State of the Art and Research Challenges in the Security Technologies of Network Function Virtualization

Xiaochun Wu
Zhejiang Gongshang University

Kaiyu Hou
Northwestern University

Xue Leng
Zhejiang University

Xing Li
Zhejiang University

Yinbo Yu
Wuhan University

Bo Wu
Tsinghua University

Yan Chen
Northwestern University

***Abstract*—In recent years, network function virtualization (NFV) has drawn considerable attention due to its potential for service agility and low total cost of ownership. As the core of an NFV implementation, security issues in the management and control platform must be comprehensively addressed—from architectural concept to deployment. In this article, we first analyze the state of the security architecture based on ETSI-NFV, and then propose useful security practices for an NFV-based management and control ecosystem. To encourage future research, we also identify the ongoing research challenges and open security issues relevant to the NFV.**

Digital Object Identifier 10.1109/MIC.2019.2956712
Date of publication 5 December 2019; date of current version 25 March 2020.

The network function virtualization (NFV) offers a new way to replace expensive dedicated hardware appliances with generic servers that use software to design, deploy, and manage networking services. The NFV enables operators, carriers, and ISPs to quickly deploy new applications by provisioning supporting services rapidly compared with the three-to-six-month provisioning time required for hardware-based services. The service agility offered by the NFV technologies provides the ability to launch and decommission services more rapidly and efficiently than before; even customers can turn the NFV-based services ON or OFF, much like features. Recent NFV research results have greatly influenced the design of Internet of Things (IoT), 5G, and the cloud computing technologies.

Despite the above advantages, substantial challenges exist regarding the smooth use of an NFV-based solution, especially in terms of a secure network service ecosystem. To achieve a consistent approach and a common architecture for hardware and software infrastructure, the ETSI Industry Specification Group (NFV ISG) has issued a problem statement concerning the NFV security and the potential security vulnerabilities of an NFV. The existing security solutions for an NFV can be incorporated into the ETSI-NFV architecture to either strengthen the security of the NFV architecture itself or enhance the security of the service network by providing security policies. The provision of a secure network service ecosystem relies on a security control “brain” that needs to not only ensure the security of the physical and virtual operating environments but also prevent illegal code injection by isolating shared resources and forbidding unauthorized access. Meanwhile, the security of the service deployment transmission channel should be ensured such that the information is not stolen or tampered with during transmission.

The previous work in[1] was the first security paper to introduce the challenges and opportunities in the NFV security; however, it focused primarily on the NFV infrastructure (NFVI) security risks and best practices. Other NFV papers[2,3] that addressed security in part of the content included brief analyses of potential security threats in NFV. The NFV security issues comprise a very broad category that includes establishing reliable and credible self-examination and access mechanisms for underlying infrastructure, ensuring the isolation of different tenants when the virtualized network functions (VNFs) are abstracted, keeping the security of the instances of the VNF itself when the service chain changes dynamically, and using the VNF composition to provide better security services. We need to design rapid detection and correction of faults due to configuring misoperation. Meanwhile, vulnerability monitoring, rapid isolation, service recovery, etc., should also be designed for malicious security. We feel that these main security problems can be classified into three groups, security service deployment, monitoring, and trusted management, to elaborate on the implementation of the NFV security services. Nevertheless, a systematic summary of the core security technologies in the implementation of an NFV-based network is still lacking. This knowledge gap motivates this survey and summary of the current development state of solutions based on the above steps. Accordingly, our main contributions in this article are summarized as follows.

- To the best of our knowledge, this is the first work to provide a full overview and a comprehensive discussion of the security issues at all levels of the ETSI architecture.
- This article provides a detailed analysis of and a solution for the VNF orchestration and deployment, trust management, and monitoring.
- This article proposes new research directions.

BRIEF SECURITY OVERVIEW OF NFV INFRASTRUCTURE

To build a reliable security ecosystem, network functions (NFs) and network services in the NFV environments require global consideration of end-to-end security for network resources to ensure that security policies are dynamically updated and automatically aligned in a hybrid network consisting of physical and virtualized networks.

