Background of Network Function Virtualization

Subjects: Others

Contributor: Abdulrahman K. Alnaim

Network Function Virtualization (NFV) is a virtual network model, the goal of which is a cost-efficient transition of the hardware infrastructure into a flexible and reliable software platform.

Keywords: network function virtualization (NFV) ; cloud computing ; virtualization

1. Introduction

Network service providers need to deploy network equipment such as firewalls, domain name servers (DNSs), load balancers, routers, and switches at the consumers' premises to deliver a network service. These network hardware devices may connect many computers with different operating systems and protocols, which increases the complexity of network infrastructure [1][2]. In addition, TSPs may require deploying additional network equipment to cover the consumers' needs [3], which expands the network infrastructure and increases the operational expenditure (OpEx) and the capital expenditures (CapEx), making managing the network infrastructure a cumbersome process [4]. The heterogeneity of these pieces of equipment also makes it difficult to have a secure network environment.

A different paradigm has emerged in the network industry that developed the network infrastructure and its service delivery. Network Function Virtualization (NFV) takes advantage of virtualization to deliver virtual network functions, i.e., virtual firewalls, virtual switches, etc. It promises independence in hardware and software development, because they are not integrated with each other, and reduces the OpEx, the CapEx, and even the total cost of ownership (TCO) [5][6]. NFV also ensures a sharable and scalable network environment in which many NFV consumers can share and scale the network resources provided by TSPs according to their requirements. Researchers consider here TSPs as NFV providers.

Although NFV promises many benefits, as mentioned previously, it leads to security issues [7][8]. NFV providers are required to undertake substantial efforts to ensure a secure NFV service environment. To provide a secure NFV service, researchers need to study and understand the possible threats. In [9], researchers looked at the main security threats in NFV and the possible countermeasures to these threats. In this research, researchers classified the vulnerabilities and mapped them to their possible threats. Here, researchers use misuse patterns to describe one of these threats, the threat of maliciously modifying non-control data. Misuse patterns are used to describe how an attack is carried out from the point of view of the attacker [10]. They also define the environment in which the attack can be carried out, the possible countermeasures to mitigate it, and the forensic information that could be used to trace the attack once it happens [10]. The patterns are part of an ongoing catalog that can be used by system designers to consider security aspects when building an NFV system.

The threat of modifying non-control data has been studied in various systems. In [11], the researchers demonstrated real-world applications vulnerable to such attacks to show how non-control data attacks are realistic. In [12], the researchers described some possible cases of kernel non-control data attacks. In [13], a data-oriented programming technique was used to construct non-control data attacks on nine applications. The researchers also used a dataflow stitching technique to generate data-oriented exploits that led to non-control data attacks. Another scenario of a non-control data attack is explained in [14], in which a memory corruption vulnerability was leveraged. Although many researchers have explained the possibility of this threat, no one in the literature has used patterns to analyze the threat of non-control data in an NFV system. Patterns have proved convenient to describe the threats in several environments, such as cloud [15], IoT [16], and VoIP [17].

2. Background

2.1. Network Function Virtualization

NFV transforms the traditional network architecture from a static architecture that comprises physical hardware to an agile one that provides network functions as software running in virtual machines (VMs). Decoupling the network functions from its dedicated hardware and emulating them to virtual servers will result in the following benefits [18]:

- Flexibility: The network will be provided as a software service, ensuring flexible and faster deployment;
- Elasticity: NFV consumers will be able to dynamically scale the network resources;
- Extensibility: It would be possible to dynamically add more network services within the network service;
- Faster deployment: The network service will be configured faster.

The European Telecommunication Standards Institute (ETSI) introduced the first architecture of NFV, shown in **Figure 1** [19]. It consists of three main components: the network function virtualization infrastructure (NFVI), virtualized network functions (VNFs), and NFV management and orchestration (MANO).

