Security of Software-Defined Infrastructures with SDN, NFV, and Cloud Computing Technologies

1

Doan B. Hoang and Sarah Farahmandian

1.1 Introduction

Software-defined networking separates the control plane from the underlying network data plane for both efficient data transport and fine-grained control of network management and services. SDN allows network virtualization and provision of virtual networks on demand. Network functions virtualization is a network architecture concept in which network functions are virtualized, implemented in software, and deployed strategically with the support of a dynamic virtual/physical infrastructure/platform to provide network services.

Cloud computing relies on its aggregation and centralization of virtual resources and their flexible provision and orchestration to provide services to its customers.

Software-defined networks, network functions virtualization platforms, and clouds have established themselves as modern IT service infrastructures. They all rely on the virtualization technology to virtualize and aggregate physical resources into pools of virtual resources (virtual machines, virtual networks, virtual storage, virtual functions, and virtual services) and provision them to users on demand. Security has been recognized as an essential and integral part in the design of systems, infrastructures, organizations, and services; yet, the current state of security research and practice is at best fragmented, local, or case specific. With modern infrastructures that support ever-increasing complex and pervasive applications, such as social networks, Internet of everything, mobile applications, cloud services, new security models, and innovative security, technologies must be invented to match the complexity of emerging applications and the sophistication of their attackers.

D.B. Hoang (✉) • S. Farahmandian
University of Technology Sydney, Ultimo, NSW, Australia
e-mail: Doan.Hoang@uts.edu.au; sarah.farahmandian@student.uts.edu.au

S.Y. Zhu et al. (eds.), *Guide to Security in SDN and NFV*, Computer
Communications and Networks, DOI 10.1007/978-3-319-64653-4_1

This chapter discusses the security of those software-defined infrastructures using their paradigms and their underlying technologies: virtualization of network infrastructures, virtualization of virtual machines, network functions, and security functions and services. In particular, it explores security architectures, virtual security elements, and virtual connectivity infrastructures for supporting security goals and services. The chapter is organized as follows. Section 1.2 summarizes the defining characteristics and the common virtualization technology of SDN, NFV, and cloud computing. Section 1.3 provides a summary of major security challenges specific to SDN, NFV, and cloud. Section 1.4 discusses key security challenges and solutions to SDN, NFV, and cloud including virtualization, isolation, and security of identity and access management. Section 1.5 discusses the security of OpenStack, a widely deployed platform for implementing cloud-SDN-NFV infrastructure. Section 1.6 reviews and discusses the development of the new software-defined security approach. Section 1.7 concludes the chapter with some remaining challenges.

1.2 Defining Characteristics of Software-Defined Networking, Network Functions Virtualization, and Cloud Computing

This section provides a brief description of SDN, NFV, and cloud computing and their defining characteristics. Virtualization is described as the common underlying technology, and its security is one of the key security challenges in SDI.

1.2.1 Software-Defined Networking

Software-defined networking has emerged as a networking paradigm that separates the data forwarding plane from the control plane by centralizing the network state and the decision-making capability in the control plane (SDN controller), leaving simple forwarding operation at the data plane (SDN network devices), and abstracting the underlying network infrastructure to the application plane. The separation of the control plane and the data forwarding plane is through a programming interface between the SDN network devices and the SDN controller.

The Open Networking Foundation (ONF) defines a high-level architecture for SDN [3], with three main layers as shown in Fig. 1.1: the application layer for expressing and orchestrating application and network service requirements; the control layer for network control, services provisioning, and management; and the infrastructure layer for abstraction of physical network resources. The infrastructure layer can be expanded into two planes: the physical plane and the virtual plane. The physical resources plane consists of the underlying physical infrastructure, and the virtual resources plane represents the virtual resources abstracted from the physical resources through virtualization.

SDN network devices are all placed at the infrastructure layer. The SDN network devices make a simple decision of what to do with incoming traffic (frames or packets) according to instructions programmed by their SDN controller. The SDN

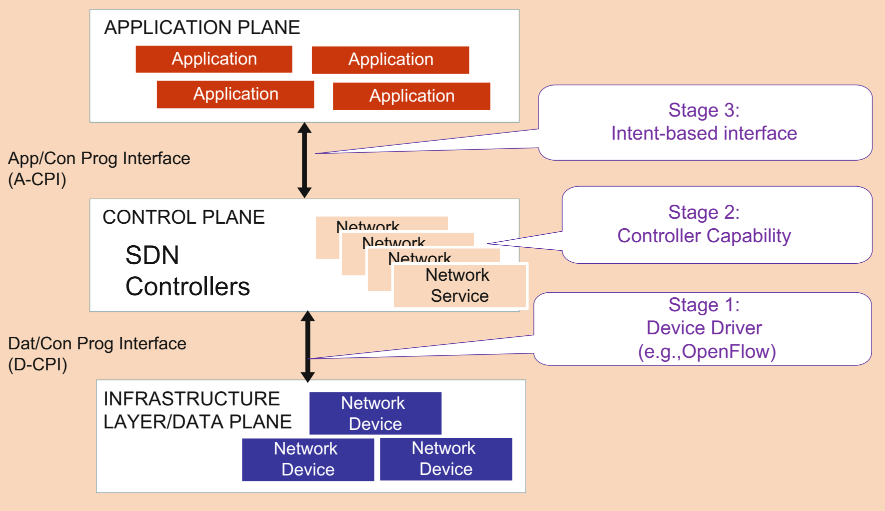

Fig. 1.1 Software-defined network architecture

controller (or group of controllers) is located in the control layer. It programs and controls the forwarding behavior of the network devices and presents an abstraction of the underlying network infrastructure to the SDN applications. Applications and network services are on the application layer. The controller allows applications to define traffic flows and paths, with the support of a comprehensive information database of all underlying network infrastructure operations, in terms of common characteristics of packets to satisfy the applications' needs and to respond to dynamic requirements by users and traffic/network conditions [11].

The SDN controller uses interfaces for communicating with other layers. To communicate with the data/infrastructure layer, a southbound interface (SBI) is used for programming and configuring network devices. To communicate with the application layer, a northbound interface (NBI) is provided for the interaction between the SDN controller and applications. The NBI is to describe the needs of the application and to pass along the commands to orchestrate the network. East/west interfaces are for information exchange between multiple or federated controllers. The OpenFlow protocol has been developed and widely adopted as one of the SBIs between SDN controllers and SDN switches. OpenFlow uses a secure channel for message transmission over the Transport Layer Security (TLS) connection.

1.2.2 Network Functions Virtualization

Network functions virtualization (NFV) is proposed aiming to virtualize an entire class of network component functions using virtualization technologies. The objective is to decouple the network functions from the network equipment. A network function is now a virtual instance of customized software program called a virtual network function (VNF). This object can be created on demand, launched into operation wherever needed without the need for installation of new equipment (on any virtual or physical servers at data centers, gateways, routers). It can be moved at will and terminated when its function is no longer needed [2]. The NFV enables network functions to be executed as software instances in a virtual machine (VM) on a single or multiple hosts instead of customized hardware equipment. Network functions virtualization can be applied to both data and control planes in fixed or mobile infrastructures. The NFV provides operators the ability to combine numerous different types of network equipment into high-volume switches, servers, and storage inside data centers, network nodes, and end user premises. It offers a new means for creating, deploying, and managing networking services.

Examples of these classless of functions include switching elements; tunnel gateway elements: IPSec/SSL (secure sockets layer), VPN (virtual private network) gateways; security functions: firewalls, virus scanner, and intrusion detection systems; traffic analysis services: load balancers, network monitoring, and deep packet inspection tools; service assurance: SLA (service-level agreement) monitoring, test, and diagnostics; mobile network elements: multifunction home router, set top boxes, base stations, and the evolved packet core (EPC) network [13].

ETSI provides an NFV reference architecture for a virtualized infrastructure and points of reference to interconnect the different components of the architecture. The NFV architecture has three key components for building a practical network service: network functions virtualization infrastructure (NFVI), VNFs, and NFV management and orchestration (MANO) [8]. Figure 1.2 shows an overall view of NFV architecture adapted from ETSI NFV model.

The NFVI includes hardware and a hypervisor that virtualizes and abstracts the underlying resources. The VNF is the software implementation of a network function which runs over the NFVI. The NFV MANO is responsible for configuring, deploying, and managing the life cycle of VNFs. An important key principle of NFV is service chaining: as each VNF provides limited functionality on its own, service chaining allows combining multiple VNFs to create useful new network functions and services.

1.2.3 Cloud Computing

Cloud computing has become an alternative IT infrastructure where users, infrastructure providers, and service providers all share and deploy resources for their business processes and applications. Business customers are shifting their services and applications to cloud computing since they do not need to invest in their own costly IT infrastructure but can delegate and deploy their services effectively to cloud vendors and service providers [37].

