N/A

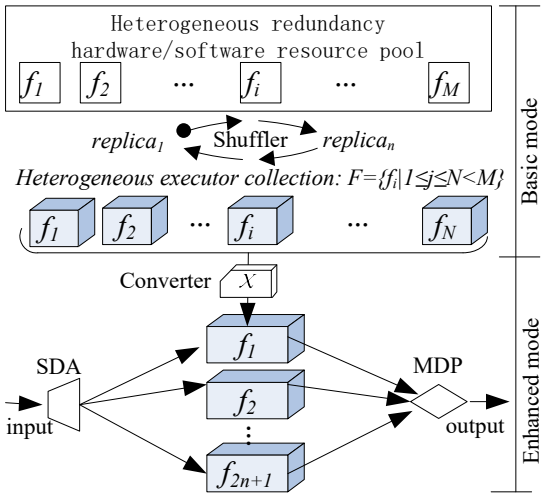


Fig. 1: The abstract model of unified paradigm both MTD and mimic defense

to cloud native technology. The DHR model increases the uncertainty of the cloud-native environment or structure, and causes continuous and irregular changes in the attack surface at a macro level, which aims to significantly increase the difficulty and cost of the attacker. It is an essential precondition for implementing the mimic defense theory, and also can be the sufficient condition for MTD idea to perform effectively. Fig. 1 shows the abstract model of unified paradigm both MTD and mimic defense. The security gain of the DHR model can better cope with the new security challenges brought by cloud native technology architecture. At the same time, the new information network technology promotes new development of security defense capability, the existing features of cloud native, such as versatility, lightweight virtualization, microservice, and centralized control create more convenient basic facilities for the realization of DHR model in infrastructure. It is a good technical and economic choice to alleviate the contradictions between current or future cloud services and security credibility.

Based on cloud native's definable, virtualized, and dynamic design concepts, self-adaptive cooperative protection breaks the ecological closure of security equipments. Following the principle of minimum openness, effective interaction is realized among security components or between security components and application software to enhance the overall security. It can achieve unified management and control, hybrid orchestration, seamless fit and symbiotic evolution both security components and service application software. And security policies can be dynamically adjusted and elastically deployed according to the service needs.

B. The DHR Issues in the Cloud Native Environment

As an architecture technique of the information system, the DHR model has universal applicability. For specific application scenarios of cloud network convergence, some specific issues also need to be solved. The first issue is how to construct a functionally equivalent heterogeneous component pool. Although heterogeneity components pool has raised the threshold for attackers to attack and crack a system to a certain extend, it also increases the difficulty of system development and maintenance, and the cost is expensive for a large-scale cloud native infrastructure for network slicing. Therefore, in the background of business agility and multi-vendor component integration that spans organizations, communications and computing platforms, the highly scalable construction method that satisfies the constraints of cost and efficiency is critical.

The second issue is the dynamic scheduling of function executors. The dynamic character of a DHR model requires that a function executor can be dynamically selected and loaded from the heterogeneity components pool, and the abnormal executor can be isolated and replaced, and the online executor can dynamically change its apparent structural representation. Although a lot of efforts have been made to improve the service features of elasticity, automation, flexibility, and openness that SDN/NFV promises, the efficient and effective method has yet to be developed how its processes is used to instantiate the heterogeneous redundant executor, so that the dynamic scheduling of executor can be realized conveniently

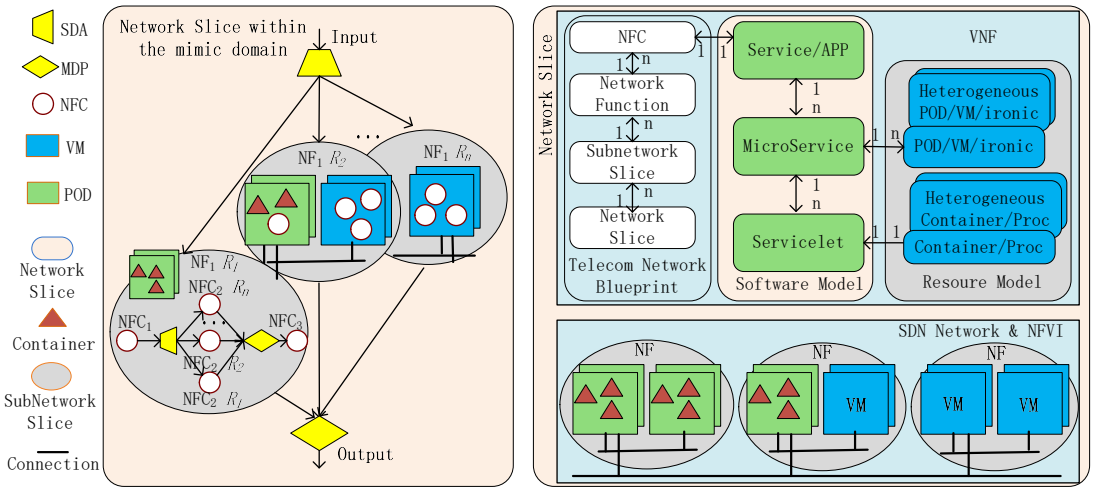


Fig. 2: Mapping relationship among service, software and resources in iCS

and economically. At the same time, the implementation cost of DHR introduction is reduced.

The third issue is the setting of the service dispatch agent (SDA) and mimic decision point (MDP) in the service processing flow. The MDP is not suitable for all service components in the cloud native environment, such as strict real-time services and components without normalized input/output interfaces. The difference in the attributes of the components need to be considered in the cloud foundation service. Likewise, the management and automatic deployment of SDA and MDP need to be considered in the cloud foundation service. Therefore, the functions of existing cloud foundation service need to be extended to meet the requirements of the service system. In which, the management of SDA and MDP implements their lifecycle management, the logic relationship management with components, as well as resource reservation strategy. The SDA and MDP should be deployed automatically by the cloud foundation service and deployed incrementally based on different security levels.

C. Information Model and Design Constraints of iCS System

Conceptual architecture identifies the high-level components of a system, and the relationships among them. Fig. 2 shows the mapping relationship among service, software and resources in iCS system. A network slice is taken as an example, it contains multiple subnet slices, each of them implements a NF function. A subnet can be composed with mixed heterogeneous environment of containers, bare-metals and VMs. PODs are filled with multiple containers to load servicelet for microservices. To implement a network slice, servicelets are assembled into microservices, and then microservices further assembled into services through registration and discovery mechanism. After mapping services to the NFCs, network slices are formed in telecom blueprint through layers of orchestration and management on NFCs/NFs/SubNetwork slices.

Although MTD increases the difficulty of exploiting software vulnerabilities to a certain extent, it does not completely eliminate threats. Following the mimic theory [30], a mimic decision mechanism is introduced to generate a more reliable output than a single executor in a multi-version assembly. As shown in Fig. 2, the SDA/MDP can be automatically deployed at a location with a normalized I/O interface, such as NFC, subnet, network slicing, etc. Network service orchestration and management are extended to realize the establishment, management and revocation of SDA/MDP, so that SDA/MDP can be flexibly and dynamically managed and deployed to meet the system requirements of the DHR model, and the increase and decrease of the subsequent docking components will not affect core service system.

In the cloud native scenario, the precondition of the mimic decision needs to be considered in advance in the phase of architecture as binding requirements, so that the mimic decision can be implemented in more components. The key design principles of mimic constraints for a microservice include: The interface design among microservices follows the input and output normalization principle. Independent deployment and management principles of microservice. High reliability principle, the external service is uninterrupted when a single microservice node is abnormal, the external service interface does not change when resilient scaling.

D. Automated development framework for VNF replica collection

Some previous works have laid a solid foundation for software diversity as a defensive technology in the virtualized cloud environments [31][32]. According to the theory of dynamic defense, the higher the degree of heterogeneity and redundancy, the greater the range of changes in the attack surface, so that there is a higher level of security and reliability [33]. However, higher heterogeneity and redundancy bring higher cost, and it is necessary to consider specific implementation schemes under the constraints of cost and efficiency according to the requirements of the security level. The heterogeneous redundancy of iCS is characterized as outlined below:

1) For the lightweight VNF that constitutes the service chain of the 5G network slice, the iCS basic mode as the

default configuration, the diversity compiler tools are integrated to build equivalent heterogeneous executor. Which has the advantages of convenient deployment, low cost and generalized robustness. The attack process relies on unchanged, predictable, and consistent laws in the system. VNF random and diverse security genes are injected into the existing cloud native system, so that it has the capability of proactive defense. According to the predefined levels of heterogeneous level (e.g. how many different application versions or platforms), the iCS framework is compatible with any diversified technology, which can generate variants with equivalent properties that differ enough to prevent certain types of attacks.

2) In the existing cloud network convergence environment, its 1+1 or N+M backup mechanism of the service system for high reliability, the resource pool, and the homogeneity of the infrastructure constitute a sufficient condition for DHR redundancy. Its redundancy level and heterogeneity level (e.g., the number and the type of infrastructure resources) can be dynamically and flexibly adjusted according to the security level requirements.

3) All heterogeneous components are numbered and managed in a unified manner, and their trust levels are dynamically maintained. The trust levels are used for cleaning and selection of the executor of subsequent processes.

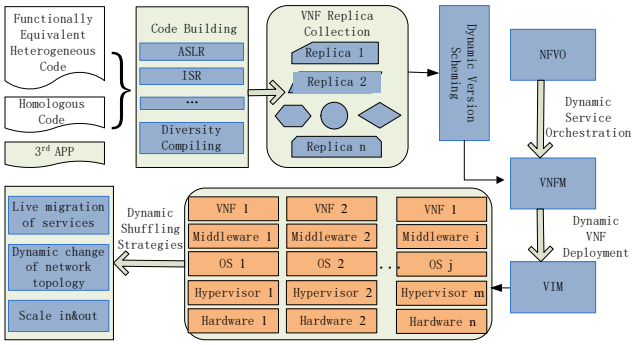


Fig. 3: CI/CD pipeline of the VNF replica collection

Fig. 3 shows how a VNF (Virtual Network Function) replica collection generated by diversity compilers is deployed in a cloud environment, the pipeline is mainly to achieve continuous integration and continuous deployment (CI/CD) of VNF. After a successful code building, VNF replica collection is assembled into the software warehouse as a functional equivalent, and upload the newly generated docker image to the integrated docker registry. They are used to meet the requirements of heterogeneity and redundancy in the DHR model. NFVO (NFV Orchestrator) focuses on providing the orchestration of service chain requested in the MANO (Management and Orchestration) domain specified in ETSI NFV standards, which implements automatic orchestration and the whole lifecycle management of virtual network services, VNFM (VNF Manager) finishes the lifecycle management and dynamic immigration of VNFs. After a orchestration and a resilient deployment, the VNF replicas are selected randomly or according to a preset policy, and then are initially deployed on the cloud infrastructure.