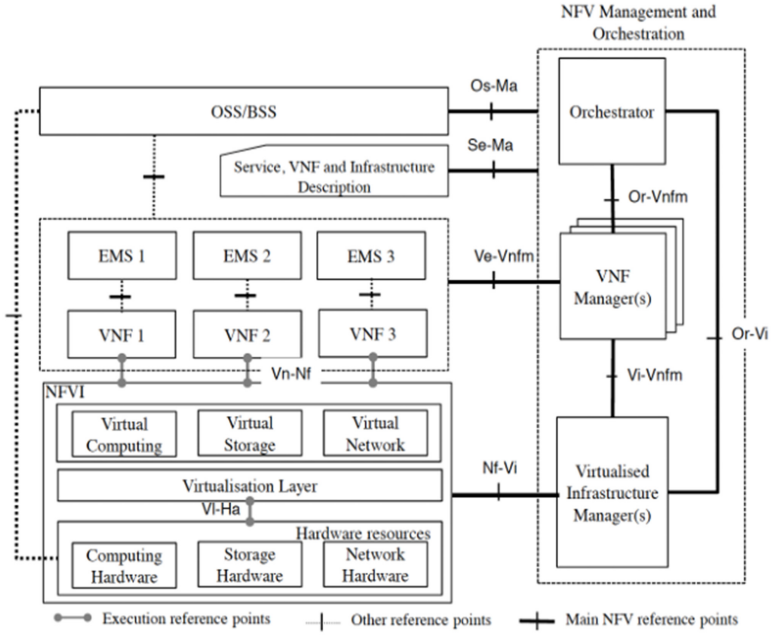

Figure 1. NFV Reference Architecture Framework [19].

The NFV infrastructure is the foundation platform for the network service and contains hardware resources (storage, CPU, network, etc.); virtualized resources (virtual storage, virtual CPU, virtual network, etc.); and the virtualization layer, which contains the hypervisor. The hypervisor, also called the virtual machine manager (VMM), deploys VMs, emulates the necessary resources, and allows for resource sharing, while also ensuring isolation among them [20].

Further, VNFs are software implementations of the network functions that are deployed on the NFVI. A single VNF may contain several components (VNFCs), which are software components of a VNF, or may contain only one network function in order to maintain its scalability. VNFs are hosted by VMs [21] or even a container [22].

The third component is the NFV MANO, which covers the lifecycle management and orchestration of the virtual network service. It contains three management units: the virtualized infrastructure manager (VIM), responsible for managing the interaction between the VNFs and the NFVI resources; the VNF manager, which manages and monitors the VNF resources; and the NFV orchestrator (NFVO), which provisions the necessary resources for the network service.

2.2. Patterns

A pattern is a solution to a recurrent problem in a given context. Patterns embody abstractions and provide common vocabularies for system designers. Their solutions are suggestions, not plug-ins [18], which means that they are prototypes and there are many ways to instantiate a pattern. There are several types of patterns, intended for specific design purposes. Design and architectural patterns are used to build the functional aspects of extendable systems [23]. Security patterns are used to build secure systems by defining a way of controlling vulnerabilities or stopping specific attacks [10].

Threat patterns describe the steps of an attack that could lead to several misuses [24]. Misuse patterns are used to describe, from the attacker's perspective, a generic method of attacking a system by exploiting a vulnerability. They also describe the environment in which an attack may be performed, the possible countermeasures to mitigate it, and the method to find forensic information to trace the attack once it happens [10].

Patterns are described using templates; each template is different based on the type of pattern. For example, a misuse pattern template contains countermeasures, consequences, and forensic sections that are not available in a design pattern template. Researchers use the Pattern-Oriented Software Architecture (POSA) template as researchers consider it more suitable for describing security aspects [10]. The descriptions of patterns may be written in textual language, and their solutions are usually shown in Unified Modeling Language (UML).