Cloud computing offers an effective solution for provisioning services at lower costs, on demand over the Internet by virtue of its capability of pooling and virtualizing computing resources dynamically. Clients can leverage a cloud to store

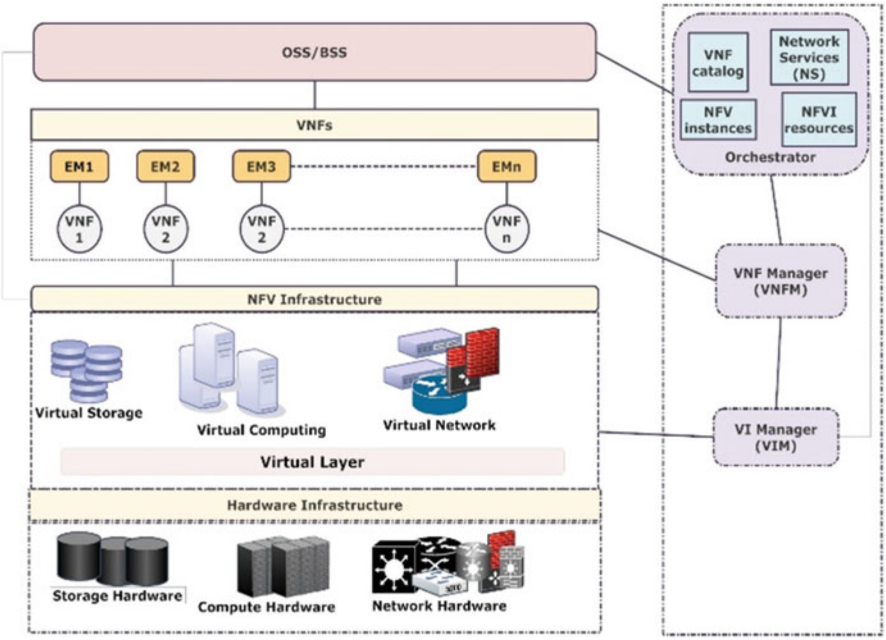

Fig. 1.2 NFV architecture

their documents online, share their information, and consume or operate their services with simple usage, fast access, and low cost on a remote server rather than physically local resources [26].

The most relevant definition is probably the one provided by the National Institute of Standards and Technology (NIST) [17]: “Cloud computing is a model for enabling ubiquitous, convenient, on-demand, network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” This cloud model is composed of five essential characteristics, three service models, and four deployment models. The five characteristics are on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. Software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) constitute the three service models. SaaS directly offers cloud services such as Google Docs, Google Map, Google Health, etc., online to users. With PaaS, developers can order a required development platform, which may consist of SDK (software development kit), documentation, and test environment, to develop their own applications. IaaS is more about packaging and provisioning underlying virtual resources to customers, who then build, orchestrate, provision, and sell tailored infrastructure resources to organizations to support their own businesses.

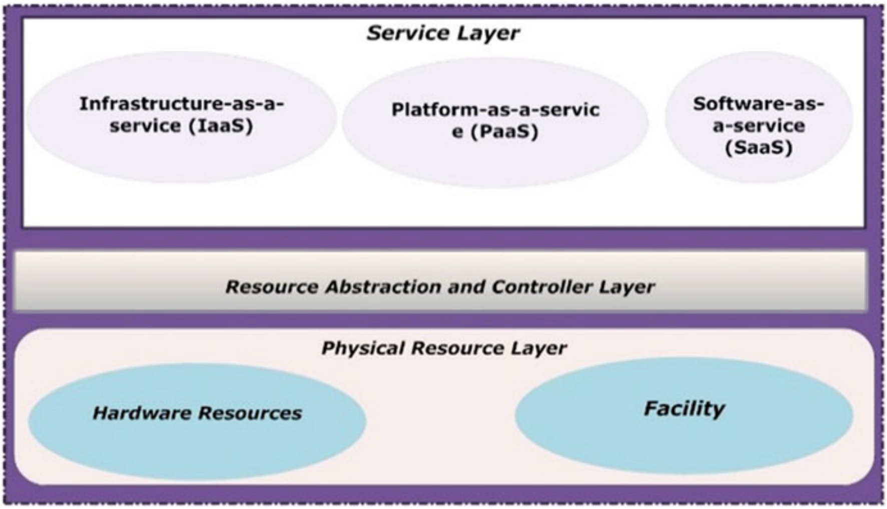
Service Layer
Infrastructure-as-a-service (IaaS)
Platform-as-a-servic e (PaaS)
Software-as-a-service (SaaS)
Resource Abstraction and Controller Layer
Physical Resource Layer
Hardware Resources
Facility

Fig. 1.3 Cloud provider—three-layer service orchestration model

NIST provides a three-layer service orchestration model as shown in Fig. 1.3. The *physical resource layer* includes all the physical computing resources: computers (CPU and memory), networks (routers, firewalls, switches, network links, and interfaces), storage components (hard disks), and other physical computing infrastructure elements. The *resource abstraction and control layer* contains the system components that cloud providers use to provide and manage access to the physical computing resources through software abstraction (virtualization layer). The resource abstraction components include software elements such as hypervisors, virtual machines, virtual data storage, and other computing resource abstractions. The control aspect of this layer refers to the software components that are responsible for resource allocation, access control, and usage monitoring. The *service layer* contains interfaces for cloud consumers to access the computing services.

1.2.4 Virtualization

Virtualization is a key technology for cloud computing, SDN, and NFV. The technology enables network functions virtualization and software-defined network the ability to create a scalable, dynamic, and automated programmable virtual network functions and virtual network infrastructures in integrated cloud platforms such as telecom clouds. Virtualization is the technology that simulates the interface to a physical object by *multiplexing*, *aggregation*, or *emulation*. It is a process that translates hardware into emulated software-based copies. The virtualization simulates the interface to a physical object by several means: with multiplexing, it creates multiple virtual objects from an instance of a physical object; with aggregation, it creates one virtual object from multiple physical objects; and with emulation, it constructs a virtual object from a different type of physical object [16].

On another level, virtualization can be defined as the logical abstraction of assets, such as the hardware platform, operating system (OS), storage devices, network, services, or programming interfaces. More commonly, virtualization is introduced as a software abstraction layer placed between an operating system and the underlying hardware (computing, network, and storage) in the form of a hypervisor. A hypervisor is a small and specialized operating system that runs on a physical server (host machine), allowing physical resources to be partitioned and provisioned as virtual resources (virtual CPU, virtual memory, virtual storage, and virtual networks). On computing resources, a hypervisor creates and manages virtual machines which are isolated instances of the application software and guest OS that run like separate computers. A virtual machine (VM) encapsulates the virtual hardware, the virtual disks, and the metadata associated with the application. In cloud data centers, since the hypervisor manages the hardware resources, multiple virtual machines each with its own operating system and applications and network services can run in parallel in a single hardware device [25]. Figure 1.4 illustrates the virtualization of virtual machines.

Virtualization allows elastic and scalable resource provisioning and sharing among multiple users. The technology allows multi-tenancy in clouds through isolation mechanism and enables each cloud tenant to perform its own services, applications, operating systems, and even network configuration in a logical environment without concerns over the same underlying physical infrastructure. Virtualization results in better server utilization and server/data center consolidation (multiple VMs run within a physical server) and workload isolation (each application on a physical server has its own separate VM).

Virtualization technology has been deployed by enterprises in data centers storage virtualization (NAS (network-attached storage), SAN (storage area network)),

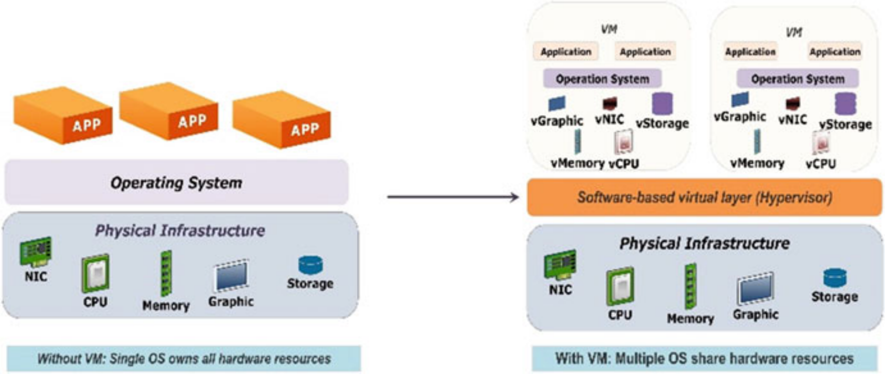

Fig. 1.4 Virtual machines virtualization

database), OS virtualization (VMware, Xen), software or application virtualization (Apache Tomcat, JBoss, Oracle App Server, Web Sphere), and network virtualization [35].

1.3 Security Challenges of NFV, SDN, and Cloud

This section summarizes concepts that are pertinent to our discussion on security issues of SDN, NFV, and cloud. It summarizes their current security challenges.