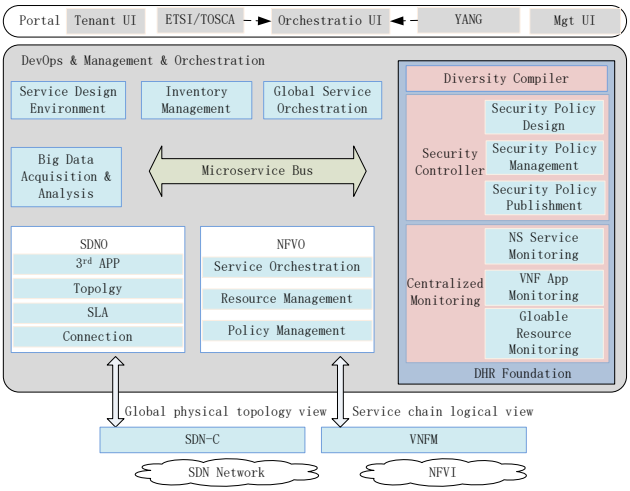


Fig. 4: Dynamic shuffle and orchestration management in the iCS system

E. Dynamic shuffle and orchestration management in iCS system

As long as the attack surface of the protected system has the characteristics of random changes, it is difficult to ensure that information stealing methods or effective attack schemes can be continuously effective. Accurate use of the existing characteristics of the cloud-native system can equivalently obtain the effect of the atypical structure of DHR, which will increase the difficulty of the attacker's perception of the protected system. In addition, under the constraints of cost and efficiency, it can achieve seamless integration and symbiotic evolution between security mechanisms and service systems. Irregular component scheduling and VNF migration will destroy the reachability of the attack chain, while virtualization and dynamic resource allocation mechanisms may destroy the stability of the attack chain. This will be a realistic technical and economic choice to ease the contradiction between current or future cloud trends and security credibility.

Fig. 4 shows the framework of dynamic shuffle and orchestration management in iCS system for network slicing. On the basis of the original orchestration and management layer, the DHR foundation is extended to realize the additional features required by the mimic theory. Dynamic and randomness are the source factor of uncertainty, the four features of cloud native are fully utilized, these existing ideas provide a solid basis for the dynamic and randomness of the DHR model. Virtualization realizes the decoupling of the underlying physical device and upper operating system and application software, Heterogeneous hardware resource types are more abundant. Following the design ideas of stateless microservices and lightweight, the executor set is presented in a smaller granularity. Thus the normalization principle of input and output interfaces among microservices is more easily satisfied.

Based on the NFV architecture of the ETSI standard, the DHR foundation maximizes the use of the existing mechanisms, such as dynamic shuffle and orchestration management, to maximize the dynamic and randomness, which can effectively reduce the cost increase brought by the DHR

security gain. At the same time, due to the better interface compatibility, it can support the applications in a wider range of fields rather than a specific dedicated system, and gain higher efficiency of development and runtime. The security controller is primarily responsible for the creation, management, and distribution of the unified security policies. The above constitutes the basic mode of iCS.

There are many different types and characteristics of components in a cloud native system, not all components can satisfy the preconditions of the mimic decision. For such VNF with scalability requirements, it is necessary to provide a unified access channel so that the SDA and MDP can be flexibly and dynamically managed, expanded and deployed to meet the system requirements of the DHR model, which requires the enhanced mode of iCS. The main role of the microservice architecture is to decompose functions into discrete service components, thereby reducing system coupling and providing more flexible service support. Thus the distributed message service (DMS) can help application developers freely forward packet, deliver messages, and build loosely coupled system on the distributed components. The automatic deployment of SDA/MDP is realized through the DMS subscription and publication mechanism.

Under NFVO orchestration [34], SDA/MDP is inserted into a service chain which is a protected sample within the mimic domain. Through automatic deployment process, SDA is automatically set on the mandatory channel of the input message, and MDP is automatically set on the mandatory channel of the output message. According to the security controllers policy, the heterogeneous replicas are automatically deployed, and the respective session between replicas is automatically subscribed. The MDP realizes the perception of the function execution abnormality of system through bit-level, load-level, behavior-level or even content-level mimic decision and triggers the corresponding post-processing mechanism.

IV. Modeling and Solving of iCS System

The symbols used in equations are shown in table II.

TABLE II: Symbol description in equations

Symbol	Description
$\lambda(a,d)$	Coefficient of remainder losses.
D_a	Resource loss of the attack target.
T	Defense tolerance, the minimum executors number required to break through defense.
∂T	Difference in tolerance. The fewer executors controlled by attacker, the greater tolerance of system.
$\lvert\partial M_{i+1}\rvert$	Variation difference of the available attack surface.
$Criti$	Reflect the importance of an attacked target.
AL	Reflect the inherent harm of an attack.
c	Number of components under attack.
$\psi(a,d)$	Coefficient of avoidable losses.
∂A_d	Resource overhead generated by defense actions.
$\gamma(a,d)$	Negative coefficient. The degree of negative impact on system availability.
Δt_i	After $i-th$ scheduling request, the duration which the attack surface M_i lasts.
VUL_i	The set of vulnerabilities on the attack surface M_i, VUL=$\{vul_1,vul_2,...,vul_N\}$
q_{vul_j}	The success rate of an attack using vulnerability vul_j to attack an executor containing vul_j.
t_{vul_j}	The time required for an attacker exploiting vulnerability vul_j to attack executor containing vul_j.
α_j	The weight an attacker exploits vulnerability vul_j to attack.
θ	The number of steps in the attack task.

A. Service model as a protected system within the mimic domain

In a iCS system as shown in Fig. 2, those shared infrastructure hardware topology of the infrastructure resource layer can be marked as a weighted undirected graph: G_s = (N_s, S_s, L_s, A_s^n, A_s^s, A_s^l), where N_s denotes a collection of underlying compute nodes, S_s denotes a collection of underlying storage nodes, L_s denotes a collection of underlying links. The attributes of the compute nodes n_s (n_s∈N_s), storage nodes s_s (s_s∈S_s) and links l_s (l_s∈L_s) are denoted by A_s^n, A_s^s, A_s^l respectively. The compute nodes attribute n_s denotes the current available computing capability, and the physical location. The storage nodes attribute s_s denotes the current available storage space, physical location and etc. The links attributes l_s denotes the current available bandwidth resources, and equivalent link. The information of G_s is maintained dynamically and stored in VIM.

Similarly, the topology of the service enabler layer can also be marked as a weighted undirected graph: G_v = (M_v, C_v, D_v, L_v, R_v^m, R_v^c, R_v^d, R_v^l), where M_v represents a collection of media flow functional block, C_v represents a collection of control functional block, D_v represents a collection of data functional block, L_v represents a collection of virtual link. The resources restraint of virtual nodes m_v (m_v∈M_v), c_v (c_v∈C_v), d_v (d_v∈D_v), and virtual link l_v (l_v∈L_v) is denoted by R_v^m, R_v^c, R_v^d, R_v^l respectively. The resources restraint of virtual node mainly considers load balance, computing capability, energy efficiency and physical location etc. The resources restraint of virtual link mainly considers bandwidth resources, path splitting and migration and latency etc.

The quadruple $V^{(i)}$(G_v, G_s, t_r, t_d) denotes a scheduling request, in which t_r stands for arrival instant of a scheduling request, t_d stands for the duration of a scheduling request, Δt_i = t_d - t_r. When the $i-th$ scheduling request arrives, the infrastructure should assign resources to meet the requirement and release resources when the service finishes. The service instantiated process can be defined as a mapping: M_i: G_v(M_v, C_v, D_v, L_v)→G_s($\bar{N}_s$, $\bar{S}_s$, $\bar{P}_s$), where $\bar{N}_s \subset N_s$, $\bar{S}_s \subset S_s$, $\bar{P}_s \subset P_s$. The mapping process initiates a resource creation or a change requirement according to the resource type and resource metadata, and the next-level network elements specifically implement corresponding resources creation. The global service orchestration (GSO) is responsible for activating and testing the end-to-end resources, and feeds the final result back to the security controller. M_i represents the attack surface of iCS system at the time of t_i, the metric of corresponding attack surface as shown in Equation (1).

$$\begin{aligned} |M_i| &= \langle |G_{v_i}|, |G_{s_i}| \rangle \\ &= \left\langle \left| \sum_{g_v \in G_v} der(g_{v_i}) \right|, \left| \sum_{g_s \in G_s} der(g_{s_i}) \right| \right\rangle \end{aligned} \tag{1}$$

where, $der(g_v)$ and $der(g_s)$ represent the security threat weights of the elements g_v and g_s in the set of G_v and G_s respectively.

