4

SDN and NFV Security

CONTENTS

4.1 Introduction ................................................. 116
  4.1.1 An Overview of Security Challenges in NFV .......... 116
    4.1.1.1 NFV Threat Vectors ............................ 116
    4.1.1.2 NFV Security Goals ............................ 117
4.2 NFV Security .................................................. 118
  4.2.1 NFV Security Classification .......................... 118
    4.2.1.1 Intra-VNF Security ............................. 118
    4.2.1.2 Extra-VNF Security ............................ 119
  4.2.2 NFV Security Lifecycle ............................. 119
  4.2.3 Use Case: DNS Amplification Attack ................. 121
  4.2.4 NFV Security Countermeasures ....................... 122
    4.2.4.1 Topology Verification and Enforcement .......... 122
    4.2.4.2 Securing the Virtualization Platform .......... 123
    4.2.4.3 Network and I/O Partitioning .................. 123
    4.2.4.4 Authentication, Authorization and Accounting (AAA) ........................................ 124
    4.2.4.5 Dynamic State Management and Integrity Protection ........................................ 126
4.3 SDN Security ................................................... 126
  4.3.1 SDN Security Classification .......................... 126
    4.3.1.1 SDN Security Threat Vectors ................. 128
  4.3.2 Design of Secure and Dependable SDN Platform ...... 130
  4.3.3 SDN Data Plane Attacks and Countermeasures ....... 132
    4.3.3.1 SDN Data Plane Attacks ....................... 133
    4.3.3.2 SDN Data Plane Attack Countermeasures ...... 134
  4.3.4 SDN-Specific Security Challenges .................... 135
    4.3.4.1 Programmmability ............................... 135
    4.3.4.2 Integration with Legacy Protocols ................ 135
    4.3.4.3 Cross Domain Connection ....................... 135
  4.3.5 OpenFlow Protocol and OpenFlow Switch Security Analysis ........................................ 135
    4.3.5.1 Attack Model .................................... 136
    4.3.5.2 Protocol-Specific Analysis ..................... 138
4.4 Summary ....................................................... 138
Network Functions Virtualization (NFV) has emerged as a technology to provide a virtualized implementation of hardware-based equipment such as firewall, routers, Intrusion Detection System (IDS). Virtual Network Functions (VNFs) can be realized through virtual machines (VMs) or containers running on top of the physical server of cloud computing infrastructure.

Software-Defined Networking (SDN) acts as enabling technology for NFV. Despite the great benefits offered by SDN and NFV, the security, privacy and trust management still remains an important problem to be addressed. The architecture of SDN and NFV has been discussed in previous chapters. In this chapter, we discuss the security challenges faced by different components of SDN and NFV, some that are part of traditional network architecture, and some introduced because of the SDN/NFV framework that should be considered before deployment of SDN/NFV technologies in a cloud network or data-center.

We survey the threat model and security challenges in NFV in section 5.1. Section 5.2 has been dedicated to the classification of NFV security from the perspective of intra and inter-virtual network functions (VNF) design. We also introduce some of the defense mechanisms that are used in NFV to deal with current threat vectors. In section 5.3, we consider SDN security threat vectors. Section 5.2.2 provides guidelines for the design of a secured SDN platform. Additionally, we discuss the threat vectors specific to the SDN data plane, SDN architecture, OpenFlow protocol and OpenFlow switching software in this section.

4.1 Introduction

4.1.1 An Overview of Security Challenges in NFV

The NFV consists of two main function blocks, i.e., NFV Management and Network Orchestrator (MANO), and Network Function Virtualization Infrastructure (NFVI)). The security of NFVI requires ensuring security compliance with standard methods of authentication, encryption, authorization and policy enforcement to deal with both internal and external threats.
### TABLE 4.1: NFV Threat Vectors