1.3.1 General Security Requirements and Definitions

For securing an entity/system, it is widely accepted that five essential security functions are required: confidentiality, integrity, availability, authenticity, and accountability (CIAAA). Confidentiality ensures that private and confidential information about data or individuals is not disclosed to unauthorized users. Integrity ensures that information and intended system operation are not tampered with inadvertently or deliberately by unauthorized users. Availability ensures that systems and services are not denied to unauthorized users. Authenticity ensures that users can be verified and trusted as who they claim they are and that inputs arriving at the system came from a trusted source. Accountability generates the requirement for actions of an entity to be traced uniquely to that entity [30].

A system, an organization, or a cyberspace consists of three key elements: *real and virtual entities*, an *interconnecting infrastructure*, and *interactions among entities through the infrastructure*. Real and virtual entities include real things of physical devices such as human beings, computers, sensors, mobile phones, electronic devices, and virtual abstraction of entities such as data/information, software, and services. Infrastructure includes networks, databases, information systems, and storage that interconnect and support entities in the system/space. Interaction encompasses activities and interdependencies among system/cyberspace entities via the interconnecting infrastructure and the information within concerning communication, policy, business, and management [15]. Information or cybersecurity can be considered systems, tools, processes, practices, concepts, and strategies to prevent and protect the cyberspace from unauthorized interaction by agents with elements of the space to maintain and preserve the confidentiality, integrity, availability, and other properties of the space and its protected resources [15].

Essentially, cybersecurity is concerned with identifying vulnerabilities of cyberspace, assessing the risk associated with threats that exploit the vulnerability, and providing security solutions. A security vulnerability is a weakness in a system (component/product/system/cyberspace) that could allow an attacker to compromise the confidentiality, integrity, availability, authenticity, or accountability of that system. Threats and risks are closely related, but they are not equivalent. A threat is any entity, action, or condition that results in harm, loss, damage, and/or a deterioration of existing conditions. The risk associated with a threat is a characteristic that embraces three components: *the impact or importance of a threat incident*, *the likelihood or potential of a future threat incident*, and *the potential loss due to a threat incident*. Evaluating the risk associated with a threat provides the impetus for going forward with security solutions and the requirements for those solutions [36].

1.3.2 NFV Security Challenges

Because network components are virtualized, NFV networks contain a level of abstraction that does not appear in traditional networks. Securing this complex and dynamic environment, that encompasses the virtual/physical resources, the controls/protocols, and the boundaries between the virtual and physical networks, is challenging for many reasons according to CSA [18]:

- *Hypervisor dependencies* Hypervisors are available from many vendors. They must address security vulnerabilities in their software. Understanding the underlying architecture, deploying appropriate types of encryption, and applying patching diligently are all critical for the security of the hypervisors.
- *Elastic network boundaries* In NFV, the network fabric accommodates multiple functions. Physical and virtual boundaries are blurred or nonexistent in NFV architecture, which makes it difficult the design of security systems.
- *Dynamic workloads* While NFV is about agility and dynamic capabilities, traditional security models are static and unable to evolve as network topology changes in response to demand.
- *Service insertion* NFV promises elastic, transparent networks since the fabric intelligently routes packets that meet configurable criteria. Traditional security controls are deployed logically and physically in-line. With NFV, there is often no simple insertion point for security services that are not already layered into the hypervisor.
- *Stateful versus stateless inspection* Security operations during the last decade have been based on the premise that stateful inspection is more advanced and superior to stateless access controls. NFV may add complexity where security controls cannot deal with the asymmetry flows created by multiple, redundant network paths and devices.
- *Scalability of available resources* Deeper inspection technologies—next-generation firewalls and Transport Layer Security decryption, for example—are resource intensive and do not always scale without offload capability.

The ETSI Security Expert Group focuses on the security of the software architecture. It identified potential security vulnerabilities of NFV and established whether they are new problems or just existing problems in different guises [32]. The identified new security concerns resulting from NFV are as shown in Table 1.1.

Table 1.1 Summary of potential areas of concern [32]

Topology validation and enforcement
Availability of management support infrastructure
Secure boot
Secure crash
Performance isolation
User/tenant authentication, authorization, and accounting
Authentication time services
Private keys within cloned images
Backdoors via virtualized test and monitoring functions
Multi-administrator isolation

1.3.3 SDN Security Challenges

SDN introduces a new networking paradigm, and its impact is in the form of a new framework, new components, structural layers, and interfaces. SDN brings with it new security challenges beyond those existed in traditional networks. As SDN decouples the control plane from the data plane, the technology brings with it new sets of components, interfaces, as well as many new security issues. Security challenges in SDN can be divided based on its three layers: *the data plane*, *the control plane*, and *the application plane*. The data plane can suffer from various security threats such as malicious OpenFlow switches, flow rule discovery, flooding attacks (e.g., switch flow table flooding), forged or faked traffic flows, credential management, and insider malicious host. The application plane inherits security challenges such as unauthorized or unauthenticated applications, fraudulent role insertion, lack of authentication methods, and lack of secure provisioning. The control plane faces several security issues related to centralized SDN controller, communication interfaces, policy enforcement, flow rule modification for modifying packets, controller-switch communication flood, system level SDN security challenges (related to lack of auditing accountability mechanisms), and lack of trust between the SDN controller and third-party applications [9]. Since the control plane in the SDN architecture acts as the heart of this virtual network infrastructure, security vulnerabilities on this layer can cause failure to the entire virtual network architecture.

Scott-Hayward et al. presented a comprehensive analysis of the security challenges of SDN [27]. Security challenges associated with the SDN framework by affected layer/interface are categorized as follows:

- *Application Layer* Unauthorized access is through the unauthenticated application. Malicious applications may introduce fraudulent rule insertion. Configuration issues arise from lack of policy enforcement.
- *Control Layer* Unauthorized access can be introduced through unauthorized controller access and unauthenticated application. Data modification is introduced in the form of flow rule modification to modify packets. Malicious applications can

introduce fraudulent rule insertion and controller hijacking. Denial of service (DoS) may occur due to controller-switch communication flood. Configuration issues may arise because of the lack of TLS (or other authentication techniques) adoption or lack of policy enforcement.

- *Data Layer* Unauthorized access may occur with unauthorized controller access. Data leakage may result from flow rule discovery (side-channel attack on input buffer) or forwarding policy discovery (packet processing timing analysis). Data modification is a result of flow rule modifications. Malicious applications may introduce controller hijacking. Denial of service may occur due to controller-switch communication flood or switch flow table flooding. Configuration issues may arise from lack of TLS (or other authentication techniques) adoption.
- *Application-Control Interface* (*NBI—Northbound Interface*) Unauthorized access may occur because of unauthenticated applications. The malicious application may introduce fraudulent rule insertion. Configuration issues may occur due to lack of policy enforcement.
- *Control-Data Interface* (*SBI—Southbound Interface*) Unauthorized access can be introduced through unauthorized controller access. Data modification is introduced in the form of flow rule modifications. Malicious applications can introduce controller hijacking. Denial of service may occur due to controller-switch communication flood. Configuration issues may arise from lack of TLS (or other authentication techniques) adoption.

1.3.4 Cloud Security Challenges

While there are many security concerns in cloud computing, Cloud Security Alliance (CSA) released twelve critical security threats specifically related to the shared, on-demand nature of cloud computing for cloud computing with the highest impact on enterprise business [5]:

1. *Data Breaches* A data breach is an incident in which sensitive, protected, or confidential information is released, viewed, stolen, or used by an individual who is not authorized to do so.
2. *Weak Identity*, *Credential*, *and Access Management* Data breaches and enabling of attacks can occur because of a lack of scalable identity access management systems, failure to use multifactor authentication, weak password use, and a lack of continuous automated rotation of cryptographic keys, passwords, and certificates.
3. *Insecure APIs (Application Programming Interface)* Provisioning, management, orchestration, and monitoring are all performed using a set of software user interfaces (UIs) or application programming interfaces. These interfaces must be designed with adequate controls to protect against both accidental and malicious attempts to circumvent policy.

4. *System and Application Vulnerabilities* System vulnerabilities are exploitable bugs in programs that attackers can use to infiltrate a computer system for stealing data, taking control of the system or disrupting service operations.
5. *Account Hijacking* This is a significant threat, and cloud users must be aware of and guard against all methods such as phishing, fraud, and exploitation of software vulnerabilities to steal credentials.
6. *Malicious Insiders* A malicious insider threat to an organization is a current or former employee, contractor, or another business partner who has authorized access to an organization's network, system, or data and intentionally misused that access in a manner that negatively affected the CIAAA of the organization's information system.
7. *Advanced Persistent Threats (APTs)* These are a parasitical form of cyber-attack that infiltrates systems to establish a foothold in the computing infrastructure of target companies from which they smuggle data and intellectual property.
8. *Data Loss* Data stored in the cloud can be lost for reasons other than malicious attacks. An accidental deletion by the cloud service provider or a physical catastrophe such as a fire or earthquake can lead to the permanent loss of customer data.
9. *Insufficient Due Diligence* An organization that rushes to adopt cloud technologies and chooses cloud service providers (CSPs) without performing due diligence exposes itself to a myriad of commercial, financial, technical, legal, and compliance risks.
10. *Abuse and Nefarious Use of Cloud Services* Poorly secured cloud service deployments, free cloud service trials, and fraudulent account sign-ups via payment instrument fraud expose cloud computing models such as IaaS, PaaS, and SaaS to malicious attacks. Malicious actors may leverage cloud computing resources to target users, organizations, or other cloud providers.
11. *Denial of Service* (*DoS*) Denial-of-service attacks are attacks meant to prevent users of a service from being able to access their data or their applications by forcing the targeted cloud service to consume inordinate amounts of finite system resources so that the service cannot respond to legitimate users.
12. *Shared Technology Issues* Cloud service providers deliver their services by sharing infrastructure, platforms, or applications. The infrastructure supporting cloud services deployment may not have been designed to offer strong isolation properties for a multi-tenant architecture (IaaS), re-deployable platforms (PaaS), or multi-customer applications (SaaS). This can lead to shared technology vulnerabilities that can potentially be exploited in all delivery models.