B. Attack-defense game model and theoretical evaluation of attack difficulty of iCS

Some studies have proposed a quantifiable calculation model of attack-defense costs and a cyber attack-defense game model, which can describe the multi-stage and multi-state attack-defense game process [35]. An irrational attacker only considers how to maximize the reward rather than the cost of an attack. The defense method only needs to study which attack strategy can get the most reward from the perspective of the attacker. But it is more difficult to defend against rational attackers. They will consider the balance between the reward and the cost of the attack. Only the defense research of rational attackers is considered in the paper. It is assumed that the attacker is a rational and intelligent decision-making subject. For a specific cloud native environment, the $CADG$ (Cloud native Attack-Defense Game) is given as Equation (2) shows:

$$CADG = (S, P, U, A, R) \tag{2}$$

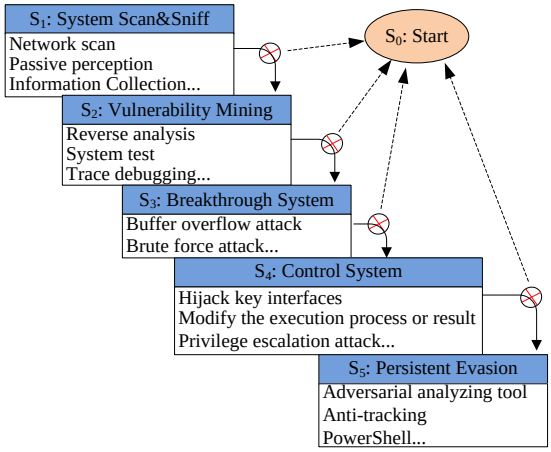


Fig. 5: State transition diagram of the attack chain

S=$\{S_0, S_1, S_2, S_3, S_4, S_5\}$, which is the state of the attack-defense process in the defense tolerance model. Corresponding to the threat analysis in section II, we consider the attack chains that rely on vulnerabilities and backdoors in this effort, as shown in Fig. 5, it is agreed that once an attack operation is intercepted, the attack chain will be reset. The putative attack chain consists of five typical stages: system scan&sniff, vulnerability mining, breakthrough system, control system, and persistent evasion. Each state represents the different security stages in a system after a series of attack and defense measures are taken by both attackers and defenders. Where S_0 is the initial security state when no attack occurs, and S_5 represents the termination state where the system is completely broken by a attacker, and the attacker achieves his attack goal. The system has been substantially damaged in the state S_3.

The success rate of the attack missions P is used as a metric of attack difficulty. Lower value is more desirable. Assuming that the attack success rate is P_i at the $i-th$ stage without protection, the success rate of the attack mission P can be expressed as:

$$P = \prod_{i=1}^{5} P_i \qquad (P_i \leq 1) \tag{3}$$

In the basic mode of iCS, the attack surface is changed dynamically, randomly, and variously, so as to destroy the laws that the attack process relies on. Even if the system has security vulnerabilities, the original known attack path cannot be reached because the possible attack path cannot be predicted. Assuming that N variants (topology parameters, service component replicas) are selected for shuffling in the $i-th$ stage, and their staged attack success rate is recorded as $P_{i,j}$, then the attack success rate P_b can be expressed as:

$$P_b = \prod_{i=1}^{5} \left[\frac{\sum_{i=1}^{N} P_{i,j} * t_j}{t} \right] \qquad (P_{i,j} \leq 1, t_j \leq t) \tag{4}$$

In the iCS enhanced mode, multiple executors with random, diverse, and dynamic characteristics exhibit the same or different results on the same attack path, and the system output is determined by the collective vote of MDP for the multiple results. Therefore, the system security does not depend on a single individual, but is determined by multiple different functionally equivalent variants. Assuming that there are $2n+1$ service component replicas in the $i-th$ stage that are selected for shuffling, the stage success rate of the attack against the repilca j is recorded as $P_{i,j}$. The value of δ_j is "1" when passed by MDP voting, otherwise it is "0". Therefore, the attack mission success rate P_e can be expressed as:

$$P_e = \prod_{i=1,j=1}^{i=5,j=2n+1} P_{i,j} * \delta_j \quad (P_i \leq 1, \delta_j \in (0,1), n \in (1,2,3...)) \tag{5}$$

Since the stage success rate of the attack, P_i and $P(i,j)$, are always less than or equal to 1, there is always $P_e \leq P_b \leq P$. The equal sign is valid only if all of the stage success rate of the attacks are equal to 1 (which is virtually impossible). In the enhanced mode, if an attack mission has to traverse the attack chain multiple times, the success rate of the attack mission will also decay non-linearly, which means that the attack is more difficult. The attack behaviors that are more cleverly constructed and have a higher degree of dependency on the environment are more likely to be defended by the iCS defense mechanisms, which is why the iCS is able to deal with the unknown threats.

$U = \{att, def\}$ is the parties of attack-defense, and the defender is the iCS defense system.

$A = \{a, d, \varepsilon\}$ is the action set of the parties of attack-defense, a=$\{a_{01}$,a_{02},...$a_{i,j}$...,$a_{i,n}\}$ is the action vector of attacker, ε is the empty action of attacker, that is, no attack action is taken; $d = \{d_{01}$,d_{02},...$d_{i,j}$...,$d_{i,m}\}$ is the action vector of defender, the ε is the defender does not take any defensive actions. The i represents the current state S_i of the protected system. The model in this paper is a non-zero-sum game with

incomplete information. The strategy of both attack-defense parties against each other is unknown. Therefore, there are no-op ε in attack-defense actions.

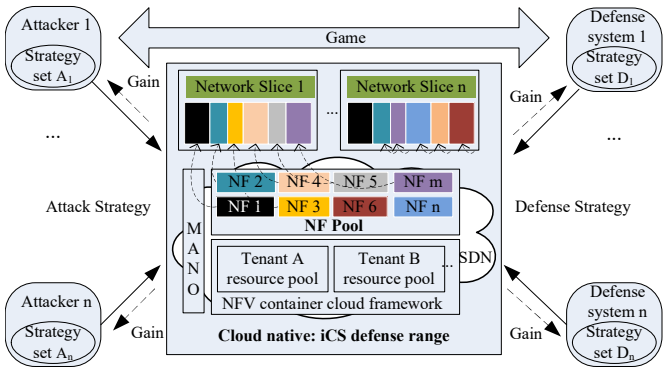


Fig. 6: Attack-defense game model and network topology

$$R^a = \begin{pmatrix} R^a_{11} & R^a_{12} & \cdots & R^a_{1m} \\ R^a_{21} & R^a_{22} & \cdots & R^a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ R^a_{n1} & R^a_{n2} & \cdots & R^a_{nm} \end{pmatrix}$$

$$R^d = \begin{pmatrix} R^d_{11} & R^d_{12} & \cdots & R^d_{1m} \\ R^d_{21} & R^d_{22} & \cdots & R^d_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ R^d_{n1} & R^d_{n2} & \cdots & R^d_{nm} \end{pmatrix}$$

$R = \{R^a, R^d\}$ is the set of profit functions for both attack-defense parties. R^a is the profit of the attacker, which means the profit of a successful attack by the attacker. The attacker profit R^a is generally less than the cyber system loss D_a. The attacker profit is based on the Resource loss of attack target D_a for simple analysis in this paper. Correspondingly, R^d is defined as the defender profit. After a defense strategy is adopted for a certain attack, R^d is related to these factors: the avoidable losses of the cyber system, the defense cost invested, and the system security gain. R expresses the level of profits that both attack-defense parties can obtain from the game, including the profits and cost evaluation criteria of different strategies. Fig. 6 shows the attack-defense game model in the cloud native environment. The profit function set can be expressed as a matrix R. The attack strategies are represented by each row in the matrix. The defender selects each column in the matrix as the defense strategies.

The defense goal of iCS is to ensure the credibility and reliability of the meta-functions of the protected system. While ensuring the availability of meta-functions, its mutation and shuffling processes can refresh the system state. The profit source of the attacker is the increase in the attack surface $|M_i|$ of target system, and the reduction of dynamics and redundancy and heterogeneity in the target system corresponding to the defender profit. The consumption of attack behavior itself is not considered, that is, it believes that the attack resources available are not restricted in this paper. The attacker profit function is given:

$$\begin{cases} R^a_i = \lambda(a,d) * D_a - |\partial M_{i+1}| + \partial T \quad (0 < \lambda(a,d) \leq 1) \\ D_a = \sum_{i=1}^{c} AL * crit * (I_{cost} * f_i + C_{cost} * f_c \\ \qquad +A_{cost} * f_a) \qquad (f_i + f_c + f_a = 1) \end{cases} \tag{6}$$

D_a can be described by criticality ($crit$), Attack Lethality (AL), and security attribute damage. Security attributes include three elements: integrity I_{cost}, confidentiality C_{cost}, and availability A_{cost} [35]. For a specific security attribute, $\{f_i,\ f_c,\ f_a\ \}$ is used to describe the partiality relationship among the three elements. The security attribute damage can be evaluated according to the damage that an attack causes to the target system, and it can be classified as high, medium, and low. The partiality relationship of the three elements can be determined according to the specific cyber environment.

The defender profit includes the system dynamics, redundancy and heterogeneity generated by iCS, which improves the tolerance of the protected system and refreshes the attack surface. The security gains may come at the cost of a decrease in system performance or an increase in power consumption. The defender profit function is given:

$$\begin{cases} R^d_i = \psi(a,d) * D_a + |\partial M_{i+1}| - \partial T \\ \qquad -\partial A_d - \gamma(a,d) * A_{cost} \\ (0 < \psi(a,d) \leq 1), (0 < \gamma(a,d) \leq 1) \end{cases} \tag{7}$$

The above profit function is the strategy profit of attack-defense parties in state i during the game. The strategies not only determine the profit of both parties in the current state, but also affect the future state.

C. Anti-attack model of iCS

The attacker and the iCS system constitute the offensive and defensive sides of the intrinsic proactive defense. Since the attacker is only considered to attack the system based on the vulnerability, their confrontation and game is related to the vulnerability of each component. The security performance of the surface M_i in the iCS system is determined by the heterogeneous components, the network topology and component sets, the orchestration sequence of service chain, and the dynamic scheduling timing. After the movement of the attack surface at time t_{i+1}, the attack surface becomes M_{i+1}, the difference between it and the attack surface M_i corresponding to , which is expressed as Equation (8).

$$|\partial M_{i+1}| = \langle |\partial G_{v_{i+1}}|, |\partial G_{s_{i+1}}| \rangle \tag{8}$$

Then the gain rate of attack surface, i.e. the gain rate of the iCS basic mode, is expressed as Equation (9).

$$\frac{|\partial M_{i+1}|}{|M_{i+1}|} = \left\langle \frac{|\partial G_{v_{i+1}}|}{|G_{v_{i+1}}|}, \frac{|\partial G_{s_{i+1}}|}{|G_{s_{i+1}}|} \right\rangle \tag{9}$$

Where,

$$|G_{v_{i+1}}| = \sum_{g_v \in G_{v_{i+1}}} der(g_v); |G_{s_{i+1}}| = \sum_{g_s \in G_{s_{i+1}}} der(g_s);$$

$$|\partial G_{v_{i+1}}| = \sum_{g_v \in (G_{v_{i+1}} \cap G_{v_i})} (der_{i+1}(g_v) - der_i(g_v)) + \sum_{\substack{g_v \in G_{v_{i+1}} \\ g_v \notin G_{v_i}}} der_{i+1}(g_v) \quad ;$$

$$|\partial G_{s_{i+1}}| = \sum_{g_s \in (G_{s_{i+1}} \cap G_{s_i})} (der_{i+1}(g_s) - der_i(g_s)) + \sum_{\substack{g_s \in G_{s_{i+1}} \\ g_s \notin G_{s_i}}} der_{i+1}(g_s) \quad ;$$

The higher the attack surface gain rate means the higher the change degree of attack surface, the harder it is for an attacker to obtain the attack surface.