<table>
<thead>
<tr>
<th>Threat Vector</th>
<th>Description</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>VNF Service Flooding</td>
<td>Attackers can flood the service or the network interface using attacks such as DNS lookup, multiple authentication failure attempts resulting in denial-of-service (DoS) in signaling plane and data plane.</td>
<td>Availability</td>
</tr>
<tr>
<td>Application Crashing</td>
<td>Attackers can send malformed packets to the services running in NFV environment and cause network service disruption (e.g., buffer overflow exploit).</td>
<td>Availability</td>
</tr>
<tr>
<td>Eavesdropping</td>
<td>Attackers can eavesdrop on sensitive data and control plane messages.</td>
<td>Confidentiality</td>
</tr>
<tr>
<td>Data Exfiltration</td>
<td>Unauthorized access to sensitive data such as user profiles.</td>
<td>Confidentiality</td>
</tr>
<tr>
<td>Data and Traffic Modification</td>
<td>Attacker can perform Man-in-the-Middle (MITM) attack on the network traffic in transit, perform DNS redirection or modify sensitive data on network elements (NE).</td>
<td>Integrity</td>
</tr>
<tr>
<td>Control Network and Network Elements</td>
<td>The attacker can exploit protocol vulnerabilities or implementation flaws to compromise a network. Additionally, attackers can exploit vulnerabilities on the management interface to take control of NE.</td>
<td>Control</td>
</tr>
</tbody>
</table>

#### 4.1.1.1 NFV Threat Vectors

Threats to the NFV network perimeter and core services can violate the service level agreements (SLAs), such as NFV data confidentiality and service availability. We analyze the threat vectors that can impact the NFV framework in the table 4.1:

In addition to these threat vectors, there can be insider threats where attacker inside the NFVI can make changes to the data on Network Element (NE) or make changes to the network configuration.

#### 4.1.1.2 NFV Security Goals

To define the security perimeter and its scope in NFV we need to identify the security goals in NFV environment at various levels of granularity. ETSI defines following security goals at high-level in an NFV environment:
• Establish a secure baseline of guidance for NFV operation while highlighting optional measures that enhance security to be commensurate with risks to confidentiality, integrity, and availability (CIA).

• Define areas of consideration where security technologies, processes, and practices have different requirements that of non-NFV systems and operations.

• Supply the guidelines for the operational environment that supports and interfaces with NFV systems and operations but avoid redefining any security considerations that are not specific to NFV.

The virtual network functions (VNFs) in an NFV environment often possess sensitive data and the NFV administrator should take care of data authentication in NFV workloads. The sensitive data authentication can consist of passwords, tokens, Cryptographic Keys, private keys and documents containing sensitive data. Each VNF can be responsible for one or more functions, and capabilities. Authorization for the use of these functions and capabilities should be performed using standard techniques, e.g., identity, trust, delegated or joint decision making and API security.

The security in NFV is not limited to managing the NFV network and endpoints, but secured mechanisms must be designed for the lifecycle management of VNFs. The VNF creation requires changes to networking, credentialing, license and configuration information. Guidelines for VNF creation using newly defined configuration or cloning from a previously created VNF must be in place. Well defined and secured mechanisms should be utilized for VNF lifecycle operations such as VNF deletion, Workload Migration, VNF configuration and patch management.

4.2 NFV Security
4.2.1 NFV Security Classification

NFV Security architecture can be considered from various perspectives. Security domains of NFVI can be classified into networking, Compute and Hypervisor domains as discussed by Yang et al [289]. ETSI [89] classifies the security domain of NFV into Intra-VNF security, i.e., security between the VNFs and Extra-VNF security, i.e., security external to VNF.

4.2.1.1 Intra-VNF Security

VNFs communicating with each other directly have special security requirements, since communication path is not restricted to the network level. The characteristics of Intra-VNF security include:
• Secured orchestration for and between the VNFs
• Flows between VNFs are not often through layer 3 firewall or any other security policy enforcement point.
• Service Chaining capabilities often need to be enforced if available.
• Requires security mechanism in intra VNF communication and resiliency to attacks.
• Security and virtual appliances need to be configured to be part of the traffic flow.