1.4 Security Challenges and Solutions for Cloud-SDN-NFV Integrated Software Infrastructure

Since virtualization, isolation, and identity and access management (IAM) are the common underlying technologies and techniques for cloud, SDN, and NFV, they are fundamental and critical in term of security to all these infrastructures. We discuss the security aspects and guidance for virtualization, isolation, and identity and access management in this section.

1.4.1 Security of Virtualization

With virtualization, the complete state of an operating system and the instances of the application software together with their associated virtual hardware, disks, and metadata are captured by the VM. This state can be saved in a file, and the file can be copied and shared. Creating a VM reduces ultimately to copying a file. VM is an essential component of the cloud, SDN, and NFV. In SDN, a virtual network is created (virtualized) from the underlying network resources, and its virtual image can be captured by a file. Within this file, VMs exist as network elements (switches, routers, and communication links) of the virtual network. In NFV, a single VM or multiple VMs capture the complete state of a VNF instance which can be recorded as a file.

In the architecture of these infrastructures, a hypervisor is a centerpiece that performs the task of virtualizing resources. Virtualization thus brings with it all the security concerns of the guest operating system, along with new virtualization-specific threats, including hypervisor attacks, inter-VM attacks, inter-virtual network attacks, and inter-virtual function attacks [4].

1.4.1.1 Fundamental Security Issues with Virtualization

This part describes a number of security issues pertaining to virtualization and virtual environments.

Software Life Cycle of Virtual Image Object The traditional assumption is that the software life cycle is sequential on a single line, so management processes progress monotonically along the sequence. However, the virtual execution object model maps to a tree structure rather than a line. At any point in time, multiple instances of the virtualized entity (e.g., VM, VNF) can be created, and then each of them can be updated, different patches installed, and so on. This problem has serious implications for security [16].

The Indefinite Attack in a Virtual Environment Some of the infected VMs, VNs (Virtual Network), and VNFs may be dormant at the system clean up time, and later, they could surface up and infect other systems. This scenario can repeat itself and guarantee that infection will perpetuate indefinitely. In the non-virtual environment, once an infection is detected, the infected systems are quarantined and then cleaned up.

Rollback VM Attack Rollback is a feature that reverts all changes made by a user to a virtual machine when the user logs off from the virtual machine. As the complete state of a VM can be recorded, the feature opens the door for a new type of vulnerability caused by events recorded in the memory of an attacker. The first scenario is that one-time passwords are transmitted in the clear and the protection is not guaranteed if an attacker can replay rolled-back versions and access past sniffed passwords. The second scenario is related to the requirement of some cryptographic protocols regarding the *freshness* of the random-number source used for session keys and nonce. When a VM is rolled back to a state in which a random number has been generated but not yet used, the door is left open for protocol hijacking [16].

Security Risks Posed by Shared Images A user of a public cloud such as Amazon Web Service (AWS) has the option to create an image (Amazon Machine Image, AMI) from a running system, from another image in the image store, or from the image of a VM and copy the contents of the file system to the bundle. Three types of security risks were identified and analyzed: (1) backdoors and leftover credentials, (2) unsolicited connections, and (3) malware. The software vulnerability audit revealed that 98% of the Windows AMIs and 58% of Linux AMIs had critical vulnerabilities [16]. Analysis of these risks is left as an exercise at the end of the chapter.

Hypervisor Security Another critical security issue in virtualized environments is hypervisor vulnerabilities. A hypervisor creates virtual resources (VMs, VNs, and VNFs) inside the SDI and has the ability to monitor each of them. This feature introduces a high security risk in terms of confidentially, integrity, availability, authenticity, and accountability. It may allow an attacker to view, inject, or modify operational state information connected with the SDI through a direct/indirect method, and as a result, the attacker is able to read/write contents of resources such as memory, storage, and other components of the SDI. Hypervisor hijacking is a type of attacks that allow an adversary to take control of a hypervisor and access all VMs created by that particular hypervisor or other less secure hypervisors in the infrastructure. In the worst case, it may even introduce misconfigurations in SDN controllers when integrated with NFV technology. Furthermore, existing errors or bugs inside a virtual function or a hypervisor may allow an attacker to compromise other virtualized network functions for more serious attacks.

1.4.1.2 Solutions and Guidance

Cloud Security Alliance (CSA Security Guidance V3.0) has produced guidance for critical areas of focus in cloud computing and offered recommendations on the following issues:

- *Virtual machine guest hardening* Proper hardening and protection of a VM instance can be delivered via software in each guest.
- *Hypervisor security* The hypervisor needs to be locked and hardened using best practices. The primary concerns should be the proper management of configuration and operation as well as physical security of the server hosting the hypervisor.
- *Inter-VM attacks and blind spots* VMs may communicate with each other over a hardware backplane, rather than a network, and as a result, standard-network-based security controls are blind to this traffic and cannot perform monitoring or in-line blocking. In-line virtual appliances help to solve this problem.

- *Migration of VMs* An attack scenario could be the migration of a malicious VM in a trusted zone, and with traditional network-based security control, its misbehavior will not be detected. Installing a full set of security tools on each individual machine is another approach to add a layer of protection.
- *Performance concerns* Installing security software for physical servers onto a virtualized server can result in severe degradation in performance. Security software needs to be virtualization-aware.
- *Operational complexity from VM sprawl* The ease at which VM's can be provisioned has led to an increase in the number of requests for VM's in typical enterprises. This creates a larger attack surface and increases the odds of misconfiguration or operator error opening a security hole. Policy-based management and use of a virtualization management framework are critical.
- *Instant-on gaps* A VM can be started and stopped with ease, and this creates a situation where threats can be introduced into the gap when a VM is turned off and when it is restarted, leaving the VM vulnerable. Best practices include network-based security and virtual patching that inspects traffic known attacks before it can get to a newly provisioned or newly started VM.
- *Virtual machine encryption* VMs are vulnerable to theft or modification when they are dormant or running. The solution to this problem is to encrypt VM images at all times, but there are performance concerns.
- *Data comingling* There is concern that different classes of data (or VM's hosting different classes of data) may be intermixed on the same physical machine. *VLAN*, *firewalls*, *and IDS/IPS* should be used to ensure VM isolation as a mechanism for supporting mixed model deployments. Data classification and policy-based management can also prevent this.
- *Virtual machine data destruction* When a VM is moved from one physical server to another, enterprises need the assurance that no bits are left behind on the disk that could be recovered by another user or when the disk is de-provisioned. Zeroing memory/storage encryption of all data are solutions to this problem. Encryption keys should be stored on a policy-based key server away from the virtual environment.
- *Virtual machine image tampering* Pre-configured virtual appliances and machine images may be misconfigured or may have been tampered with before you start them.
- *In-motion virtual machines* The unique ability to move VMs from one physical server to another creates complexity for audits and security monitoring. In many cases, VMs can be relocated to another physical server (regardless of geographical location) without creating an alert or trackable audit trail.

1.4.2 Security by Isolation

Isolation is a technique for separating or partitioning different concerns that can be used for both resource management and security purposes. For example, process isolation in the time-sharing operating system is realized with virtual address space, and network isolation in the early network operating system is realized with a firewall. In network management, system management, and service management, isolation is used to identify, detect, and isolate faults, misconfiguration, and performance issues. Security isolation has been a key approach to system and network security. Virtualization has been adopted by the systems community as the technique of choice for providing isolation.

The responsibility of the infrastructure service provider (ISP) is to provide a secure infrastructure that ensures tenant's virtual machines are isolated in a multi-tenancy environment, and the various networks within the infrastructure are isolated from one another. Virtual networks can be one or many networks over which virtual machine traffic flows. Isolation of virtual machines within this network can be enhanced with the use of virtual firewall solutions that set firewall rules at the virtual network controller. Although virtual machines are often marketed as the ultimate security isolation tool, it has been shown that many existing hypervisors contain vulnerabilities that can be exploited. In a multi-tenant environment, *traffic isolation*, *address space isolation*, *performance isolation*, and *control isolation* are often required for different purposes. Traffic isolation prevents any data packets from leaking between tenants. Address space isolation allows the tenants to isolate their network by choosing their end-host IP and media access control (MACs) addresses independently from each other. Control isolation enables the tenants to control and configure their network without affecting other tenants [23].