As shown in Figure 1, the security management functions in a security orchestrator should cooperate fully with the NFV architecture proposed by an ETSI to provide effective network security services. To realize safe and reliable network services, an NFV architecture should rely on the root trust booting from the underlying hardware resources in the NFVI, attestation between NFVI and the virtual infrastructure manager (VIM), effective physical and virtual network

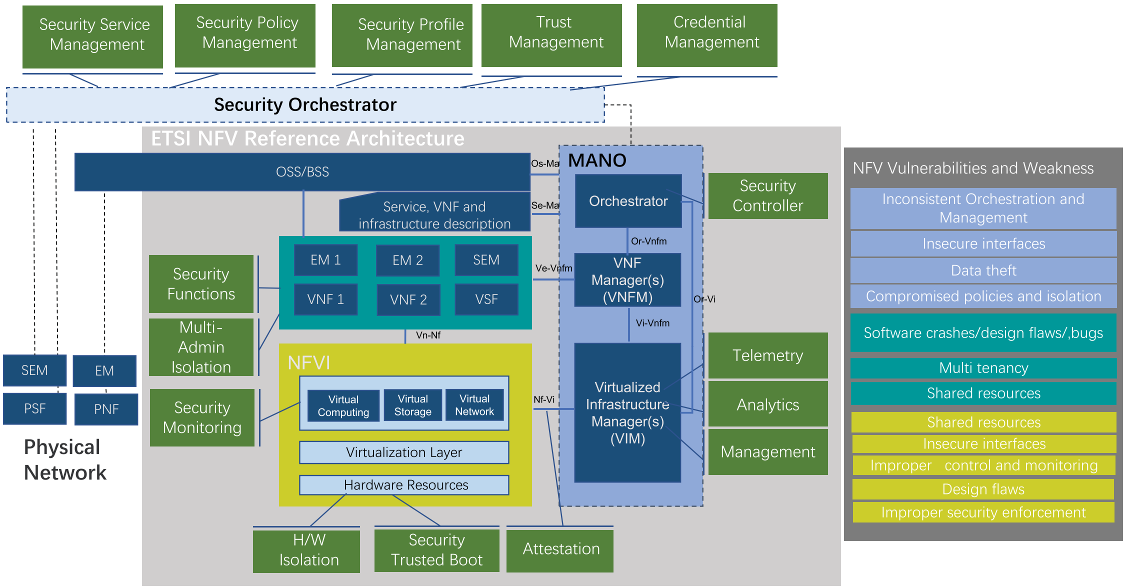

Figure 1. Security orchestration and management in the ETSI-NFV architecture.

monitoring, isolation of multiadmin managers, and reasonable security function orchestration. Not a single one of these conditions can be dispensed with. Nevertheless, as a core part of security control, the management and orchestration (MANO) should collaborate with the security orchestrator, perform secure network service orchestration and network element security management, adjust security policies, and manage the network service lifecycle based on feedback from the VNF manager (VNFM) and VIM. To avoid security risks from multitenancy and shared resources, VNFM should also add security function modules during the VNF instance initialization, VNF expansion/reduction, and VNF instance termination. The VIM's secure functions of telemetry, analytics, and management implement secure monitoring and failure reporting and provide a safe virtualized resource pool for high-level VNFMs and NF virtual orchestrators.

Many organizations and research institutes are rethinking the enhancement of security management in ETSI-NFV because it faces vulnerabilities and weakness at all layers, as shown in Figure 1. For example, the NFV-SEC group focuses on establishing trust and security in the NFVI and recently expended considerable effort toward proposing trusted computing (TC) technologies to protect the integrity of the sensitive components (e.g., BIOS, OS kernel) in the NFV environment.[1] Table I provides a summary of the NFV security projects and lists the research focus of each project. The existing solutions for NFV security can be incorporated into the ETSI-NFV architecture as security orchestrators. An individually abstracted security orchestrator can achieve an exhaustive end-to-end view of the security in a hybrid network. The orchestrator translates the defined security, strengthening the security configuration of the VM running a VNF, initiating the network service, etc. Furthermore, most projects involving current NFV orchestrators (e.g., OpenMANO, OpenBaton, OpenStack) are based on the ETSI-NFV reference architecture, which helps manage the VNF lifecycle and orchestrate infrastructure resources to support end-to-end network services. However, these systems only partially expand the security features of MANO. For instance, Pattaranantakul *et al.*[4] proposes a security extension module based on the TOSCA data model, which is commonly used by the NFV data model that is commonly used by the NFV MANO architecture. To date, the best practices for the NFV security MANO have not been described.

VNF ORCHESTRATION AND DEPLOYMENT

The VNF deployment is an important aspect of the VNF lifecycle management. In early studies

Table I. NFV security project.