The ability of an attacker is determined by its attack success rate for a single vulnerability on a heterogeneous component, the time it takes for the attack, and its strategy of exploiting the vulnerability or combination of vulnerabilities. So the DHR system model will be expressed by these factors and their attributes. These factors or attribute parameters are combined to express the DHR system model as a 7-tuple, as shown in Equation (10).

$$\Omega = \left\langle M_i, \Delta t_i, VUL_i, q_{vul_j}, t_{vul_j}, \alpha_j, \theta \right\rangle \tag{10}$$

The iCS system modeling can reveal the security mechanism of the iCS system against attacks. The attack success rate $P_{S,\theta}$ indicator of a attack task is used to characterize the effectiveness of the iCS system defense. The relationship between the indicator and the parameter is:

$$P_{S,\theta} = f(M_i, \Delta t_i, VUL_i, q_{vul_j}, t_{vul_j}, \alpha_j, \theta) \tag{11}$$

Then, the attack success rate of the attack task (θ steps) in the enhanced mode of the iCS system is:

$$P_{S,\theta} = \left\{ \frac{\sum_{i=1}^{N} \left[\alpha_i \cdot q_{vul_i} \cdot I(vul_i) \cdot \sum_{\substack{1 \le i \le m \\ 1 \le j \le N}} \delta_k(vul_{j_i}) \right]}{m} \right\}^{\theta} \tag{12}$$

Where, m is the number of changes of M_i combinations. The function $I(vul)$ reflects the impact of the dynamic scheduling in the system:

$$I(vul) = \begin{cases} 1 & t_{vul} \le \Delta t_i \\ 0 & t_{vul} > \Delta t_i \end{cases} \tag{13}$$

Where, the function (4-9) reflects the result of the MDP decision algorithm, and each executor needs to meet k-order vulnerability consistency required by the decision algorithm $f_i(vul_j)$ in order to a successful attack:

$$\delta_k(vul_{j_i}) = \begin{cases} 1 & if \quad f_i(vul_j) = True \\ 0 & else \end{cases} \tag{14}$$

V. Experimental Results and Discussion

The initial goal of the design of 5GC (fifth-Generation Core) was to become a cloud native application, and advances the mobile network to shift from a network for connectivity to a network for services [1][36]. In a end-to-end slicing network, the 5GC is responsible for the scheduling and management of global resources, and is responsible for managing the global network topology and all industry requirements. The mobile network has evolved to an open, sharing and on-demand 5G ecosystem from a closed small garden of the 4G era. The 5GC is the key to enabling thousands of industries, there are higher security risks compared with closed rigid legacy networks. The 5GC is a typical application scenario of the cloud native environment, and is used to quantitatively evaluate the system overhead of iCS. The testbed includes Intel Xeon CPU E5-2690 v4 2.6GHz server, forwarding plane using SR-IOV network transmit-receive mode and control plane of DPDK (Data Plane Development Kit) transmit-receive driver. To enable hardware hyper-threading, each vCPU should correspond to a hardware hyper-threading. The entire testbed of 5GC consists of an infrastructure layer (NFVI), a packet forwarding plane, a control plane and a management plane. In order to evaluate the impact of the iCS system on the 5GC overhead, the iCS components are loaded to the original 5GC such as DHR foundation, SDA, and MDP. Computing resources are the most important metric for NFVI resource overhead. In our experiments, the value of the computing resource overhead under specific traffic loads is calculated. A typical traffic model is set: the traffic per subscriber is 50 kbps during busy hours, the bearer per attached subscriber is 1.6, and attach and detach attempt times per subscriber are all 1 during busy hours. The experiment of the 5GC overhead reflects also the resource overhead ∂A_d generated by defense actions on a macro scale.

A. The overhead of 5GC with iCS basic mode

In the iCS basic mode, the VNFs executor and network communication parameters are shuffled in the spatiotemporal dimension by DHR foundation. It includes four shuffling strategies that can be deployed in the system independently or in any combination. The shuffling period is set to one hour. The DHR foundation with dynamic scaling capability uses a 1+1 backup method. The shuffling period is set to one hour.

a) The corresponding heterogeneous replica is randomly loaded when VNFs scaling in&out

The DHR foundation component is loaded in the MANO system, which is responsible for the control of the iCS dynamic shuffling strategy. When a VNF meets the scale out condition, a different executor is selected from heterogeneous functionally equivalent pool of the VNF, so that multiple heterogeneous versions of the same VNF are running in spatiotemporal dimension.

Potential impact components include DHR foundation and MANO. Following the original scale in&out mechanism, when the KPI reaches a certain threshold, a new heterogeneous VNFs component scales out. Compared with the original scale in&out mechanism, no new scheduling overhead is introduced;

the increase in the number of heterogeneous versions will increase storage and management overhead.

b) Periodically shuffle all of VNF executors

In a shuffling cycle, all currently running VNFs executors are replaced in order to have multiple heterogeneous executors in the time dimension. Potential impact components include DHR foundation, MANO and VNF. Experiments show that the resource overhead of DHR foundation depends on the total number of VNF executors running in the system. When the number of VNF executors increases, the resource overhead of DHR foundation also increases linearly. In order to meet the reliability requirements, all components in the 5GC system adopt the active/standby or load sharing method. This allows us to adopt some clever scheduling strategies. When implementing the operation of the VNF executor replacement, the impact on the resource overhead of the VNF itself is reduced to a very low level: 1) For VNFs that use load sharing, a session access is restricted in the VNF executor before a replacement operation, and then perform the replacement operation when the traffic on it is reduced to a minimum. 2) For VNFs in active/standby mode, heterogeneous executors are replaced on the standby VNF. After the VNF finishes the active/standby switch over, the new standby VNF is replaced with a heterogeneous executor. Through the above strategies, both service continuity can be ensured and excessive resource overhead of VNF can be avoided.

c) Periodically traverse VNF for live migration of services

DHR foundation drives MANO to periodically migrate VNF Components on each VM, which can prevent side-channel attacks. The migration time of each virtual machine depends on the memory size of the VM and the service scale during the migration. When the service scale is large, the VM will constantly generate ”dirty memory”, so that the live migration cannot be ended. We set the maximum downtime of the migration to 1200s. After this timer is reached, the virtual machine on the source host is forcibly suspended to finish the live migration. It may cause service interruption when the time of forced live migration is too long.

Potential impact components include DHR foundation, MANO, VIM (Virtualised Infrastructure Manager) and Hypervisor. It does not cause significant resource overhead on the management plane that the implementation of periodic live migration of VNF through the instruction chain of DHR foundation, MANO, and VIM. However, the resource overhead of the hypervisor node responsible for VM migration has increased significantly, and it is severely limited by the ratio of the current service traffic to the total system capacity. 1) When the service scale is 10%, all VM can be live migrated within about 300s, and the service interruption duration is less than 500ms; 2) When the service scale exceeds 50%, some VMs, such as those responsible for session access management and signaling processing, need to trigger forced suspension to finish the VM migration work, which will cause about 5s of service interruption; 3) When the service scale is 100%, forcibly suspending a VM will cause the service to be disconnected, the service needs to be re-established, and the live migration fails.

In actual engineering, a VNF dynamic live migration should performed when the traffic is low. At this time, the impact on the service is small; if the VNF live migration policy is executed when the busy service, it may cause the policy execution to fail and affect service continuity. The average CPU increase of the Hypervisor at 100% of the service traffic is recorded to evaluate the maximum value of the resource overhead brought by the strategy in this paper.

d) Periodically change communication interface parameters

The VNF interface communication parameters on the forwarding path are dynamically changed to evaluate the impact on the system overhead. The IP address of VNF/VNFC in the testbed is configured by SDN controller, and regularly changed through it. SFC (Service Function Chain) on the forwarding plane is used to forward services through the GRE-NSH tunnel. To ensure service continuity, the service chain needs to be updated synchronously to keep the forwarding topology relationship unchanged. There are various value-added services on the Gi interface of the 5GC, forming a complex forwarding network with the components such as UPF (User Plane Function), NAT (Network Address Translation), and firewall. This model has higher reference value because of in a real service network.

Potential impact components include SDN controller, vSwitch and VNF. The security policy is triggered by DHR foundation, but the implementation mainly depends on the SDN controller. The main impacts include:

1) Cause the change of ARP (Address Resolution Protocol) table in the SDN network, and then trigger the update of the OpenFlow forwarding flow table;

2) ARP message update and DHCP (Dynamic Host Configuration Protocol) message update will also cause packet-in/packet-out operation in a openflow path;

3)The link list of SFC needs to be updated and synchronized to the relevant SFF (Service Function Forwarders) nodes.

The overhead of scheduling depends on the change frequency of the communication parameters and the system scale. The SDN controller has a good processing capability of the southbound interface, experimental results show that the above operations have minimal impact on the SDN controller. According to actual statistics under 100G network traffic, each interface is changed once in 60s, when constrain the scheduling in one vCPU, the scheduling overhead has increased by 27% on average. Similarly, the flow table is updated on the vSwitch when the VM address changes, due to the small amount of updates, the scheduling overhead can be ignored.