The design of classical security devices is unable to protect the components of virtualized environments, since the traditional security depends on physical network devices and these devices cannot see the significant security activities inside virtualized environments [12]. Isolation will become an important technique for monitoring virtual security boundaries.

1.4.2.1 Isolation Classification

In this section, we classify different types of isolations and their potential usage:

Tenant Isolation In a cloud configuration, tenants share the same underlying physical infrastructure. Without network isolation, tenants could intentionally or unintentionally consume a large part of the network, intrusively see data on the network that does not belong to them, or invoke breaches such as unauthorized connection monitoring, unmonitored application login attempts, malware propagation, and various *man-in-the-middle* attacks.

Domain Isolation In order to label packets and enforce the isolation policies, it is necessary to determine the domain for each data flow. Each domain is associated with a set of input ports of the edge switches. Since the architecture distinguishes intra-tenant, inter-tenant, and external communications, the controller needs to check to which IP range the destination IP address belongs. There is a separate database table for mapping public IP addresses to the tenants who have been allocated such addresses.

Data Isolation Customers in fields such as banking or medical records management often have very strong data isolation requirements and may not even consider an application that does not supply each tenant with its own individual database.

VM Isolation A hypervisor divides the host hardware resources among multiple VMs. It coordinates all accesses by VMs to the underlying hardware resources and thus provides the necessary isolation between the virtual machines. In other words, VMs can share the physical resources of a single computer and remain completely isolated from each other as if they were in separated physical machines [33].

Traffic Isolation in Hypervisor-Based Environments Network traffic isolation is through the creation of segmented networks. In physical network isolation, network interface cards will be dedicated to a specific application or group of applications, and thus physical segmentation is provided between networks. In logical/virtual network isolation, software such as VLAN or network interface virtualization is used. Each interface is assigned a unique IP and MAC address; thus, each is logically distinct. The VLAN tagging can be defined in the host server to isolate network traffic further. Traffic for multiple applications share the same physical interfaces, but each application sees only the network traffic and resources assigned to it and cannot see traffic or resources assigned to other applications.

Traffic Isolation in Zones-Based Environments Similar to hypervisor-based virtualization, when a zone is provisioned, one or more network interfaces are presented, and the IP stack is enabled. The IP and MAC addresses are configured on the logical interface. Routing policies and network security can be hardened in these zones when the zones are provisioned.

Network Isolation Any isolated virtual network can be made up of workloads distributed anywhere in the data center. Workloads in the same virtual network can reside on the same or separate hypervisors. Additionally, workloads in several multiple isolated virtual networks can reside on the same hypervisor. Virtual networks are also isolated from the underlying physical infrastructure. Because traffic between hypervisors is encapsulated, physical network devices operate in an entirely different address space than the workloads connected to the virtual networks.

Network Segmentation Network isolation is between discrete entities. Network segmentation applies to homogeneous entities, e.g., protection within a group. Traditionally, network segmentation is a function of a physical firewall or router, designed to allow or deny traffic between network segments or tiers. For example, segmenting traffic between a web tier, application tier, and database tier. In a virtual network, network services that are provisioned with a workload are programmatically created and distributed to the hypervisor vSwitch. Network services, including L3 segmentation and firewalling, are enforced at the virtual interface.

1.4.2.2 Standard Network Security Solutions by Isolation

With compliance and regulatory requirements, network isolation along with network security has become essential elements of any service infrastructure deployment. The technology used for network traffic isolation does not always cover issues with security breaches that stem from external networks, side-channel attacks, or regulatory concerns between tenants. Network security is built on top of network isolated traffic. Standard security solutions include:

- *Network Firewalls*: Firewalls are often situated at the edges of networks to filter potential security threats coming from untrusted sources. Network firewalls may be hardware devices, software such as soft switches, or a combination of both.
- *LAN Tagging*: Tagging allows multiple logically separated networks (VLANs) to use the same physical medium. Thus, two separate VLANs cannot communicate with each other. VLAN configurations are performed at the switch and define the mapping between VLANs and ports. Packets sent by a virtual network interface on a VLAN cannot be seen by virtual interfaces on other VLANs, and broadcast and multicast packets sent from a virtual network interface on a VLAN will be distributed only to the network interfaces on the same VLAN.
- *Role-Based Security*: On the client side, the user devices must have hardened user authentication. On the database server side, role-based security, or role-based access control (RBAC), needs to be employed.

1.4.3 Security of Identity and Access Management

Identity and access management (IAM) is considered as one of the most critical and challenging security concerns in both physical and virtual infrastructure. The identity and access management concentrates on authentication, authorization, and administration of identities. The major concerns in IAM are related to identity verification of each entity, granting a correct level of access to cloud resources, policy managements, and role-based access controls. IAM architectures are more complex and different in cloud infrastructure in comparison to traditional IAMs since they have to deal with virtual functions and their dynamic changes. The aim of IAM is to prevent unauthorized access to physical and virtual resources as this can jeopardize the reliability, integrity, confidentiality, and availability of user's services and data. Security challenges such as identity theft, phishing, unauthorized access, and data tampering are associated with a weak identity authentication and access control mechanisms in cloud infrastructure. Identity management authenticates identification for individual entities like tenants or services by keeping their privacy from one another. Access control deals with authorization and policies to ensure only authorized entity has permission to access services [14]. The NFV technology brings further complexity in designing IAM architectures as components are virtualized and capable of changing dynamically within the infrastructure. Providing a dynamic and on-demand IAM architecture that can protect cloud against known and unknown attacks is one of biggest security challenges.

Cloud services are accessible through various types of virtual devices and applications with different privileges and authentication methods. Each cloud application can transmit user data and credentials with different policies (encrypted or un-encrypted), and this process can expose a serious vulnerability for man-in-the-middle attack [10].

Usernames and passwords have always been used as a long-time mechanism for authentication. However, vulnerabilities such as weak password policies, nonrotational password, and shared password among different cloud users and resources can expose sensitive information to attackers. There are different types of attacks exploiting vulnerabilities of IAM mechanisms to disclosure cloud tenant's data and information such as [10]:

Phishing Attack The aim is to collect cloud customer information such as their login credentials and credit card numbers, social numbers, etc. The attackers are launching their attacks by exploiting vulnerabilities in IAM methods that have no support for user-centricity, weak password policy, and weak web application controls.

Side-Channel Attack Since multi-tenancy and virtualization enable resource sharing among tenants (trusted/untrusted), cloud infrastructure could be a target of side-channel attacks such as time and bandwidth monitoring attacks. This kind of attacks occurs as a result of weak and improper distributed and structured access control architectures. The attacker can collocate its malicious VM and perform a side-channel attack and leak sensitive information relevant to cloud service provider or hypervisor that hosting targeted VM. This type of attacks can bypass access control policies of either the hypervisor or VM's guest OS, and access through shared resources belongs to another VM in the same platform [10].

Data Tampering Attack in Cloud This type of attacks is referred as unauthorized data modification related to identification of cloud service customer in an identity data store within the cloud. The attack can target existing cloud resources and services. It happens due to loopholes and misconfiguration of access control methods inside the cloud infrastructure.

Identity Forgery/Spoofing Attack Lack of multifactor authentication and improper access control method can lead to unauthorized copying or modification of identity tokens or credentials issued from cloud providers or trusted cloud authorities. Furthermore, this kind of attacks helps attackers in committing fraud and identity theft.

To prevent those threats, IAM mechanisms must be placed in each of the following layers [14]:

- *System layer* Only users with acceptably defined policy rules can access hosts or systems in cloud environment.

- *Application layer* Accessibility to any cloud application or its functions must be governed by access control rules, and access is only permitted after confirming the identity of cloud user.
- *Network layer* Since the cloud is a multi-tenancy architecture, it is critical that tenants be unable to see any portion of the network and its underlying systems in the cloud network unless access policies allow them.
- *Process layer* The way a user can use or run functions and processes of a cloud application should be strictly defined based on access control rules and policies.

1.5 Case Study: Security of OpenStack Platform

OpenStack is known as an IaaS cloud platform based on sharing storage, compute, and network resources. OpenStack is a collection of open-source technology projects with various functional components. OpenStack is an example of an integrated software-defined infrastructure involving ETSI NFV architecture framework, SDN network infrastructure, and cloud IaaS. It provides an automated infrastructure for cloud users. The OpenStack uses SDN technology to generate automated network infrastructure, NFV to create VNFs, and cloud to orchestrate and manage services. It is an IaaS cloud solution based on the integration of numerous services that interact through a set of OpenStack APIs, which is available to all cloud users. OpenStack consists of several main components; each represents a specific task within the OpenStack infrastructure. Figure 1.5 shows an overview of OpenStack with its main components [21].