Project	Security focus	Detail
OPNFV-Moon[4]	VIM security	Proposes monitoring methods and policy engines to define security policies and manages security enforcement mechanisms to protect different layers of the NFV infrastructure.
EU-SHIELD[5]	Trusted computing / / virtualized network security functions	Offers security as a service (SecaaS) based on virtualized network security functions (vNSFs).The trustworthiness of the vNSFs relies on trusted computing technologies.
FP7-Secured[6]	User-oriented security policies	Offers a common security mechanism to the interfaces between components in its orchestrator TeNOR and integrates a monitoring framework with the core component deployed at the VIM layer.
FP7-T-NOVA[6]	MANO stack / monitoring scalability and metrics aggregation	Offers a common security mechanism to the interfaces between components in its orchestrator TeNOR and integrates a monitoring framework with the core component deployed at the VIM layer.
I ntel-OpenCIT[1]	Integration of its framework with the open source MANO(OSM)	Provides a trust architecture supporting the attestation of both physical platforms and virtual instances in a cloud environment.
OpenStack-Heat[4]	Orchestration security	Provides a template-based orchestration mechanism formalized in YAML that can be extended to support SFC network security policies.

of VNFs, secure deployments generally focused on providing a secure virtual environment and applying the VNFs (e.g., vDPI, virtual firewalls) in NFV. Current trends deliver secure services more efficiently, including making full use of the policies to increase agility and composability for new security services and solving deployment problems more reasonably. There are three main factors for deploying a secure network service.

1) Single VNF composition (VNFC): There are security risks when implementing all types of VNFs through the monolithic VNFs available in early modes. Larger monolithic VNFs mean that the main problem affecting security is reliability. All modules run within the same process; therefore, a bug in any module, such as a memory leak, can potentially bring down the entire process. In recent years, software development teams have focused on decomposing a VNF into smaller functional blocks of reusable microservices[7] with faster response times and relatively no downtime or interruptions to operational processes. Although microservices have obvious advantages, many security challenges still exist when using the microservice-based NFV to deploy a new service. The main drawback is that the complexity of a distributed system causes it to suffer from security vulnerabilities. Microservices usually use a REST mechanism as the main data-interchange format; therefore, attention should be paid to providing secure data transmission. An additional challenge is that such systems require authentication mechanisms by third-party services to ensure that the transmitted data are securely stored. Furthermore, testing microservices in VNFs may be more complex than earlier approaches. To restrict the trust placed on individual microservices and to limit the potential damage from a compromised microservice, the system requires mechanisms that monitor and enforce the connections among microservices. However, the Istio platform[8] deployed with Envoy proxy sidecars provides a solution to the security issue of microservices by enabling traffic encryption at scale and ensuring mutual identity verification, which secures interservice communications across the heterogeneous deployments.
2) Respective order in the chain: Ensuring a correct and efficient order in a secure service chain has also been a topic of recent research. Figure 2 shows the service chain of a firewall, IDS, and proxy for security purposes. The NFs depending on differentiated sharing can be divided into three categories: first, those used only by a single tenant