2.3. Modifying Non-Control Threat

Most security threats are related to altering the control flow of the targeted system, either by injecting a code or by reusing existing code such as return and call instructions [25][26]. It mainly refers to the data loaded in the processing counter during program execution, in which the attacker exploits, for instance, a memory corruption vulnerability, such as buffer overflow or integer overflow, to compromise the system [27][28]. It has been indicated that threats related to non-control data are also possible in real-world applications and are closely equivalent to control data threats [11][29]. Non-control data attacks do not affect the control flow of a system; instead, they are carried out by altering the non-control data of a targeted program, such as configuration data, decision-making data, user identity data, and user input [28]. There are also other critical non-control data in the kernel level susceptible to an attack, such as user privilege data, resource utilization data, and service policy data [12][30].

The threat of modifying non-control data is also possible in the virtualization environment. The hypervisor is being used as a virtualization layer in many systems, including Network Function Virtualization (NFV) [31], due to its ability to enable resource sharing and, at the same time, ensure isolation among virtual machines. Since hypervisors have a smaller code base, it has been assumed that they and the VMs running on top of them are secure [32][33][34][35]. However, some hypervisors indeed are quite complex and have large lines of codes. For instance, Xen contains more than 900K lines of codes [36]. Kernel-based Virtual Machine (KVM) contains around 850K lines of codes [37]. Continuing bug and exploit reports indicate that hypervisors are not secured, as has been assumed [38][39][40][41], and neither are non-control data, which are considered an exploitable attack vector [12][13][30].

References

1. Sinh, D.C.; Le, L.V.; Lin, B.S.P.; Tung, L.P. SDN/NFV—A New Approach of Deploying Network Infrastructure for IoT. In Proceedings of the 27th Wireless and Optical Communication Conference, WOCC, Hualien, Taiwan, 30 April–1 May 2018; pp. 1–5.
2. Masutani, H.; Nakajima, Y.; Kinoshita, T.; Hibi, T.; Takahashi, H.; Obana, K.; Shimano, K.; Fukui, M. Requirements and Design of Flexible NFV Network Infrastructure Node Leveraging SDN/OpenFlow. In Proceedings of the 2014 International Conference on Optical Network Design and Modeling, Stockholm, Sweden, 19–22 May 2014; pp. 258–263.
3. Manzalini, A.; Italia, T.; Roberto Saracco, I.; Labs, E.; Cagatay Buyukkoc, I.; Gladisch, A.; Fukui, M.; Shen, W.; Eliezer Dekel, J.; David Soldani, I.; et al. Software-Defined Networks for Future Networks and Services Main Technical Challenges and Business Implications. White Paper Based on the IEEE Workshop SDN4FNS. 2014. Available online: https://discovery.ucl.ac.uk/id/eprint/10043677/1/White%20Paper%20IEEE%20SDN4FNS-FinalVersion.pdf (accessed on 18 June 2022).
4. Yoshida, M.; Shen, W.; Kawabata, T.; Minato, K.; Imajuku, W. MORSA: A Multi-Objective Resource Scheduling Algorithm for NFV Infrastructure. In Proceedings of the 16th Asia-Pacific Network Operations and Management Symposium, Hsinchu, Taiwan, 17–19 September 2014; pp. 1–6.
5. Bouras, C.; Ntarzanos, P.; Papazois, A. Cost Modeling for SDN/NFV Based Mobile 5G Networks. In Proceedings of the International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, Lisbon, Portugal, 18–20 October 2016; pp. 56–61.
6. Yoon, M.S.; Kamal, A.E. NFV Resource Allocation Using Mixed Queuing Network Model. In Proceedings of the 2016 IEEE Global Communications Conference, GLOBECOM, Washington, DC, USA, 4–8 December 2016; pp. 1–6.
7. Lal, S.; Taleb, T.; Dutta, A. NFV: Security Threats and Best Practices. IEEE Commun. Mag. 2017, 55, 211–217.