The fundamental components of OpenStack are known as compute (Nova), network (Neutron), block storage (Cinder), object storage (Swift), identification (Keystone), image (Glance), dashboard (Horizon), and orchestration (Heat).

1.5.1 Security Challenges and Threats in OpenStack

As a cloud infrastructure platform, integrated with SDN and NFV technologies, OpenStack inherits all traditional security issues for cloud as well as issues introduced by SDN and network functions virtualization discussed earlier in the chapter. In this section, we present major security challenges in OpenStack and their recommended solutions.

Hypervisor Security Nova is responsible for the management of virtual machines through a virtual layer supported by various types of hypervisors. This OpenStack multi-hypervisor is prone to security challenges related to hypervisor security. Another major hypervisor security issue in Nova is related to compatibility and trust relation between different types of hypervisors and their configuration from different vendors.

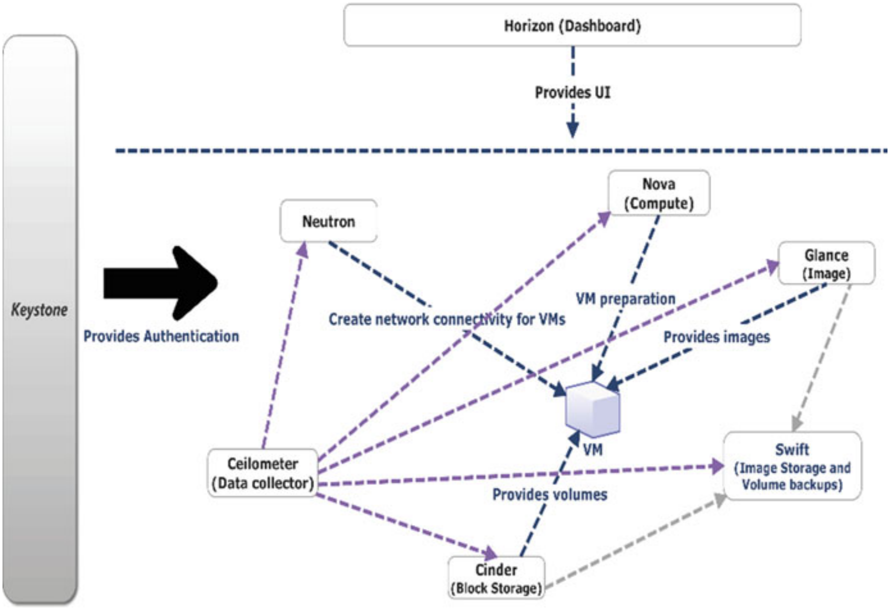

Fig. 1.5 Overall picture of OpenStack components and their relation

Neutron Vulnerability According to the CVE (*Common Vulnerabilities and Exposures*) list, one of the security issues of OpenStack Neutron is related to existing vulnerabilities of IPtables firewalls. This vulnerability enables attackers to bypass deliberate MAC- and DHCP-spoofing security mechanisms. Neutron can be a victim of denial-of-service (DoS) attack as a result of abnormal Dynamic Host Configuration Protocol (DHCP) discovery messages or non-IP traffic. The attacker can exploit software vulnerabilities within Neutron virtual machine and launch a DoS attack.

Identity Service It is one of the most critical components of the OpenStack architecture that is responsible for the authentication and authorization of users and component in OpenStack. It keeps records of policies and roles of users and tenants of the infrastructure. Keystone/identity service is known as the identity management component of OpenStack. OpenStack components will access their required information through a REST API. It also permits various access control methods such as username and password, token-based systems, and role-based methods. Keystone uses two authentication methods which are based on UUID (universally unique identifier) and standard PKI (public key infrastructure) token. Keystone can be targeted for denial of services, reply attacks, and information disclosure attacks if an attacker is able to bypass defined access control policies and gains access through user credential when sending username and password in a clear-text format or storing them in keystone logs. Since it is only based on tokens, an attacker can gain user's privilege by compromising the user token [6].

OpenStack API It is a RESTful web services endpoint that enables access to OpenStack components. In OpenStack majority of the APIs are based on HTTP web services, and they do not use SSL/TLS for encrypting data. Malicious users can try to exploit existing vulnerabilities of these APIs to send malformed inputs, long input porously, or call unauthorized functions to launch attacks [1].

Horizon Vulnerability A vulnerability known as session fixation vulnerability was discovered in OpenStack Horizon. In OpenStack, a client's session state (including authorization token) is stored in cookies, so if an attacker steals the cookies, he/she can act as a legitimate user and perform harmful actions [24, 29]. Another security issue is related to inability of password reset by users [1].

1.5.2 OpenStack Security Solution Recommendation

- *Nova security* It is responsible for the provision and management of virtualized compute resources. Since Nova is based on multi-hypervisors technique, it is recommended to store hypervisor logs in a secure remote storage. Furthermore, hypervisors security could be improved by placing each hypervisor in a separate context. One solution is to use sVirt or SELinux/AppArmor for placing hypervisors in separate security context. It is recommended to use TLS for any communication between Nova components and other Nova services like Keystone and Glance [31].
- *Neutron security (previous quantum)* This component provides virtual network connectivity and IP addresses for each VM instance within the infrastructure. It is recommended to use an SDN controller that can automate network configuration dynamically such as Juniper Contrail SDN controller or Brocade SDN Controller. To provide isolation in network structure, it is suggested to establish L2 isolation with VLAN segmentation and tunneling using GRE (Generic Routing Encapsulation) or VXLAN (Virtual Extensible LAN) or other protocols. Additionally, to avoid DoS/DDoS attacks, it is useful to apply network resource quotas for existing tenants. OpenStack network security can be achieved by using security groups and firewalls. Security groups mechanism provides traffic security between the east and west traffic (intra-VLAN traffics) and firewalls to protect OpenStack network north to south traffic (inter-VLAN traffic and edge traffic) [31].
- *Keystone security* Because of various limitations in Keystone related to password strength, expiration time, and fail attempt lock down policies, it is recommended to use password policy enforcement or deploy third-party authentication systems such as *LDAP* and *Active Directory*. Due to existing threats like identity theft, information disclosure, and spoofing, it is critical to use two-factor authentication mechanisms. Since Keystone provides token for the authentication and authorization process, it is useful to use Fernet token (based on cryptographic authentication method using symmetric key encryption) designed for *REST APIs* instead of standard and less secure tokens [31].

- *Cinder security* This component provides block-level storage to store device images used by Nova for installing a VM instance. Since all images are stored as files inside Cinder, if an attacker can access to those files, he/she can run a malicious instance inside the network. It is critical that only the user with the highest privilege (e.g., root) can have access right to Cinder configuration files [31].

1.6 Integrated Software-Defined Infrastructure Security

SDN, NFV, and cloud all share the *software-defined* concept where physical resources are virtualized into software components. In fact, they share the underlying physical infrastructure and the virtualization layer and require controllers and orchestrators to provision services. Naturally, SDN, NFV, and cloud evolve into an integrated software-defined infrastructure or software-defined system (SDS) for optimizing the use of resources, eliminating the redundancy in their structure, and providing a richer set of services on demand. The security of such an integrated software-defined infrastructure will entail more than just the security issues common to all domains, the security issues specific to each domain, the security gaps among them, and the security of the overall infrastructure. The security architecture of the infrastructure and its own security must be considered. This section elaborates on the software-defined security (SDSec), reviewing its development and describing our SDSec Service—SDS$_2$.

1.6.1 SDSec Concept

Traditional security mechanisms are not able to deal with virtualized environments. The design of classical security devices is unable to protect the components of virtualized environments, since the traditional security depends on physical network devices and these devices cannot see the significant security activities inside virtualized environments [12]. In order to combat security attacks where attackers make use of software to exploit the vulnerabilities of our infrastructures and virtualized agents to attack our infrastructures from anywhere and on multiple fronts instantaneously, we need to deploy the very tools and technologies of the attackers. SDSec is a new approach in designing, deploying, and managing security by separating the forwarding and processing plane from security control plane, similar to the way that SDN abstracts the forwarding plane from control and management plane. Such separation provides a distributed security solution, which scales as VMs by virtualizing the security functions, and provides the ability to manage it as a logical, single system [28]. In SDSec the security hardware appliances such as intrusion detection, firewalling, and others are replaced by software functions (virtual security functions in NFV). Orchestration of security components (virtual security networks and virtual network functions) into security services is the task of an orchestrator in the layer above the controller.