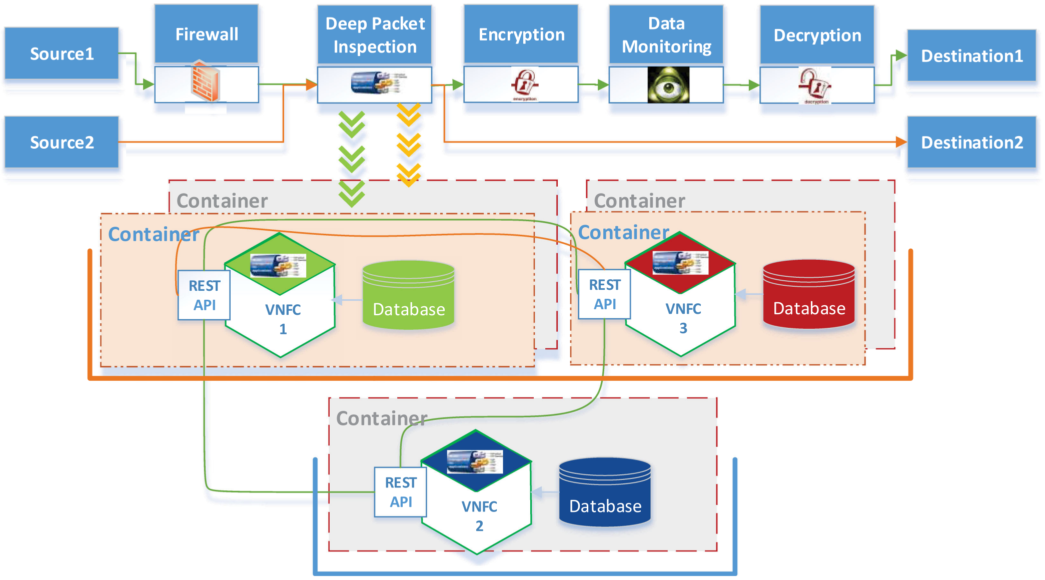

Figure 2. Deployment of a secure service chain.

network, such as a web proxy, second, those in which the same configuration can be shared by different tenant networks, such as a firewall, and third, those shared by the whole cloud, such as DPI. To meet different tenant requirements, NFs may be placed such that a prohibitively large number of NFs is required. One of the existing solutions leverages traffic steering to accomplish service chaining,[9] but extending the traffic paths of service functions may aggravate transmission delay and increase the risk of man-in-the-middle (MitM) attacks. Other solutions use a minimal number of NF instances, merging the same category instances of VNFs as noted above; however, this approach complicates the system due to the need to coordinate multiple security policies and increases the chances of data leakage. Many researchers have proposed programming approaches to address these optimization problems, such as the formulation of mixed integer linear programming problems to minimize energy and traffic-aware costs. However, research concerning microservice placement order of VNFs is lacking, and new types of solutions should be developed due to service coupling and the uneven distribution of responsibilities in microservices, such as services that play more central roles than others. In addition, the cost of security risks should also be used as a restriction during location selection.

3) Deployment location of the VNFC in the NFVI: A reasonable deployment location for the VNFC in the NFVI is a basis for achieving fast, scalable, and dynamic composition network services. Deployment patterns for NFs (e.g., hardware, thread-based, VM-based) affect the service performance and security. Nevertheless, the use of VNF containers usually has considerable advantages compared to the use of VMs and hypervisors, especially in terms of efficiency and performance. However, most NFV deployments to date have run VNFs on a virtual machine, such as on a kernel-based virtual machine or ESXi, and (to a lesser degree) on Hyper-V virtualization layers. Isolation is the largest factor; for example, containers do not provide a mechanism for managing resource quotas, causing the system to be susceptible to "noisy neighbor" risks.

To maximize overall resource utilization and improve service elasticity, resource sharing across NFs should be possible. However, risks exist in sharing among multiple unrelated VNFs due to resource pooling, e.g., attacks on one VNF might affect other VNFs running on the containers in the same VM or on the same physical server. When a VNF is compromised, it should be quarantined while ensuring service continuity for the other VNFs. Therefore, multitenancy requires support for secure slicing of the NFVI resources during the NFV deployment. Provisioning the NFs by guaranteeing complete isolation across resource entities (e.g., hardware units, the hypervisor, virtual networks) includes the implementation of secure access between VMs/containers and host interfaces and secure VM-to-VM or container-to-container communications. Unlike traditional virtual machines that share only the hypervisor and run their own kernels, containerization is a lightweight mechanism that shares the entire kernel. However, the kernel provides a much larger attack surface than a hypervisor does. Fortunately, recent security improvements have focused on minimal host OS distributions that reduce the attack surface and execute host management tools in isolated management containers. The container management solutions (e.g., Kubernetes) also provide self-healing features, such as autoplacement, restart, and replacement using service discovery and continuous monitoring. Today, cloud service providers often use the VM technology to isolate users, and containers, such as Docker are often used to isolate applications and instances. A combined environment consisting of both VMs and containers is typically used by most researchers embracing container movement.

TRUST MANAGEMENT IN NFV

Any identity-based trust or behavior process-based trust should be established in the NFV services from an access control perspective because identity-based trust can maintain the trust state of identities from platform to end users/subscribers in a virtualized environment, and behavior process-based trust can maintain dynamic trust relationships and collaborate to fulfill an update in a trustworthy manner. The NFV remote attestation involves methods to apply identity-based trust or behavior process-based trust by the MANO stack and requires the identification of the trust root(s) to establish a trust chain for the NFVI, individual VNFs, and MANO subsystems and verify the trust service function chain (SFC). Given a sufficient level of assurance for the components in the different software elements constituting VNFs, they can be securely used by the network services. From a security services perspective, several credible mechanisms must be involved to meet the security requirements of the layers in the NFV architecture, including the following three factors.