After the VNF address changes, all SFF nodes related to VNF need to update the flow table of service chain. Because of the use of SR-IOV method, the SFF function is implemented in VNF node. That is, the components such as UPF, NAT, and firewall need to update the GRE-NSH tunnel encapsulation. The forwarding path does not change substantially, the SFF only needs to update the corresponding forwarding table and tunnel encapsulation. The experimental results show that, this part of the work does not introduce appreciable scheduling overhead. This also reflects the advantages of SFC in simplifying and quickly adjusting the routing of service traffic.

Table III shows the impact of four scheduling strategies in basic iCS mode on resource overhead. Among them, strategy

TABLE III: The impact evaluation of resource overhead in iCS basic mode

Strategy	Major components affected	Influence Ddegree
a	DHR Foundation	Slight
b	DHR Foundation	Medium
c	Hypervisor	Large
d	SDN controller	Slight

b and strategy c have obvious impact on resource overhead.

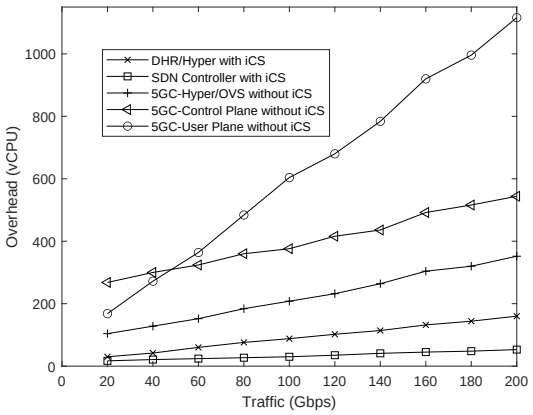


Fig. 7: Overhead in iCS basic mode

Fig. 7 shows the overhead in the computing resources of DHR foundation, Hypervisor, and SDN controller when the four iCS strategies are enabled at the same time. It can be inferred that as long as the live migration operation of VNF service is not performed when busy service, the four typical strategies in the iCS basic mode will not introduce significant resource overhead to the 5GC system.

B. The overhead and cost of 5GC with iCS enhanced mode

The testbed of iCS enhanced mode is enhanced in three parts based on the basic mode. On the management plane, the management and scheduling capabilities is added for SDA/MDP and component replicas in the DHR foundation. On the control plane, an AMF (Access and Mobility Management Function) with three heterogeneous replicas is used as the evaluation object, and corresponding SDA_C and MDP_C components are developed. The SDA_C is responsible for message dispatch, and MDP_C for output decisions, it determine which output of the three replicas will be used as the input of the next phase. On the forwarding plane, a service chain consisting of UPF/NAT/firewall components is used as the evaluation object. in addition to these VNF component replicas, corresponding SDA_U and MDP_U components are also developed to perform overall traffic dispatch and decision-making on the service chain. These components with automatic scaling feature adopt N+1 backup method.

a) The resource overhead of SDA&MDP

The communication between network elements in control plane is similar to bus communication, so SDA_C and MDP_C is added to the control plane bus. When the source VNF sends a message to the control plane bus, the message is checked and decided through the MDP_C component, including key elements for IP, SCTP, and GTP sessions, and sent by the SDA_C component to multiple heterogeneous executors of target VNF.

It is mainly service chain communication in the user forwarding plane. In order to reduce system overhead and delay, SDA_U and MDP_U components are added only to the service chain entrances and exits of components such as UPF, NAT, and firewall, to dispatch and decide the entire service chain. In the experiment, the key elements in the GRE-NSH tunnel and the service session are mainly checked and decided.

The resource overhead of SDA and MDP is related to the number of VNF heterogeneous executors running. Fig. 8 shows the resource overhead of components such as SDA_C, MDP_C, SDA_U, and MDP_U as service traffic changes in the environment of two heterogeneous executors. It can be seen that SDA and MDP's resource overhead on the user plane is significantly larger than the control plane, even in the experiments we designed, the user plane introduced less decision points than the control plane. This also shows that the resource overhead of SDA and MDP has a high correlation with service traffic. In addition, it shows that the resource overhead of MDP is slightly higher than that of SDA. The reason is that MDP needs to buffer and compare the packets. These I/O operations increase the resource overhead.

b) The resource overhead of VNF replicas

The system overhead of the component replicas is basically equivalent to the corresponding VNF. In iCS enhanced mode, SDA and MDP are only part of the overall resource overhead increase, and a large part is the overhead introduced by the VNF heterogeneous executor. It is compiled that three static heterogeneous versions of the control plane and user plane VNFs, and loads two or three of them. Regarding the control plane and user plane, no matter whether the heterogeneous executor is constructed by using VNF or service chain as the unit, the resource overhead finally reflected is twice or three times the original. Fig. 8 shows the comparison of two heterogeneous executors (identified by "iCS2"), three heterogeneous executors (identified by "iCS3") in the iCS enhanced mode, and original 5GC resource overhead. Due to the addition of heterogeneous executors on the control

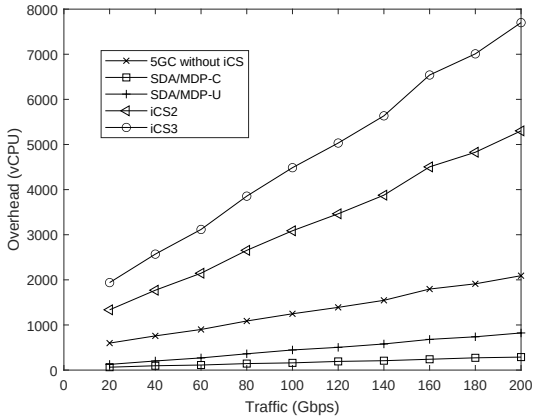


Fig. 8: Overhead in iCS enhanced mode

plane and user plane, and the introduction of SDA and MDP, the 5GC resource overhead in the iCS enhanced mode has doubled, even in scenarios with large service traffic.

In addition to the system overhead increased such as shuffling and replicas running, the development of components such as DHR foundation, SDA/MDP, especially the development of heterogeneous replicas of service chain components which is multiple functional-equivalent, behaviorally-different software versions (code implementation), has considerable development costs. Using the firewall as a VNF example, we used two teams to develop this feature completely independently. In fact, the development team of the firewall replica still has considerable development costs. Even if the open source foundation is adopted, the development architecture of the commercial firewall team is learned, and the release quality is reduced. The Fig. 9 shows a comparison of the development costs (in thousands of lines of valid code) introduced by the iCS system during the development phase, as well as the associated deployment costs.

C. An adversarial example of network malicious sniff attacks

a) Defense mechanism provided by iCS

The vast majority of network attacks generally take network malicious sniffing as a pre-step. In network malicious sniffing, the attackers scan the cyber space to find surviving hosts that can be connected and identify the vulnerabilities that can be exploited. The existing cloud native network has static characteristics, and the network sniffing tools (such as Wireshark and Nmap) can be easily obtained, which give attackers a great and asymmetrical advantage, and make it difficult to detect and prevent a network malicious sniffing. The iCS increases the diversity, heterogeneity, dynamics and randomness of the cloud native network. It can confuse attackers and prevent them from connecting to the important assets of cloud native network. The defense mechanism is reflected in:

1) Through the dynamic shuffling and orchestration management of the VNFs, the vulnerable hosts of a system are randomly replaced and not used as service processing nodes.

2) Periodically changing communication interface parameters in the network topology [37], the network sniffing and scanning phase of the attack chain is interrupted by changing the attack surface, ultimately leading to attack failure.

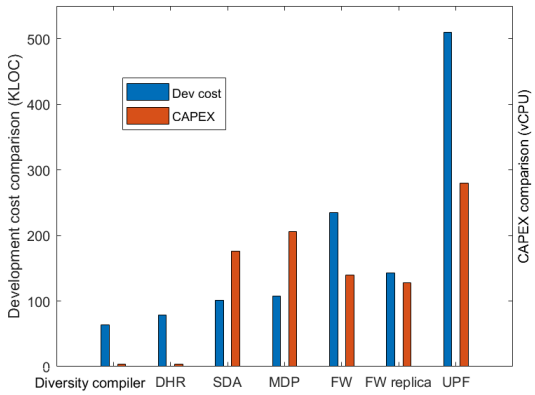


Fig. 9: The comparison of the development costs and the comparison of deployment costs at 60G traffic

The specific realization goals in iCS are: 1) The survival vulnerable hosts mixed in service processing nodes are randomly shuffled to disrupt the consistency of attack effects, so that the attacker mistakenly believes that there are no vulnerable hosts or the vulnerable hosts frequently downtime; 2) The decoy nodes are set up in the system to generate false scan results, which make the attacker mistakenly believe that a vulnerable host has been discovered, exposing subsequent attack methods. The iCS is based on the existing trapping ideas [38], and its difference is that it makes full use of the existing ability of the cloud native system to realizes the automatic deployment and dynamic adjustment of the decoy system.