Several products that consider SDSec approach have been developed [7, 12]. Catbird implements a number of features and attributes that distinguish the SDSec approach from traditional security approaches. Catbird consists of two main elements: C*atbird control center* and *a set of virtual machine appliances (VMAs)* implemented as VMs. The system configures a mesh topology, where the Catbird control center is located at the center of the network as the policy enforcement point to manage and distribute the security controls across the connected VMAs. For every virtual switch, there is a Linux-based VMA (virtual memory address) implemented inside it, executing different security tasks through a hypervisor interface [19].

vArmour is another SDSec solution that exploits the benefits of virtualized environments. The architecture of vArmour is like any software-defined system architecture, where the control plane is decoupled from the forwarding plane. The vArmour Distributed Security System consists of a logically centralized controller and multiple autonomous enforcement point appliances connected by an intelligent fabric and constitutes a security (SDSec) service layer to enforce a security rule to a whole data center [20].

vShield is another solution for VMware vCloud. vShield provides the customer the ability to build policy-based groups and establish logical boundary between them. vShield integrates of several components: vShield App and Zones protects the virtual data center applications by creating segmentation between enclaves or silos of workloads. vShield Edge secures the edge of the virtual data center boundary and protects the communication between segmentations. vShield Endpoint offloads antivirus processing. vShield Manger provides a centralized control point to manage all vShield components [34].

1.6.2 Software-Defined Security Service (SDS$_2$) Architecture

We propose SDS$_2$ as a SDSec Service that uses cloud virtual resources and can be deployed by cloud provider to protect its integrated infrastructure.

SDS$_2$ exploits four main concepts: logically centralization of security control, virtualization of security connectivity, security functions virtualization, and orchestration of virtual resources. Applying the NFV concepts for security, virtualization technologies are used to implement virtual security functions (VSFs) on a VM or an industry-standard commodity hardware. These virtual security functions can be created on demand and moved to or instantiated in strategic locations in a software-defined dynamic virtual network environment. Applying the SDN concepts for security, network virtualization is deployed to provision virtual security networks (VSNs) connecting virtual security functions. A logically centralized SDSec controller forms a domain-wide view of the underlying network of virtual security functions. The SDSec controller is able to program, configure, and control the VSFs autonomously. Applying cloud computing concepts for security, physical storage, network, and computing resources are virtualized to accommodate virtual network functions, virtual security networks, and virtual security storage. Cloud platform is used for orchestrating the provisioned security components to provide

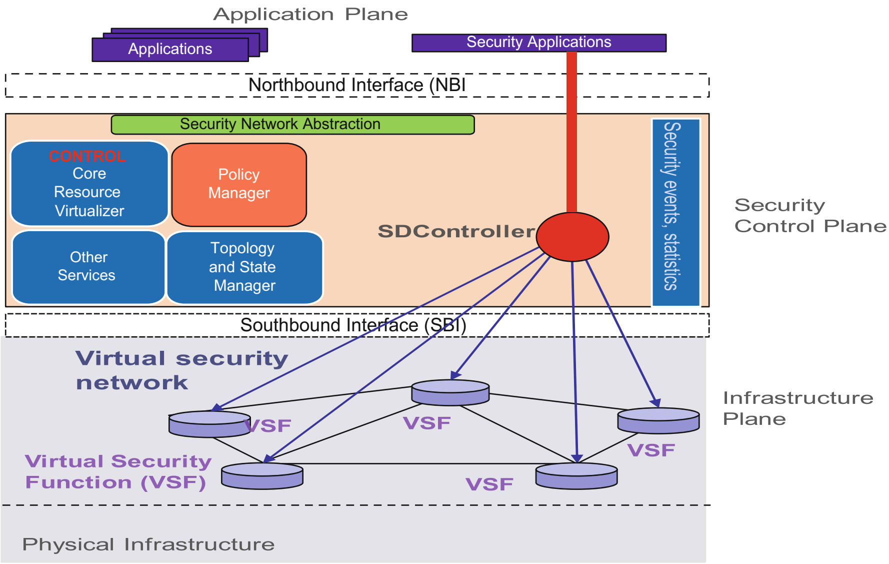

Fig. 1.6 SDS_2 architecture

security services for the target cloud infrastructure. The proposed SDS_2 architecture is shown in Fig. 1.6. It consists of three separate planes: the security application plane, the security control plane, and the security infrastructure plane or data plane. The SDSec control plane, which includes one or more SDSec controllers, provides an abstraction to build security services over virtual security elements. It is considered as a SDSec network operation system that provides basic security services via interfaces: the southbound interface (SBI) to network devices and the northbound interface (NBI) to security applications.

1.6.2.1 SDS_2 Controller

Like an SDN controller, SDS_2 controller is the brain of the whole security system, controlling its components and operations. It has a global view of its virtual security network and interconnected virtual security functions. The SDS_2 controller is similar to an SDN controller in that it consists of multiple components, but they deal only with security functions and security services. Security policy manager, topology and security state manager, and virtual security functions manager are its main components.

The SDS_2 controller has a complete topological graph of the connectivity of its virtual security functions (VSFs), allowing it to construct appropriate responses to attacks in real time. The controller will be able to construct service chaining of VSFs to create new security services to address emerging threats. Security intelligence is logically centralized in the software-based controller that maintains the global view of the security network and hence the global view of the security status of the protected system which appears to the security applications and policy engines as a single security element. It is essential that the SDS_2 controller is able to construct basic services and compose complex services into new services based on the capability of its underlying network of virtual security functions.

The security controller is programmable. It configures and manages all virtual security functions under its control through its virtual security network using a southbound interface. The SDS_2 allows the security manager to configure, manage, secure, and optimize network security resources (VSNs and VSFs) quickly via dynamic, automated programs.

1.6.2.2 SDS_2 Northbound Interface (NBI)

The SDS_2 controller communicates with its applications and security service orchestrators through its northbound interfaces. An intent-based northbound interface is appropriate for this as that allows the applications/orchestrators to express their required services in terms of what they need rather than how the required services are constructed and delivered. In an intent-based NBI model, users describe the requirements of their (security) application in their own domain-specific language, and the SDS_2 controller then translates and implements the required security application using its virtual security resources in the infrastructure layer. A security policy application can be implemented as an SDS_2 service [22].

1.6.2.3 SDS_2 Virtual Security Function

This is a security element or function implemented in software and deployed on a virtual resource such as a VM in a physical server (host). This is a generalization of NFV VF that abstracts a physical security appliance and deployed on a commodity server.

A VSF is created to perform a specific security function. It is a software object that can be created, instantiated, and operated on any VM. A VSF is a software entity that has a life cycle starting from the instant when it is created through its operation and then its termination. VSFs can be chained by a service chaining function to create a new security function. It can also be combined with other to create complex security functions. Typical VSFs include firewalls, virus scanners, intrusion detection systems, security gateways, and deep packet inspections. Other functions include policy/rule checkers, security metric meters, etc.

1.6.2.4 SDS_2 Southbound Interface (SBI)

OpenFlow and OFConfig can be used to configure SDS_2 VSFs, but they may be heavy for security purposes. A simple protocol may be designed to program, configure, and manage VSFs and allow them to report its operational status to the controller. It should be noted that VSFs are not switches or routers; they only perform their defined security functions and relay their data/status to their controller and other VSFs when directed such as in chaining operations.

1.6.2.5 Application of SDS_2 to Data Center Security

With the SDSec approach, we are able to design, implement, and modify the individual subsystems independently. A data center is an integrated cloud-SDN-NFV infrastructure whereby entities in it include physical resources (physical servers, routers, links, storage, and their interfaces), tenants, and their virtual resources (virtual networks, virtual machines, virtual storage, virtual services, and their virtual interfaces).

A common approach to managing system complexity is to identify a set of layers with well-defined interfaces among them. Layering minimizes the interactions among the subsystems and simplifies the description of the subsystems. Security of a system is often achieved by ensuring the integrity of its subsystem and authorized access to the system (subsystems) at their interfaces. A less common approach in system security is through isolation as discussed in Sect. 1.4.2. The security isolation approach can identify not only physical but also virtual boundaries that are missing in traditional security mechanisms. Furthermore, security isolation is effective in localizing security issues and can be tailored to deal with appropriate concerns.

With this in mind, SDS_2 can be implemented and offered as a security service to protect a data center. Depending on the data center, different numbers and types of virtual security functions can be instantiated, dynamic virtual security networks can be provisioned to interconnect those VSFs, and a logically centralized SDS_2 controller can be created on demand to serve the required security service. The provisioned SDS_2 configuration can be attached/imposed on the specified data center as dictated by its policies and architecture. The SDS_2 will enable security isolation through its software-based agents located at critical locations in both physical and virtual layers within the infrastructure under the control of the controller.

1.7 Summary

This chapter discusses the security of software-defined infrastructures with cloud, SDN, NFV, and technologies. It provides a brief description of cloud, SDN, NFV, and virtualization in terms of their defining features and summarizes critical security challenges of these infrastructures. It then discusses the fundamental underlying technologies and techniques (virtualization, isolation, and identity and access management) for software-defined infrastructures, their security issues, and solution guidance. Security OpenStack platform is described as a case study. Finally, the chapter reviews efforts in software-defined security and describes a new software-defined security service solution architecture.