1) Authorization and authentication: In NFV, new security issues pertaining to authentication and authorization platform certification indicate that the current identity is not unique but rather spread across two or more layers (e.g., the network infrastructure for identifying tenants and the network capabilities for identifying actual users). In end-to-end virtual network architectures, identity stacking can occur at multiple tiers, and any type of horizontal (many NFs can be abstracted from a virtual machine, while the end-to-end network service chain is composed of several virtual organizations) or vertical (i.e., multiple tenants using the same network functionality) integration pattern must address identity-related issues. Recently, attestation research has focused on the attestation of virtual machines through measurement, attestation, and verification technology. Therefore, in addition to applying cryptographic techniques by MANO to verify system integrity, remote attestation mechanisms in the administrative domain should also be enforced to prevent threat injections due to insufficient VM authentication mechanisms. Another problem is that trust based on a third party (e.g., a certificate authority) or an out-of-band channel (e.g., addition of public keys in the known host file) are not suitable for fundamentally solving the problem of trust in an NFV multiparty dynamic environment. The formation of a recognized and standardized negotiation mechanism by the industry is difficult. The challenges to forming such a mechanism include the following: first, trust among multiple parties (e.g., VNFs), especially mutually antidependent parties, needs to be automatically established, second, trust decisions for an identity by any party are not publicly verifiable but rather hidden under one or

more layers, third, establishment of a continuous and reconstructable trust communication mechanism is a significant challenge for the VNFC identity and continuous trust management throughout the VNFM lifecycle when faced with dynamically changing network elements, such as VNF migrations, and fourth, there is a lack of tamper-proof evidence for any trust decision. Thus, the NFV must provide a trust evaluation criterion. For example, establishing a unified consultation mechanism in a dynamic trust environment would be favorable to multiparty actors. The key to the above problems is the definition of unified interactive authentication standard protocol specifications to ensure that widespread multivendor interoperability can be achieved.

2) Building a long chain of trust: The hypervisor and the various management/orchestration elements in an NFV require a long chain of trust. A long chain of trust must be maintained in an NFV environment due to external control and self-service, which creates vulnerabilities. A chain of trust for the NFV components, as shown in Figure 2, is mainly built through the following four major steps. First, the networking infrastructure and the management platform must be secure. Trust of the underlying NFV platforms should begin with the trusted platform module, which is the hardware root of trust. Boot integrity measurement must be used to establish a set of common NFV attestation technologies.[1] A trustworthy boot to ensure validation of the boot integrity of the NFVI components affects many architectural layers during VNF instantiation, including the hardware platform, hypervisor, virtualization container, VNF operating system, and VNF applications. Second, an externally trusted security orchestrator should be built to perform VNF-related security operations. These operations include system attestation, identity-based service chain construction, verification of VNF image integrity before launch, and the provision of orchestration policies for binding VNFs to given NFVI elements. Third, virtual security appliances, such as firewalls, should be deployed, and the islands should be transformed into controlled network zones. Fourth, virtualized functions should be placed in the secure zones established previously. Through these steps, an NFV centralized orchestrator can ensure consistent and horizontal security implementation throughout the policy management mechanism. Recently, to address the above problem, a security and trust framework[10] for the NFVI trust platform and trust functions were proposed to securely deploy various trustworthy security services over virtualized networks. We believe that integration of the long chain of the trusted framework with the NFV MANO management layer to provide a multiparty trust service is a future research direction.

3) Establishment of a trust model and evaluation: Traditional security models are based on an implicit trust model, such as the “trust but verify” approach, which is not completely adapted to the dynamic features of the cloud as a new network edge. We must find a dynamic, automated security policy that extends across conventional security boundaries yet still provides fine-granularity segmentation and isolation of critical resources. Therefore, building an effective security trust model for an NFV is another approach to establishing a trust mechanism for NFV. This concept is an in-depth defense approach that prevents single points of failure from compromising the entire SFC and makes it feasible to disrupt all types of internal attacks as early as possible. In,[11] a zero-trust model (trust nothing, verify everything) is compatible with microservice-based automated network service chaining. This model cannot only authenticate users and applications but also extend down to the level of individual packets. However, further testing and evaluation of the performance overhead and security risks introduced by this model are required.