4.2.1.2 Extra-VNF Security

The VNF security is dependent upon the security of physical infrastructure, external services, and environment. The key issues that need to be considered for the security of NFV environment because of the factors external to the VNFs are:

• NFV deployment may span across several regulatory and jurisdiction domains, leading to multiple sets of Service Level Agreement (SLA) and Quality of Service (QoS) requirements. Extra-VNF security should have the ability to administer cross-border and domain requirements, e.g., Workload Migration from one public NFV tenant to a secured NFV tenant may impact the QoS or security of the destination NFV tenant.

• Authentication, authorization, and accounting across NFV domains across a mix of domains, humans and system entities. For instance, one NFV deployment can have multiple administrative domains, e.g., a) NFVI. b) SDN c) Orchestration d) VNFM e) Service Network.

4.2.2 NFV Security Lifecycle

The VNF life-cycle as shown in the Figure 4.1(a) comprises five phases, i.e., VNF development, instantiation, operation, enhancement, and retirement. The security management processes 4.1(b) for NFV should be embedded into these phases of VNF life-cycle. The scope of NFV security comprises of NFV framework, hardware, software and service platform that supports NFV. We consider hardware platform to be following the required NFV security guidelines and consider security for hardware platform out of the scope of this chapter.

We can segment the NFV architecture into components that can be a direct or indirect target of attacks. The goal of the attack can range from reconnaissance, service degradation, service disruption to unauthorized access to critical information in the NFV framework. Reynaud et al [229] have identified five critical assets in NFV framework that can be potential attack targets;
FIGURE 4.1: Virtual networking approaches enable different logical views of the underlying physical networks.

1. **Virtual Network Functions**: VNFs suffer from software vulnerabilities. They can be a source of an attack or target. The vulnerabilities like buffer overflow and DoS attacks against cloud and web-based services are typical threat vectors that can be caused by VNFs.

2. **Virtualization Layer**: The virtualization layer can be a target of many security attacks, e.g., malicious code execution on the physical host, Return Oriented Programming (ROP) based attacks, where an attacker can elevate the VM privilege, CPU resource monopolization attack, Data Theft and VM monitoring attacks as discussed by Riddle *et al* [230].

3. **Communication with and within NFV MANO**: The attacker can eavesdrop on the traffic between NFV MANO and NFVI. The attacker can perform a MITM attack is an attack vector targeting this particular communication channel.

4. **VNF Orchestrator and/or Manager**: NFV over OpenStack can be
targeted to ephemeral storage vulnerability (CVE-2013-7130). An attacker can steal Cryptographic Keys from other VNFs or steal root disk contents of the other users by exploiting this vulnerability.

5. **Virtualized Infrastructure Manager (VIM):** Attacks can target the infrastructure manager in the NFV, e.g., Ruby vSphere console in VMWare vCenter Server suffers from privilege escalation vulnerability (CVE-2014-3790). This allows remote users to escape chroot jail and execute arbitrary code in the infrastructure domain.

### 4.2.3 Use Case: DNS Amplification Attack

The VNFs can be a target of DoS attacks. The goal of the attacker can be network resource exhaustion or impacting service availability. If the attacker can exploit a vulnerability present on old versions of some software of a VNF, e.g., CVE-2018-0794 (MS Office Remote Code Execution vulnerability). A huge volume of network traffic can be generated from compromised VNFs and directed towards other VNF present on the same Hypervisor in VNFI. For example, the Figure 4.3 shows NFVI comprising of a number of DNS servers as a component of virtual evolved packet core (vEPC).
The NFVI orchestrator can spawn additional DNS servers on-demand depending upon the traffic load in the network. In step (1) of the attack, the attacker spoofs the IP address of the victims and launches a large number of malicious DNS queries. The orchestrator realizes the traffic load in the network is above the normal threshold and spawns out additional vDNS VNFs in Step (2). Multiple recursive DNS servers in the network respond to the victim, and in-effect receive amplified DNS query responses - Step (3), which can ultimately result in service unavailability or disruption.