SDSec is a promising research approach in software-defined infrastructure; however, there remain several challenges. SDI requires virtual networks from SDN, virtual network functions from NFV, and computing, storage, and orchestration resources from cloud, but there has not been a standard integrated architecture for SDI, and this presents a huge challenge in designing a sound framework for an SDI security architecture. Cloud, SDN, and NFV each have its own hypervisor in the virtualization layer, controller in the control layer, orchestrator in the application layer, and different protocols/interfaces between layers; this complicates the task of defining virtual boundaries where one can apply security measures to protect the overall infrastructure.

1.8 Questions

Q1. The communication messages between the SDN controller and an OpenFlow switch are transmitted over a secure channel that is implemented via a Transport Layer Security (TLS) connection over TCP.

(a) What would be security implications if TLS is not used in terms of threats and attacks?

(b) Consider three examples from widely deployed systems: Transport Layer Security (TLS), Internet Protocol Security (IPsec), and Secure Shell (SSH). The three approaches combine integrity and encryption in three very different ways—the first encrypts the MAC, the second applies MAC to the encryption, and the third uses independent MAC and encryption. But which is right? Are they all secure? Compare and discuss these three constructions. Can TLS be replaced by SSH or IPsec?

Hints: *Landwehr, C., Boneh, D., Mitchell, J. C., Bellovin, S. M., Landau, S., Lesk, M. E., "Privacy and Cybersecurity:The Next 100 Years," Proceedings of the IEEE, Vol. 100, May, 2012.*

Q2. Identify and analyze the types of security risk posed by shared images. For example, a user of AWS has the option to choose among Amazon Machine Images (AMIs): an AMI created from a running system, from another AMI, or from the image of a stored VM?

Q3. Since security appliances can be virtualized and implemented as software-based (virtual) security components (VSFs) that can be placed in the same virtual machine in an infrastructure, how are these VSFs isolated from one another? Discuss possible solutions.

Q4. One of the difficulties in handling security in current systems is related to defining certain isolation boundaries in both physical and virtual resources/infrastructures. Discuss how isolation can solve security challenges related to physical and virtual boundaries within the cloud infrastructure with integrated SDN and NFV technologies?

Q5. Since software-based VSFs are dynamically created and may migrate to different parts of the infrastructure, how can cloud provider protect their VSFs throughout their life cycle? Does a virtual network of VSFs play a part in their security and how?

References

1. Albaroodi H, Manickam S, Singh P (2014) Critical review of openstack security: issues and weaknesses. J Comp Sci 10(1):23
2. Alliance ODC (2013) Open data center alliance master usage model: software-defined networking rev. 2.0
3. Berde P, Gerola M, Hart J, Higuchi Y, Kobayashi M, Koide T, Lantz B, O'Connor B, Radoslavov P, Snow W (2014) ONOS: towards an open, distributed SDN OS. In: Proceedings of the third workshop on Hot topics in software defined networking, ACM, pp 1–6

4. CSA (2011) Security guidance for critical areas of focus in cloud computing V3.0
5. CSA (2016) CLOUD SECURITY ALLIANCE The Treacherous 12 – Cloud Computing Top Threats
6. Cui B, Xi T (2015) Security analysis of openstack keystone. In: Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2015. In: 9th international conference on, IEEE, pp 283–288
7. Darabseh A, Al-Ayyoub M, Jararweh Y, Benkhelifa E, Vouk M, Rindos A (2015) SDSecurity: a software defined security experimental framework. In: 2015 IEEE International Conference on Communication Workshop (ICCW), 8–12 June 2015. pp 1871–1876
8. ETSI G (2014) 003,"Network Functions Virtualisation (NFV); terminology for main concepts in NFV"
9. Govindarajan K, Meng KC, Ong H A (2013) literature review on software-defined networking (SDN) research topics, challenges and solutions. In: 2013 Fifth International Conference on Advanced Computing (ICoAC), IEEE, pp 293–299
10. Habiba U, Masood R, Shibli MA, Niazi MA (2014) Cloud identity management security issues & solutions: a taxonomy. Complex Adapt Syst Model 2(1):5
11. Hoang D (2015) Software defined networking–shaping up for the next disruptive step? Aust J Telecommun Digital Econ 3(4):48–62
12. Jararweh Y, Al-Ayyoub M, Benkhelifa E, Vouk M, Rindos A (2016) Software defined cloud: survey, system and evaluation. Futur Gener Comput Syst 58:56–74
13. Jim Metzler AMA (2016) The 2016 guide to SDN and NFV – part 4: Network Functions Virtualization (NFV) a status update
14. Kecskemeti G, Kertesz A, Nemeth Z (2016) Developing interoperable and federated cloud architecture. IGI Global, Hershey, pp 1–398
15. Le N, Hoang D (2016) Can maturity models support cyber security? In: The IEEE international workshop on Communication, Computing, and Networking in Cyber Physical Systems (CCN-CPS)
16. Marinescu DC (2013) Cloud computing: theory and practice. Morgan Kaufmann, Newnes
17. Mell P, Grance T (2011) The NIST definition of cloud computing National Institute of Standards and Technology, Gaithersburg
18. Milenkoski A, Jaeger B, Raina K, Harris M, Chaudhry S, Chasiri S, David V, Liu W (2016) Security position paper network function virtualization. Cloud Security Alliance-Virtualization Working Group
19. Networks C (2014) Catbird® 6.0: private cloud security
20. Networks v (2015) vArmour distributed security system: protecting assets in the world without perimeters
21. OpenStack (2015) OpenStack-Networking Guide
22. Pham M, Hoang DB (2016) SDN applications-The intent-based Northbound Interface realisation for extended applications. In: NetSoft Conference and Workshops (NetSoft), 2016 IEEE, pp 372–377
23. Ranjbar A, Antikainen M, Aura T (2015) Domain isolation in a multi-tenant software-defined network. In: 2015 IEEE/ACM 8th international conference on Utility and Cloud Computing (UCC), IEEE, pp 16–25
24. Ristov S, Gusev M, Donevski A (2013) Openstack cloud security vulnerabilities from inside and outside. Cloud Comp :101–107
25. Sahoo J, Mohapatra S, Lath R (2010) Virtualization: a survey on concepts, taxonomy and associated security issues. In: Computer and Network Technology (ICCNT), 2010 Second international conference on, IEEE, pp 222–226
26. Schubert L, Jeffery K (2012) Advances in clouds. Report of the cloud computing expert working group, vol 1. European Commission
27. Scott-Hayward S, Natarajan S, Sezer S (2015) A survey of security in software defined networks. IEEE Commun Surv Tutorials 18(1):623–654

28. SDxCentral (2017) SDN security challenges in SDN environments. https://www.sdxcentral.com/security/definitions/security-challenges-sdn-software-defined-networks/
29. Slipetskyy R (2011) Security issues in OpenStack. Master's thesis, Norwegian University of Science and Technology
30. Stallings W (2015) Foundations of modern networking: SDN, NFV, QoE, IoT, and cloud. Addison-Wesley Professional, Boston
31. Superuser O (2016) OpenStack security, piece by piece
32. Virtualization NF (2014) NFV security problem statement. ETSI NFV-SEC 1
33. Viswanathan A, Neuman B (2009) A survey of isolation techniques. University of Southern California, Information Sciences Institute, Los Angeles
34. VMware (2013) VMware vCloud networking and security overview
35. Xing Y, Zhan Y (2012) Virtualization and cloud computing. In: Future wireless networks and information systems. Springer, Dordrecht, pp 305–312
36. Young C (2016) Information security science-measuring the vulnerability to data compromises, 1st edn. Syngress Elsevier, Cambridge, MA
37. Zhou M, Zhang R, Zeng D, Qian W (2010) Services in the cloud computing era: a survey. In: Universal Communication Symposium (IUCS), 2010 4th International, IEEE, pp 40–46

Dr. Doan B. Hoang is a Professor in the School of Computing and Communication, Faculty of Engineering and Information Technology, the University of Technology Sydney (UTS). He was the Director of iNEXT—UTS Centre for Innovation in IT Services and Applications (2007–2016). Currently, he is the leader of VICS (Virtualized Infrastructures and Cyber Security) research lab. He is also a Research Associate and UTS Principal Investigator at the Open Networking Foundation (ONF). His current research projects include virtual resource optimization, software-defined (SD) architecture for provisioning IoT applications on demand, SDSecurity services, security metrics and maturity models for cloud, and trust assessment model for personal space IoTs and IoTs for assistive healthcare. Professor Hoang has published over 200 research papers. Before UTS, he was with Basser Department of Computer Science, University of Sydney. He held various visiting positions: visiting professorships at the University of California, Berkeley; Nortel Networks Technology Centre in Santa Clara, USA; the University of Waterloo, Canada; Carlos III University of Madrid, Spain; Nanyang Technological University, Singapore; Lund University, Sweden; and POSTECH University, South Korea.

Sarah Farahmandian is a Ph.D. student at the University Technology Sydney (UTS). She got her master's degree on information security, in particular, on securing cloud computing environment against distributed denial-of-service attacks from the University of Technology Malaysia (UTM). Her current research focuses on isolation and security in cloud computing, SDN, and NFV. She published several research papers in these areas.