MONITORING AND EVENT MANAGEMENT IN NFV

Cisco estimates that approximately 73% of data-center traffic will be VM-to-VM by the end of 2019. However, in a virtualized data center, diagnosing network performance or failure to spot malicious agents is extremely difficult.

Either the consolidated vertical “function silos” or the collapsed stack conceal these interfaces. For example, an NFV connection is usually established by a virtual socket instead of by IP packets. Therefore, probing the desired data within the VM or the hypervisor is difficult.[12] To reduce the attack surface and protect user privacy, methods, such as minimizing TC and virtual machine software, are introduced. However, to this end, administrators must strengthen the monitoring of the virtual environment and malicious behaviors.

1) Monitoring in virtual environments: Monitoring virtual environments is the basic premise for network security. However, virtual environment monitoring is a complex problem because it involves heterogeneous hardware and software supplied by different vendors. Any VF adjustment command may change an entire service chain or a chain subset. Therefore, we need active monitoring (e.g., the heartbeat of each component) or passive monitoring (e.g., new connections to endpoints) for such commands. However, monitoring should not degrade the performance of other, unmonitored functions in this environment. In an NFV network, capturing/imaging and processing monitored traffic can seriously affect performance and increase the risk of the MitM attacks. Currently, static or dynamic security policies are configured to meet monitoring requirements. For example, in a carrier network, the NFV orchestrator may monitor an NF performance in relatively real time. In addition, implementing security in a virtualized network includes individually configuring several NFs and deploying services. To be specific, NFs are configured by policies (a set of rules), but their actual behaviors are typically influenced by neighboring NFs[13] Therefore, monitoring challenges include: first, verifying that VNFs performed well and second, addressing problems that arise from adjustments to the monitoring policy by relearning.
2) Anomaly detection techniques for SFCs and performance anomalies: SFCs are vulnerable to many types of attacks, such as unauthorized VNF reconfigurations (for denial of service or to gain unauthorized privileges for specific users), flow redirection, and duplication. Deploying anomaly detection techniques to maintain the integrity of SFCs is necessary for maintaining resilience to well-known zero-day threats. A method of introducing an additional SFC integrity module for the standard NFV architecture was proposed in.[13] This method enables NFV orchestrators to analyze NFV elements and perform suggested actions to maintain network service integrity. Another group of anomalies, known as performance anomalies, are caused by faults in an SFC and can be accidentally or intentionally introduced by intrusion or misuse (e.g., by forcing software crashes or overloading resources to cause denial of service). These performance anomalies also carry potential security risks. Unfortunately, the current anomaly detection techniques are unsuitable for solving such challenges because anomaly detection techniques usually require classification algorithm models to be trained with data obtained from extensive tests or historical datasets. However, because of first, the short timespans involved in SFCs, second, the inaccuracies of splicing previous datasets into new contexts, and third, the poor threshold calibration of specific services, the above conditions may be unattainable. Therefore, identifying causal relationships in the VNF service chaining and building a relationship model between the VNF instances in the network is helpful for anomaly detection in SFCs.
3) Handling security crashes and recovery: If a VNF is compromised (e.g., a misconfiguration or attack), restarting virtual resources is insufficient. Each failure can cause errors in network services. Addressing the aftermath of security failures is particularly important. When the NFV MANO layer senses that a VNF component has failed, that VNF should be quarantined (to avoid a VM escape attack due to the failure of proper isolation between the hypervisors and the VNFs); then, an attempt can be made to heal the VNF. If the compromised VNF cannot securely recover from runtime vulnerabilities or failures and restore the NFs to their operational states, sensitive data (e.g., private keys) from removed virtual machines are still

recoverable from the underlying storage unless a secure erasure method[14] has been enforced. Achieving secure recovery with minimal or no downtime is important and is especially critical for low-latency applications. The container-based virtualization achieves better performance and scalability, whereas traditional hypervisor-based virtualization enforces stronger isolation and more secure and reliable solutions. Cotroneo *et al.* [15] proposed a benchmarking case study for virtualization solutions, namely, VMware ESXi/vSphere (hypervisor-based) and Linux/Docker (container-based). When internal errors occur, ESXi forces host shutdown to trigger a failover on another machine if the hypervisor or VM state becomes unstable. In contrast, Linux/Docker attempts to ignore errors and continue execution, thus hindering the fault recovery process. To achieve higher fault detection coverage, the NFV system designers should pair Docker with additional solutions to detect problems not reported by the OS (e.g., memory overloads) and configure recovery actions for specific symptoms (e.g., internal kernel errors and I/O errors). Most failures, fault injection approaches and their related tools show that the real challenge is to verify successful processing and security and to relearn how to address problems after failures.