8. Yang, W.; Fung, C. A Survey on Security in Network Functions Virtualization. In Proceedings of the IEEE NetSoft Conference and Workshops: Software-Defined Infrastructure for Networks, Clouds, IoT and Services, Seoul, Korea, 6–10 June 2016; pp. 15–19.
9. Alwakeel, A.M.; Alnaim, A.K.; Fernandez, E.B. A Survey of Network Function Virtualization Security. In Proceedings of the IEEE Southeastcon, St. Petersburg, FL, USA, 19–22 April 2018; pp. 1–8.
10. Fernandez, E.B. Security Patterns in Practice: Designing Secure Architectures Using Software Patterns; J. Wiley & Sons: Hoboken, NJ, USA, 2013; ISBN 9781119998945.
11. Chen, S.; Xu, J.; Sezer, E.C.; Gauriar, P.; Iyer, R.K. Non-Control-Data Attacks Are Realistic Threats. In Proceedings of the 14th Conference on USENIX Security Symposium, Baltimore, MD, USA, 31 July–5 August 2005.
12. Baliga, A.; Kamat, P.; Iftode, L. Lurking in the Shadows: Identifying Systemic Threats to Kernel Data (Short Paper). In Proceedings of the IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 20–23 May 2007; pp. 246–251.
13. Hu, H.; Shinde, S.; Adrian, S.; Chua, Z.L.; Saxena, P.; Liang, Z. Data-Oriented Programming: On the Expressiveness of Non-Control Data Attacks. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 969–986.
14. Carlini, N.; Barresi, A.; Payer, M.; Wagner, D.A.; Gross, T. Control-Flow Bending: On the Effectiveness of Control-Flow Integrity. In Proceedings of the USENIX Security Symposium, Washington, DC, USA, 12–14 August 2015; pp. 161–176.
15. Hashizume, K.; Yoshioka, N.; Fernandez, E.B. Misuse Patterns for Cloud Computing. In Proceedings of the 2nd Asian Conference on Pattern Languages of Programs—AsianPLoP ’11, Tokyo, Japan, 5–8 October 2011; pp. 1–6.
16. Syed, M.H.; Fernandez, E.B.; Moreno, J. A Misuse Pattern for DDoS in the IoT. In Proceedings of the 23rd European Conference on Pattern Languages of Programs, Irsee, Germany, 4–8 July 2018; ACM: New York, NY, USA, 2018; pp. 1–5.
17. Pelaez, J.C.; Fernandez, E.B.; Larrondo-Petrie, M.M.; Wieser, C. Misuse Patterns in VoIP. In Proceedings of the 14th Conference on Pattern Languages of Programs—PLOP ’07, Monticello, IL, USA, 5–8 September 2007; ACM: New York, NY, USA; pp. 1–13.
18. Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerland, P.; Stal, M. Pattern-Oriented Software Architecture Volume 1: A System of Patterns; Wiley: Hoboken, NJ, USA, 1996; Volume 1, ISBN 978-0-471-95869-7.
19. ETSI. Network Functions Virtualisation (NFV); Architectural Framework. 2014. Available online: https://cdn.standards.iteh.ai/samples/43827/5288dd7aff4b4de6a4a63a5034c00168/ETSI-GS-NFV-002-V1-2-1-2014-12-.pdf (accessed on 18 June 2022).
20. Chandramouli, R. Security Recommendations for Hypervisor Deployment on Servers-NIST Special Publication 800-125A; 2018. Available online: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-125A.pdf (accessed on 18 June 2022).
21. Alnaim, A.K.; Alwakeel, A.M.; Fernandez, E.B. A Pattern for an NFV Virtual Machine Environment. In Proceedings of the 13th Annual IEEE International Systems Conference, Orlando, FL, USA, 8–11 April 2019; pp. 1–6.
22. Syed, M.H.; Fernandez, E.B. A Reference Architecture for the Container Ecosystem. In Proceedings of the ACM International Conference Proceeding Series, Hamburg, Germany, 27–30 August 2018; pp. 1–6.
23. Fernandez, E.B.; Yoshioka, N.; Washizaki, H.; Syed, M.H. Modeling and Security in Cloud Ecosystems. Future Internet 2016, 8, 13.
24. Sulatycki, R.; Fernandez, E.B. A Threat Pattern for the “Cross-Site Scripting (XSS)” Attack. In Proceedings of the 22nd Conference on Pattern Languages of Programs, Pittsburgh, PA, USA, 24–26 October 2015.
25. Cybersecurity and Infrastructure Security Agency (CISA). CERT Security Advisories CISA. Available online: https://www.cisa.gov/uscert/ics/advisories (accessed on 16 January 2022).
26. Microsoft. Microsoft Security Bulletins. Available online: https://docs.microsoft.com/en-us/security-updates/securitybulletins/securitybulletins (accessed on 18 June 2022).
27. Abadi, M.; Budiu, M.; Erlingsson, Ú.; Ligatti, J. Control-Flow Integrity Principles, Implementations, and Applications. ACM Trans. Inf. Syst. Secur. TISSEC 2009, 13, 1–40.
28. Schlesinger, C.; Pattabiraman, K.; Swamy, N.; Walker, D.; Zorn, B. Modular Protections against Non-Control Data Attacks. In Proceedings of the IEEE Computer Security Foundations Symposium, Cernay-la-Ville, France, 27–29 June 2011; pp. 131–145.
29. Sotirov, A. Modern Exploitation and Memory Protection Bypasses. 2009. Available online: https://www.usenix.org/conference/usenixsecurity09/technical-sessions/presentation/sotirov (accessed on 18 June