First, the centralized monitoring in DHR foundation can periodically collect network statistics to monitor malicious scanning behavior. Then, the attack strategy decision graph models are set up in the security controller, which is trained based on the statistical data to model the scanning characteristics. Then, based on the scanning characteristics of attacker, a corresponding proactive protection strategy is constructed. The strategy uses the existing dynamic deployment and scheduling mechanism to transform the network topology and interface parameters among the service processing nodes, vulnerable hosts, and honeypot hosts, so that the interface relationship and network topology of VNF replica present dynamic characteristics. Finally, the SDNO (SDN Orchestrator) inserts flow table rules to the SDN network, and the NFVO issues dynamic service migration rules to the NFVI, the proactive protection strategy above-mentioned is executed.

b) Threat model and experimental evaluation method

The malicious scanning scheme in the experimental system is based on the following assumptions, assuming that at least one host is initially controlled and certain scanning tools are installed on the host. During sniffing, more and more hosts are controlled as the infected hosts, and then the infected hosts are used to send more sniffing requests. Since fast scanning is easy to be detected due to its obvious characteristics, it is generally considered that all malicious scanning processes are covert, slow speed and persistent [39]. The Libnmap library is used to simulate the scanning behavior of the attacker. It is a python library that enables the python developers to manipulate the nmap process and data. The scanning attack strategies used in established threat models include: uniform scanning, local-preference scanning, sequential scanning, divide-conquer scanning, and adaptive scanning [40][41]. The scanning space jump probability of sequential scanning is set to 0.4. The scanning speed is set to 100 addresses per second, and repeated scanning of the addresses is excluded. The 5GC cloud native environment mentioned above is set as the attacking target as a test bed, which includes the NFVI virtual network composed of Openflow switches and virtualization hosts. The number of virtualization hosts is set to 20000, which is managed by the NFVI resource pool. The risk density is set to 1/4, 5000 hosts among all hosts are vulnerable hosts, and 5000 hosts are decoy nodes. In the scheduling strategy, the vulnerable hosts and the normal hosts are classified as the same type of hosts without labeling, in order to simulate the situation where the network

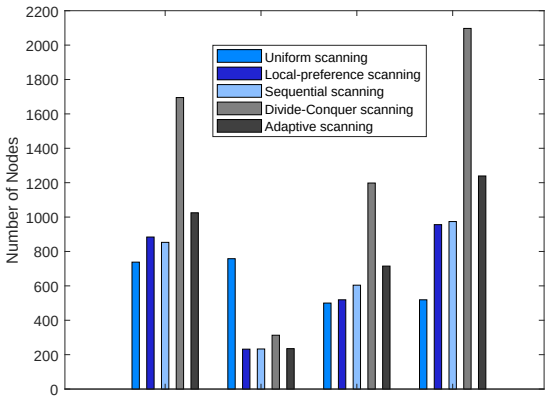


Fig. 10: The average number of vulnerable hosts or decoy nodes found in each statistical interval

threat is hidden in the service more realistically. The decoy nodes are explicitly labeled. The operating scenario is set as that half of the hosts in the resource pool bear the service load, and the other half are in the idle state. The network topology is canonicaltree and contains 256 subnets; each subnet is a class C network and contains 254 network addresses. The host addresses in the subnet are evenly distributed, which are similar to the actual BGP routing subnet, and the change interval of the parameters is set to 10 seconds.

In our experiments, the same experimental environment and dynamic network topology transformation interval were used to compare iCS system with existing address randomization schemes [42][43]. The main goal of the experiments is to evaluate the iCS's defense against malicious network scanning. The evaluation indicators are reflected in: 1) the number of vulnerable hosts found by the attackers in a scan; 2) The number of decoy nodes found by the attackers in a scan.

c) Experimental results

In the time range of 600 seconds, at 20 second intervals, the number of vulnerable hosts found by the attackers is counted, and the number of decoy nodes found by the attackers is counted. Fig. 10 shows the average number of concerned nodes in each interval when the network topology is dynamically changed by different methods under various scanning strategies. From the data in the Fig. 10, we can find that, except in the case of uniform scanning, when using iCS for defense in cyberspace, a smaller number of vulnerable hosts are scanned by attackers. It shows that the iCS can more effectively reduce the number of vulnerable hosts found during the sniffing. The main reason is that the dynamic transformation strategies of the iCS can be adjusted according to the actual scanning mode of the attackers. Meanwhile, it needs to be admitted that the method in this paper does not have superiority under the uniform scanning strategy, because the uniform scanning has no the regular rules and cannot be predicted. A new research is needed to solve the issue. The good news is that the uniform scanning is not often used because inefficiency. Similarly, a larger number of decoy nodes are scanned by the attackers , except in the case of uniform scanning.

According to the experimental results, it can be seen that the iCS effectively reduces the number of vulnerable hosts and increases the number of decoy nodes found by the attackers. Compared with the existing randomization methods, the method in the iCS has better performance. What needs special explanation is that the dynamic transformation strategies of the network topology can evolve smoothly with the development of the iCS framework. The strategies can be deployed as an application component in the DHR foundation.

D. The iCS impact on attack success

To understand the security gain provided by iCS system, we perform experiments with a notional VNF component named Buffer and two real world exploits. On the testbed, there are five heterogeneous replicas of Buffer: B_1, B_2, B_3, B_4 and B_5. The B_2 and B_3 are B_1 replicas formed by diversity compiler. Buffer runs for a random amount of time on a platform before being shuffled by a different one.

The attack goal is to triggers a service interruption for some time T. The first attack is the $R2Libc$ exploit which triggers a service interruption. Only the B_1 is vulnerable to this attack. The second attack is the Socket Pairs exploit which triggers a garbage collection vulnerability in net/unix/garbage.c (CVE-2010-4249) to saturates the CPU usage and file descriptors [44]. The B_4 and B_5 replicas are vulnerable to the attack. B_1, B_2 and B_3 are not vulnerable to the exploits. Without loss of generality, the interval between scheduling (d) is randomly and uniformly selected within 40-60 seconds. One or both exploits are lunched at random times during each trial. As a result, 0-3 replicas can be compromised (zero when the exploit is not effective against the replicas set and three when both exploits are available and B_1, B_4, and B_5 are in the replicas pool).

In the iCS basic mode, $N\in$(1, 2, 3, 5) heterogeneous replicas are selected. In the case of N=1 (baseline), it is a situation without iCS protection and is considered a baseline for comparison. For each trial, Buffer is randomly shuffled from those 5 replicas without immediate repeat. In the iCS enhance mode, the SDA/MDP corresponding to Buffer are automatically deployed. The value of the metric is calculated and presented these results in Fig. 11. The limited duration has an effect on attack results. Let d be the duration of the transaction on a platform, that is the duration when an

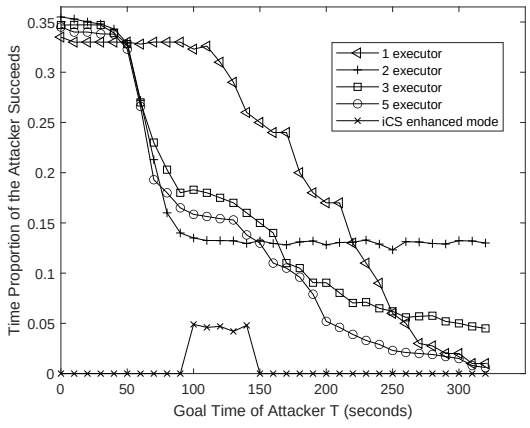
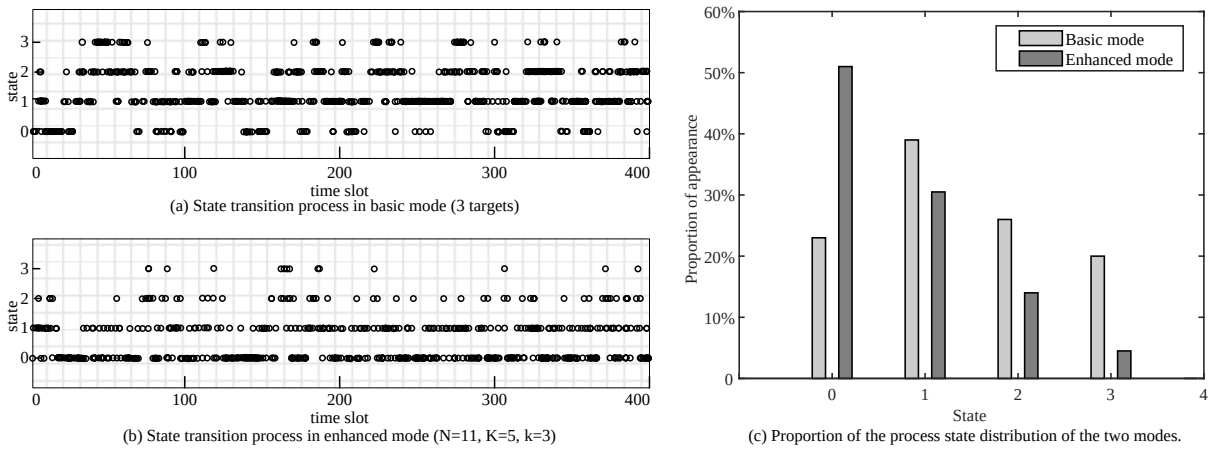


Fig. 11: Measurement results of the impact of iCS on attack success



(a) State transition process in basic mode (3 targets)

(b) State transition process in enhanced mode (N=11, K=5, k=3)

(c) Proportion of the process state distribution of the two modes.

Fig. 12: State transition process of the two modes

executor of Buffer continues to run, T be the period of service interruption. If $T>d$, the attack goal can never be achieved. For $T<d$, the attack can only success if it starts early enough during the d-T interval. The iCS enhanced mode always maintains a more beneficial and stable defense effects. Only when B_4 and B_5 are selected at the same time, the attacker has a chance of success.

In order to describe iCS security more comprehensively, the attack-defense game process is simulated by Markov Chain Monte Carlo method, and the simulation platform is Matlab R2014a. The basic mode uses the multiple-target hiding MTD game [45], where the targets number is 3. The enhanced mode adopts mimic defense, the number of redundant executors K=5, the schedulable space of executors N=11, the executors number that the attacker needs to control to achieve the attack goal is k=3, that is, the defense tolerance is T=3, and the Poisson process parameters are equal in the game for both. The actions taken are mutually exclusive and the probability is 0.5. The game process between the attacker and the defender may be instantaneous in an actual scenario, the simulation process defines that each state transition process occurs within a time-slot. According to the previous description, the attack-defense game process starts from state S_0 to state S_3: breakthrough system, which causes substantial damage to the system. In the experimental simulation, the $S_0 \rightarrow S_3$ state transition process is sampled for both the basic mode and the enhanced mode.

The detailed state transition process results are shown in Fig. 12a&b for 400 samples. It shows that the basic mode of multi-target hidden MTD is easier to migrate to S_3, while the enhanced mode of mimic defense reaches S_3 for the first time after 67 state transitions, and it is more sparse in the global distribution. The state distribution ratio of the two modes is shown in Fig. 12c. It can be seen that the state is mostly concentrated in the intermediate state in the basic mode, while it is concentrated in S_0 in the enhanced mode, and the S_3 distribution is significantly less than the basic mode. Then from the perspective of security defense, the more the state distribution is concentrated on S_0, the better the effectiveness of system defense. The probability of the attacker controlling multiple targets in the mimic defense model is smaller than the basic mode. The tolerance to attacks in the enhanced mode is reflected in the fact that the heterogeneous redundant executors jointly complete the mimic judgment mechanism. The defense mechanism provides a guarantee for the robustness of output, and the abnormality of individual executors has a low impact on the overall output.