CONCLUSIONS AND IDEAS FOR FUTURE DEVELOPMENTS

In this article, the security challenges of management and controls in the NFV environments were examined. Although many research results have been proposed to overcome security challenges in the NFV environments, many potential security risks still exist. Thus, to encourage research on this subject, we identify the following issues and possible directions for future research.

Evaluate the performance impact of enhancing NFV security in a hybrid network: although orchestration systems have bridged existing hardware solutions with virtualized solutions, for performance and functionality, a hybrid network utilizing both proprietary hardware and virtualized services should be developed. An evaluation of the performance impact of incorporating various secure technologies into the overall NFV orchestration service should be conducted. Hardware acceleration techniques exist (e.g., ASICs, FPGAs, NPUs, GPUs) on physical hosts, and software acceleration techniques are available (e.g., data plane development kits, single-root input-output virtualization); flexible use of both types of acceleration techniques is important for addressing the performance bottleneck in solutions to enhance the NFV security.

Forecast and analyze abnormal behaviors through cross-layer monitoring: VNF autoscaling and enhanced platform awareness in the second global ETSI-NFV plug tests shows that the VNF characterization is being watched in the production environment. However, we believe that this system needs additional dynamic security management policies to support the dynamic changes of networks and elements. Forecasting and analyzing abnormal behaviors using cross-layer monitoring data acquired by security tools and formulating appropriate security policies are substantial challenges. Developing special security tools would also be valuable in checking for virtual data-center threats that could be hiding within a virtual data center.

Solve the security issues of Openstack: Most NFV implementations are heavily dependent on OpenStack, which is regarded as the VIM and has become an industry standard. Nonetheless, the security issues of OpenStack cannot be ignored. To achieve these goals, open-source community organizations need to work together to form a vibrant security ecosystem.

An NFV undoubtedly provides great benefits in terms of agility, speed of service delivery, and costs and allows service providers to keep pace with the demands on their network. However, fully understanding the security ramifications of these benefits is critical. This article provides a comprehensive overview based on the ETSI-NFV security architecture. Research on VNF orchestration and deployment, monitoring, and trust management are presented, compared, and evaluated. Finally, promising research areas are revealed, and future directions are presented.

ACKNOWLEDGMENTS

This research was supported by the Natural Science Foundation of Zhejiang Province under Grant Y19F020021.

REFERENCES

1. S. Lal, T. Taleb, and A. Dutta, "NFV: Security threats and best practices," *IEEE Commun. Mag.*, vol. 42, no. 10, pp. 211–217, Aug. 2017.
2. R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-Art and research challenges," *IEEE Commun. Surv. Tut.*, vol. 18, no. 1, pp. 236–262, Jan./Mar. 2016.
3. B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, Feb. 2015.
4. M. Pattaranantakul, Y. Tseng, R He, Z. Zhang, and A. Meddahi, "A first step towards security extension for NFV orchestrators," in *Proc. ACM Int. Workshop Secur. Softw. Defined Netw. Netw. Funct. Virtualization*, 2017, pp. 25–30.
5. H. Attak *et al.*, "SHIELD: Securing against intruders and other threats through an NFV-enabled environment," *Guide to Security in SDN and NFV*, 7. Berlin, Germany: Springer, pp. 197–225, 2017
6. Eur. Commission, Brussels, Belgium *Research and Innovation Funding 2014–2020*. Accessed: Nov. 01, 2016. [Online]. Available: https://ec.europa.eu/research/fp7/index en.cfm
7. M. Fowler and J. Lewis, *Microservices a Definition of This New Architectural Term*, 2014. [Online]. Available: http://martinfowler.com/ articles/microservices.html
8. F. Moyer, "Comprehensive container-based service monitoring with Kubernetes and Istio," in *Proc. Site Rel. Eng. Conf. 18 Asia*, 2018.
9. X. Li and C. Qian, "A survey of network function placement," in *Proc. 13th IEEE Annu. Consum. Commun. Netw. Conf.*, 2016, pp. 948–953.
10. Z. Yan, P. Zhang, and A. V. Vasilakos, "A security and trust framework for virtualized networks and software-defined networking," *Secur. Commun. Netw.*, vol. 9, no. 16, pp. 3059–3069, 2016.
11. C. DeCusatis, P. Liengtiraphan, A. Sager, and M. Pinelli, "Implementing zero trust cloud networks with transport access control and first packet authentication," in *Proc. IEEE Int. Conf. Smart Cloud*, 2016, pp. 5–10.
12. ETSI, Sophia-Antipolis, France, "ETSI GS NFV-SEC 003 network functions virtualisation (NFV) security and trust guidance. Accessed: Dec. 4, 2014. [Online]. Available: http://mosaic-lab.org/uploads/papers/3ce68d82-7a66-4b02-8f59-256ebf45ef49.pdf.
13. L. Bondany, T. Wauters, B. Volckaert, F. De Turck, and L. Z. Granville, "Anomaly detection framework for SFC integrity in NFV environments," in *Proc. IEEE Conf. Netw. Softw.*, 2017, pp. 1–5.
14. J. Reardon, D. Basin, and S. Capkun, "SoK: Secure data deletion," in *Proc. IEEE Symp. Secur. Privacy*, 2013, pp. 301–315.
15. D. Cotroneo, L. De Simone, and R. Natella, "NFV-bench: A dependability benchmark for network function virtualization systems," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 4, pp. 934–948, Dec. 2017.