2022).

30. Ding, B.; He, Y.; Wu, Y.; Yu, J. Systemic Threats to Hypervisor Non-Control Data. IET Inf. Secur. 2013, 7, 349–354.
31. ETSI. Network Functions Virtualisation (NFV); Infrastructure; Hypervisor Domain. 2015. Available online: https://www.etsi.org/deliver/etsi_gs/nfv-inf/001_099/004/01.01.01_60/gs_nfv-inf004v010101p.pdf (accessed on 18 June 2022).
32. Garfinkel, T.; Rosenblum, M. A Virtual Machine Introspection Based Architecture for Intrusion Detection. In Proceedings of the Annual Network and Distributed Systems Security Symp, San Diego, CA, USA, 6 February 2003; pp. 191–206.
33. Jiang, X.; Wang, X.; Xu, D. Stealthy Malware Detection through VMM-Based “out-of-the-Box” Semantic View Reconstruction. In Proceedings of the ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 31 October–2 November 2007; pp. 128–138.
34. Payne, B.D.; Carbone, M.; Sharif, M.; Lee, W. Lares: An Architecture for Secure Active Monitoring Using Virtualization. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 18–22 May 2008; pp. 233–247.
35. Litty, L.; Andrés Lagar-Cavilla, H.; Lie, D. Hypervisor Support for Identifying Covertly Executing Binaries. In Proceedings of the USENIX Security Symp, San Jose, CA, USA, 28 July–1 August 2008; p. 258.
36. SynopSys Black Duck Open Hub-Xen Project (Hypervisor). Available online: https://www.openhub.net/p/xenproject-hypervisor/analyses/latest/languages_summary (accessed on 18 June 2022).
37. SynopSys Black Duck Open Hub-KVM. Available online: https://www.openhub.net/p/kvm/analyses/latest/languages_summary (accessed on 18 June 2022).
38. Perez-Botero, D.; Szefer, J.; Lee, R.B. Characterizing Hypervisor Vulnerabilities in Cloud Computing Servers. In Proceedings of the International Workshop on Security in Cloud Computing—Cloud Computing ’13, Hangzhou, China, 8 May 2013; pp. 3–10.
39. NIST. National Vulnerability Database—CVE-2011-1898. Available online: https://nvd.nist.gov/vuln/detail/CVE-2011-1898 (accessed on 18 June 2022).
40. NIST. National Vulnerability Database—CVE-2021-36148. Available online: https://nvd.nist.gov/vuln/detail/CVE-2021-36148 (accessed on 18 June 2022).
41. NIST. National Vulnerability Database—CVE-2021-38923. Available online: https://nvd.nist.gov/vuln/detail/CVE-2021-38923 (accessed on 18 June 2022).