E. Case analysis of attack-defense game model

The experimental simulation quantitatively verifies the defense ability of iCS based on the previous mature model. In the attack-defense game model of cloud native environment shown in Fig. 6, a MANO is used as the defensive target of iCS, and its multiple replicas are deployed as a redundant executor to work together. It assumes that the attacker implements advanced persistent threats, the attack can bypass the network firewall and penetrate into the server cluster in the cloud native environment, and can access the MANO server through the network. The attack-defense game process is simulated at the state of $S_0 \rightarrow S_3$. The size of the redundant executor is set to K=5, and the majority consistency judgement condition is set to k=3 to achieve consistency. Assuming that in addition to 0-day vulnerabilities, the system still has known common vulnerabilities. The Nessus as vulnerability scanning tool is used to scan the system for vulnerabilities. The atomic attack of the vulnerability scan in the initial state S_0 are shown in Table IV. Except for the iCS protection measures, no other special protection measures are deployed for the vulnerabilities. In view of the existing experiments to verify the defensive effect of network sniffing, effective defensive measures have not been deployed to prevent all sniffing behaviors, and the attacker can scan and discover system vulnerabilities. The attacker can only exploit one system vulnerability that has been scanned for each action, and will not repeatedly sniff the acquired system vulnerabilities. The experiment focuses on the robustness of system defense against vulnerabilities. In actual scenarios, the success rate of such attacks is much lower due to the complexity of sniffing and the dynamic transformation of communication parameters.

Table V lists the strategy set of attack-defense parties, the main proactive defense methods include ASLR (Address

TABLE IV: Atomic attack attribute and information

No.	Vulnerability attribute and information	CVSS scoring	Virtual machine or replica
vul_1	AV:L/AC:H, CVE-2017-1000364. Remote memory corruption	7.4	C_4,C_5
vul_2	AV:N/AC:H, CVE-2017-0148 SMB. Remote Code Execution	8.1	C_1
vul_3	AV:L/AC:L,CVE-2017-0299 KASLR. Local Information Disclosure	5.0	C_1
vul_4	AV:L/AC:H, CVE-2017-1000367. Local information disclosure and privilege gain	9.8	C_2, C_5
vul_5	AV:N/AC:L, CVE-2016-10229. Remote Code Execution	6.4	C_3
vul_6	AV:L/AC:L, CVE-2016-1247. Local web-root privilege gain	7.8	C_3

TABLE V: Strategy collection of attack-defense parties

State	Attack-defense strategy collection
S_0	a = {a_{01}-a_{03}: buffer overflow, code injection, ε }
	d = {d_{01}-d_{03}: ASLR, ISR, ε }
S_1	a = {a_{11}-a_{13}: code injection, ROP, ε }
	d = {d_{11}-d_{15}: ASLR, ISR, patch upgrade, shuffling, ε }
S_2	a = {a_{21},a_{22}: privilege gain, ε }
	d = {d_{21}-d_{23}: diversified compilation, shuffling, ε }
S_3	a = {a_{31},a_{32}: ROP (Return-Oriented Programming), ε }
	d = {d_{31},d_{32}: diversified compilation, ε }

Space Layout Randomization), ISR (Instruction Set Randomization), diversity compilation, and other ways to improve system heterogeneity. CVSS (Common Vulnerability Scoring System) is an industry public standard designed to evaluate the severity of vulnerabilities and help determine the urgency and importance of the required response. CVSS 3.0 is used to evaluate the system vulnerabilities. According to the information such as the atomic attack database, defense strategies, and state transition probability, the specific attack-defense process is formally expressed as Fig. 13. The $P_{i,j}(a, vul, d)$ in the figure is obtained according to the anti-attack model of iCS in SEC IV, which represents the transition probability from state S_i to state S_j under the combination of (a, vul, d) attack-defense strategies. Although some defensive actions have played a role, there are still possible attack paths that cause the state transition because of the dependency of the vulnerabilities, and finally the attack target state S_3 is reached with a low probability. Combining the related research on the success probability of attacks [35] and the peculiarities of iCS defense, the quantification profit matrix (R_i^a, R_i^d) in each state is obtained according to the profit function Equation 6 and Equation 7 in section IV.

R_0^a	d_{01}	d_{02}	d_{03}
a_{01}	-6	29	34
a_{02}	29	-6	34
a_{03}	-6	-6	0

R_0^d	d_{01}	d_{02}	d_{03}
a_{01}	10	-25	-25
a_{02}	-25	10	-25
a_{03}	10	10	0

R_1^a	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}
a_{11}	29	-6	0	0	34
a_{12}	32	32	20	0	34
a_{13}	-6	-6	0	0	0

R_1^d	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}
a_{11}	-25	10	-20	25	-25
a_{12}	-28	-28	-40	25	0
a_{13}	10	10	-20	25	0

R_2^a	d_{21}	d_{22}	d_{23}
a_{21}	-11	-15	34
a_{22}	-11	0	0

R_2^d	d_{21}	d_{22}	d_{23}
a_{21}	10	-5	-25
a_{22}	10	25	0

R_3^a	d_{31}	d_{32}
a_{31}	-11	34
a_{32}	19	0

R_3^d	d_{31}	d_{32}
a_{31}	9	-25
a_{32}	-21	0

Analysis of attack paths and defense strategies: The effectiveness and diversity of the action vector d of defender have a certain impact on defense capabilities. The defenders need to choose appropriate defensive actions against specific attack behaviors. For example, the ASLR cannot effectively defend against the attack of local web-root privilege gain in the instance. The attack path of $S_0 \rightarrow (a_{02}, vul_6, d_{01}) \rightarrow S_1$ reflects the failure of this defense action. The implementation of effective defense actions means that the number of executors controlled by the attacker decreases, and the system will return to the initial state S_0 until the defense strategy covers all of them. The defense strategy also has a certain impact on the defense capability. The diversified compiling in subsequent states to randomize the target code can effectively defend against the code injection attacks. In the case of a single executor, certain types of attacks caused by vulnerabilities may be unavoidable. For example, the attack path of (a_{02}, vul_4, d_{01}) have no effective defense countermeasures, and the state transition is directly from the initial state to the final state, that is, the single executor is successfully attacked. In contrast, the heterogeneous redundancy features under iCS can alleviate the risk caused by a single executor with vulnerability, on the one hand, it can effectively increase the complexity that the attacker controls the entire system, and on the other hand, the feedback judgement mechanism of the mimic defense system effectively reduces the success probability of attack.

It can be seen from the profit matrix that the defender has higher profits when the ASLR and ISR against specific types of attack, but the types defended against are limited. Thus, the defender chooses the shuffling method against vulnerabilities with a greater probability. On the other hand, the patch upgrade method can refresh the system status and is effective for most attacks. However, it can be seen from the profit matrix that its defense profits are not high for a single vulnerability because of the large performance and availability loss. When the common vulnerabilities of multiple executors are exploited, such as vul_1 and vul_4, the attackers will often get more profits and thus have a higher probability of being exploited.

Limitation analysis: The case analyzes the orientation of

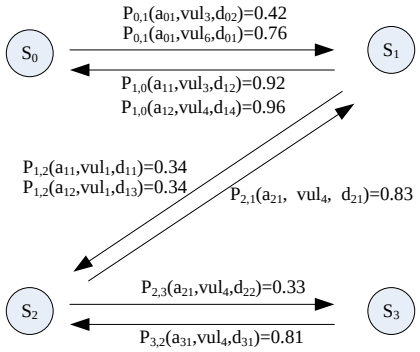


Fig. 13: The instance of state transition path and probability distributions

dynamic strategies and the role of the action vector in the attack-defense process, but fails to reflect the specific diversity direction of the defense action vector d. The state transition diagram is not complete, some low-profit strategies are ignored. For the shared vulnerabilities of multiple executors, the state diagram failed to reflect more advanced cooperative attack features of it because of relatively limited number of shared atomic attacks.

VI. Conclusion

With a traditional static model of telecommunications service, it is difficult to deal with the uncertain factors and differentiation of massive mobile Internet services. Faced with more complex and changing requirements in the differentiation scenarios of vertical industry, the 5G slicing network requires flexible and dynamic network resource allocation and more efficient resource utilization, as well as with better openness and greater innovation capabilities [46]. In addition to the traditional threats, the new risks such as openness and multi-tenancy are further superimposed on the cloud native environment that 5G slices rely on, and thus it is urgent to provide intrinsic security and proactive protection against the unpredictable attacks, and to implement dynamic on-demand customization of security policies.

The iCS foundation unifies MTD idea and mimic defense theory in a cloud native framework. Due to cost and efficiency constraints, we must maximize the use of the existing mechanism in the framework based on service oriented network virtualization, and then implement seamless integration and symbiosis evolution between security and services, to achieve proactive protection against internal and external threats.

Based on the mimic defense theory and the MTD idea, the architecture scheme of iCS is proposed as a public support platform. Under the constraints of the cost and the efficiency, the iCS system proposes a feasible approach to the seamless integration and symbiosis evolution of proactive security and cloud native environment.

The iCS paradigm has a generalized robust, which implements an intrinsic and proactive defense mechanism of cloud native environment. The next step is to solve two issues: On the one hand, the heterogeneity of components, mutations and shuffling strategies have a great impact on the defense effect of the system. It is necessary to establish a heterogeneous evaluation mechanism, and implement the selection algorithm for optimal proactive defense strategy based on the attack-defense game model. On the other hand: the attack traces presented by an attacks against the target system usually have certain characteristics, and this characteristic can be captured by the judgement feedback mechanism to find toxic replicas and analyze the attack type. The existing solutions does not fully reflect the role of the feedback mechanism.

Acknowledgment

This work is supported by the National Key R&D Program of China (2020YFB1804705), the National Science and Technology Major Project (2017ZX03001018), the Key R&D Program of Zhejiang Province (2020C01077, 2021C01036, 2020C01021), and the Major Scientific Project of Zhejiang Lab (2018FD0ZX01).