Xiaochun Wu received the Ph.D. degree in Computer Science and Technology from Zhejiang university, China in 2013. She had one year visiting experience at the Department of Computer Science, Northwestern University, Evanston, IL, USA from 2017 to 2018. She is currently an associate professor with the School of Information and Electronic Engineering, Zhejiang Gongshang University, China. Her research interests include Forwarding and Control Element Separation (ForCES), Software-Defined Networking (SDN), Network Function Virtualization (NFV), and network security. Contact her at spring-403@zjgsu.edu.cn.

Kaiyu Hou received the B.Sc. degree in software engineering and M.Sc. degree in computer science from Xi'an Jiaotong University, Xi'an, China, in 2014 and 2017, respectively. Currently, he is pursuing the Ph.D. degree major in computer science at Northwestern University, Evanston, IL, USA. His research interests include networked system and cellular network protocols. He is a student member of the IEEE. Contact him at kaiyuhou2022@u.northwestern.edu.

Xue Leng received the B.S. degree in computer science and technology from Harbin Engineering University, Harbin, China, in 2015. She is currently pursuing the Ph.D. degree major in computer science and technology with Zhejiang University, Hangzhou, China. Her research interests are softwaredefined networking (SDN), network function virtualization (NFV), and microservice. She is a student member of the IEEE and CCF. Contact her at lengxue_2015@outlook.com.

Xing Li received his B.E. degree in Software Engineering from Shandong University, Jinan, China in 2016. He is currently a Ph.D. candidate in College of Computer Science and Technology, at Zhejiang University, Hangzhou, China. He now is also a visiting Ph. D. student with Computer Science at Northwestern University, Evanston, USA. His research interests include SDN, microservices, and cloud security. Contact him at lx.chn@outlook.com.

Yinbo Yu received the B.E. degree in Electronic Information Engineering from Wuhan University, Wuhan, China, in 2014. He is currently pursuing the Ph.D. degree with the School of Electronic Information, Wuhan University. He was also a visiting Ph.D. student at the Department of Computer Science at Northwestern University, Evanston, IL, USA from 2017 to 2019. His research interests include network security and reliability, cellular network, software formal verification, SDN, NFV and Microservice. Contact him at yyb@whu.edu.cn.

Bo Wu received his B.Eng degree from the school of software of Shandong University, China in 2014, and Ph.D degree from the department of computer science and technology of Tsinghua University, China in 2019. He is working in Network Technology Lab of Huawei Technologies. His research interests include network architecture, NetAI, network security, next generation Internet and Blockchain. Contact him at wub14@mails.tsinghua.edu.cn.

Yan Chen received the Ph.D. degree in computer science from the University of California at Berkeley, Berkeley, CA, USA, in 2003. He is currently a Professor with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA. He is also an IEEE fellow. Based on Google Scholar, his papers have been cited over 11,000 times and his h-index is 49. His research interests include network security, measurement, and diagnosis for large-scale networks and distributed systems. He received the Department of Energy Early CAREER Award in 2005, the Department of Defense Young Investigator Award in 2007, the Best Paper nomination in ACM SIGCOMM 2010, and the Most Influential Paper Award in ASPLOS 2018. Contact him at ychen@northwestern.edu.