References

[1] Sharma S, Miller R, Francini A, et al. “A Cloud-Native Approach to 5G Network Slicing” *IEEE Communications Magazine*, vol. 55, no.8, 2017, pp. 120-127.

[2] Nikaein N , Schiller E , Favraud R , et al. “Towards a Cloud-Native Radio Access Network[M]// Advances in Mobile Cloud Computing and Big Data in the 5G Era,” *Springer International Publishing*, 2017.

[3] Belbekkouche A, Hasan M M, Karmouch A, et al. “Resource Discovery and Allocation in Network Virtualization,” *IEEE Communications Surveys and Tutorials*, vol. 4, no. 4, 2012, pp. 1114-1128.

[4] Taleb T, Ksentini A, Sericola B, et al. “On Service Resilience in Cloud-Native 5G Mobile Systems,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, 2016, pp. 483-496.

[5] Qiang W, Chunming W, Lin C, et al. “Open ICT-PaaS platform enabling 5G network slicing,” *Iet Communications*, vol. 13, no. 9, 2019, pp. 1242-1252.

[6] Samdanis K, Costaperez X, Sciancalepore V, et al. “From network sharing to multi-tenancy: The 5G network slice broker,” *IEEE Communications Magazine*, vol. 54, no. 7, 2016, pp. 32-39.

[7] Singh S, Jeong Y S, Park J H, “A Survey on Cloud Computing Security: Issues, Threats, and Solutions,” *Journal of Network & Computer Applications*, vol. 75, 2016, pp. 200-222.

[8] Ji X, Huang K, Jin L, et al. “Overview of 5G security technology,” *Science in China Series F: Information Sciences*, vol. 61, no.8, 2016, pp. 107-131.

[9] Balalaie A, Heydarnoori A, Jamshidi P, et al. “Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture,” *IEEE Software*, vol. 33, no.3, 2016, pp. 42-52.

[10] Claudio, A, Ardagna, et al. “From Security to Assurance in the Cloud: A Survey,” *Acm Computing Surveys*, vol. 48, no.1, 2015, pp. 1-50.

[11] Farahmandian S , Hoang D B . “Security for Software-Defined (Cloud, SDN and NFV) Infrastructures - Issues and Challenges,” *Eighth International Conference on Networks & Communications*, 2016.

[12] Rudd E M , Rozsa A , Manuel Gnther, et al. “A Survey of Stealth Malware Attacks, Mitigation Measures, and Steps Toward Autonomous Open World Solutions,” *IEEE Communications Surveys and Tutorials*, vol. 19, no. 2, 2017, pp. 1145-1172.

[13] E. Marku, G. Biczok and C. Boyd, “Securing Outsourced VNFs: Challenges, State of the Art, and Future Directions,” *IEEE Communications Magazine*, vol. 58, no. 7, 2020, pp. 72-77.

[14] Y. Liu, J. Peng, J. Kang, A. M. Iliyasu, D. Niyato and A. A. A. El-Latif, “A Secure Federated Learning Framework for 5G Networks,” *IEEE Wireless Communications*, vol. 27, no. 4, 2020, pp. 24-31, doi: 10.1109/MWC.01.1900525.

[15] R. F. Olimid and G. Nencioni, “5G Network Slicing: A Security Overview,” *IEEE Access*, vol. 8, pp. 99999-100009, 2020, doi: 10.1109/ACCESS.2020.2997702.

[16] Zhang M, Wang L, Jajodia S, et al. “Network diversity: a security metric for evaluating the resilience of networks against zero-day attacks,” *IEEE T. Inf. Foren. Sec.*, vol.11, no.5, 2016, 1071-1086.

[17] Lal S, Taleb T, Dutta A, et al. “NFV: Security Threats and Best Practices,” *IEEE Communications Magazine*, vol. 55, no. 8, 2017, pp. 211-217.

[18] Sengupta S, Chowdhary A, Sabur A, et al. "A Survey of Moving Target Defenses for Network Security," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, 2020, pp. 1909-1941.
[19] Kampanakis P, Perros H G, Beyene T, et al. "SDN-based solutions for Moving Target Defense network protection," *world of wireless mobile and multimedia networks*, 2014, pp. 1-6.
[20] Badr Y, Hariri S, Al-Nashif Y, et al. "Resilient and Trustworthy Dynamic Data-driven Application Systems (DDDAS) Services for Crisis Management Environments," *Procedia Computer Science*, vol. 51, no.1, 2015, pp. 2623-2637.
[21] Jiangxing Wu. "Cyberspace Mimic Defense," *Springer*, 2019.
[22] StephenW. Boyd, Gaurav S. Kc, Michael E. Locasto, et al. "On The General Applicability of Instruction-Set Randomization," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no.3, 2010, pp. 255-270.
[23] C. Cadar, P. Akritidis, M. Costa, et al. "Data Randomization," *Microsoft Research*, 2008.
[24] F. Rocha and M. Correia. "Lucy in the Sky Without Diamonds: Stealing Confidential Data in the Cloud," 2011 IEEE/IFIP 41st Int'l. Conf. Dependable Systems and Networks Wksps. (DSN-W), 2011.
[25] ETSI Published Specifications ETSI GS NFV-SEC 002: Network Functions Virtualisation (NFV); NFV Security, Cataloguing Security Features in Management Software.
[26] W. Yang and C. Fung. "A Survey on Security in Network Functions Virtualization," 2016 IEEE NetSoft Conf. And Wksps. (NetSoft), 2016.
[27] ETSI GS NFV-SEC 009 Network Functions Virtualisation (NFV); NFV Security; Report on Use Cases and Technical Approaches for Multi-Layer Host Administration, page 34.
[28] Somani G , Gaur M S , Sanghi D , et al. "Scale Inside-Out: Rapid Mitigation of Cloud DDoS Attacks" *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 6, 2017, pp. 959-973.
[29] Hongchao Hu, Chen F, Wang Z. "Performance Evalua-tions on DHR for Cyberspace Mimic Defense," *Journal of Cyber Security*, vol. 1, no.4, 2016, pp. 40-51.
[30] Wu J X. "Meaning and vision of mimic computing and mimic security defense," *Telecommunications Science*, vol. 30, no.7, 2014, pp. 1-7.
[31] Cox B, Evans D, Filipi A, et al. "N-variant systems: a secretless framework for security through diversity," *usenix security symposium*, 2006.
[32] Okhravi H , Hobson T , Bigelow D , et al. "Finding Focus in the Blur of Moving-Target Techniques," *IEEE Security & Privacy*, vol. 12, no.2, 2014, pp. 16-26.
[33] Manadhata P K, Wing J M. "An Attack Surface Metric," *IEEE Transactions on Software Engineering*, vol. 37, no.3, 2011, pp. 371-386.
[34] M. Dighriri, A. Saeed Dayem Alfoudi, G. Myoung Lee, T. Baker and R. Pereira, "Resource Allocation Scheme in 5G Network Slices," *32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Krakow, 2018, pp. 275-280, doi: 10.1109/WAINA.2018.00098.
[35] JIANG W, FANG B X, TIAN Z H, et al. "Evaluating network security and optimal active defense based on attack-defense game model," *Journal of Computer*, vol. 32, no. 4, 2009, pp. 817-827.
[36] Dab B , Fajjari I , Rohon M , et al. "Cloud-native Service Function Chaining for 5G based on Network Service Mesh," *IEEE International Conference on Communications (ICC)*, Virtual Conference, 2020DOI: 10.1109/ICC40277.2020.9149045.
[37] Yoon S, Cho J H, Kim D S, et al. "Attack Graph-based Moving Target Defense in Software-Defined Networks," *IEEE Transactions on Network and Service Management*, 2020, DOI 10.1109/TNSM.2020.2987085
[38] Cyber Warfare. "Building the Scientific Foundation," *Springer*, 2015.
[39] Stafford S, Li J. "Behavior-based worm detectors compared," *International Workshop on Recent Advances in Intrusion Detection*, Berlin, Heidelberg, 2010 pp. 38-57.
[40] Zou C C, Towsley D, Gong W. "On the performance of Internet worm scanning strategies," *Performance Evaluation*, vol. 63, no.7, 2006, pp. 700-723.
[41] Li Y, Chen Z, Chen C. "Understanding divide-conquer-scanning worms," *IEEE International Performance, Computing and Communications Conference*, IEEE, 2008, pp. 51-58.
[42] Achleitner S, La Porta T F, McDaniel P, et al. "Deceiving network reconnaissance using SDN-based virtual topologies," *IEEE Transactions on Network and Service Management*, vol. 14, no.4, 2017, pp. 1098-1112.
[43] Jafarian J H, Al-Shaer E, Duan Q. "An effective address mutation approach for disrupting reconnaissance attacks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no.12, 2015, pp. 2562-2577.
[44] Okhravi H, Riordan J, Carter K M, et al. "Quantitative Evaluation of Dynamic Platform Techniques as a Defensive Mechanism," *recent advances in intrusion detection*, 2014, pp. 405-425.
[45] MALEKI H, VALIZADEH S, KOCH W, et al. "Markov modeling of moving target defense games," *ACM Workshop on Moving Target Defense*, 2016, pp. 81-92.
[46] Barakabitze A A, Ahmad A, Mijumbi R, et al. "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Computer Networks*, 2020.

WU Qiang Professor, College of Computer Science and Technology, Nanjing University of Aeronautics&Astronautics, State Key Laboratory of Mobile Networks and Mobile Multimedia Technology. The main research field is including mobile network, industrial Internet, cyber security and data communications. The Chinese government honored him with the second-class National Science and Technology Progress Award in 2009, and the second-class National Technology Innovation Award in 2014.

WU Chun-ming Professor, College of Computer Science and Technology, Zhejiang University. In 2004, the Chinese government honored him with the first-class National Scientific and Technological Progress Award. In 2014, the Chinese government honored him with the second-class National Scientific and Technological Progress Award.

YAN Xin-cheng Chief System Architecture Expert of ZTE Corporation, has 20 years of experience in telecommunication. He is responsible for one of China science and technology major projects on 5G security.

CHENG Qiumei Cheng received the B.E degree in software engineering from Southwest University of Science and Technology in 2016. She is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University. Her research interests include software-defined network security, intrusion response system, traffic monitoring and reinforcement learning.