4.2.4 NFV Security Countermeasures

The ETSI NFV Industry Specification Group (ISG) and NFV Security Expert Group has identified some key areas of concern in NFV security and security best practices to deal with these security problems. In this section, we discuss NFV security countermeasures based on ETSI specification and state of the art research works in the field on NFV security.
4.2.4.1 Topology Verification and Enforcement

The network topology and communication of data-plane, control plane and management plane in NFV should be validated, ensuring the following specifications.

**Data Plane**

- Intra-host communication (the communication between VNFs on the same host).
- Inter-host communication (communication between VNFs on different Hypervisors).
- Communication path between VNFs and physical equipment.

**Control and Management Plane**

- The communication paths within the management and orchestration system (MANO).
- Paths between MANO and the Virtualized Infrastructure.
- Paths between MANO and the hardware infrastructure.
- Paths between MANO and the managed VNFs.

The topology of these two networks must be validated individually as well as together. The topology validation can be divided into different levels to manage the complexity of the operation. For instance, the physical and logical topology (VLAN, GRE) of the underlying infrastructure can be checked first, followed by validation of ports of each virtual forwarding function in VNF environment.

4.2.4.2 Securing the Virtualization Platform

An important security assumption in NFV is that VNF provider has to trust the virtualization platform on which various VNFs have been hosted. Additionally, the platform should also have some mechanism to ensure the trust in VNFs. One way of providing platform security is Secure Boot [79] technology. Secure boot can help maintain validation and assurance of Boot Integrity. There are several assurance factors that are part of Boot Integrity, including authenticity, configuration management, local attestation, certificates, Digital Signatures, etc.

A malicious attacker can tamper with the initial boot process of VNF to load malicious code during the VNF launch cycle. Secure boot can provide assurance that the code loaded in VNF execution environment is authentic, and has not been tampered with. Trusted boot process in coordination with VNF manager can provide validation during the VNF launch and installation stages.
4.2.4.3 Network and I/O Partitioning

One of the main purposes of virtualization is the isolation of VMs from crashes, loops, hangs, security attacks from other VMs. The objective is hard to be realized when:

- Granularity at which network boundaries have been defined or resources have been allocated is too coarse.
- Use of workloads is highly variable.

There are various attack vectors that can target Hypervisor resources in NFV environment, e.g., a) Local storage attack can be mounted to fill up Hypervisor local storage with logs b) remote connection attacks (remote control channel degradation).

The network resource is a critical network function. In addition to the malicious users, sometimes a large number of remote users can send a request to local resources, and it is hard to distinguish between the normal request from malicious traffic. An efficient QoS scheme can be used to ensure that critical tasks are given priority in case of high network demand. Additionally, the network must be partitioned into fine-grained segments to localize the threat only to the infected segment of the network.

Resource isolation is another mechanism to achieve fine-grained partitioning. Some methods to achieve isolation include a) Physical segregation of hardware resources b) Rate-limiting the usage of resources VNF can reserve c) Dividing available resources between competing demands using efficient scheduling mechanism, e.g., round-robin or fair-queue bandwidth scheduling.

4.2.4.4 Authentication, Authorization and Accounting (AAA)

Reliable mechanisms to ensure the identity and accounting facilities at the network and virtualization layer can be incorporated to achieve AAA.

Introduction of NFV can bring new security issues for AAA. The identity and accounting facilities span across two regions, i.e., network infrastructure layer (identifying the actual tenant), and network function layer (identifying the particular user), as shown in the Figure 4.4. Some of the AAA issues that can occur in NFV framework include:

1. **Authentication**: Unauthenticated disclosure of user information at the layers that are not supposed to consume certain identity attributes.

2. **Authorization**: Privilege escalation by wrapping unrelated identities not verifiable at a particular layer.

3. **Accounting**: Lack of accounting at different layers of network infrastructure, e.g., at the granularity of tenant can allow an attacker to over-subscribe the allocated resources in NFVI.
A generalized AAA scheme is required to support identity and access management at each tenant and VNF level. The current AAA mechanisms assume there will be a single identity, policy decisions, enforcement points, and single accounting infrastructure. Achieving this in current NFV framework using mechanisms such as tunneling, can scalability and resiliency concerns. Although achievement of all possible objectives - security enforcement, scalability, flexibility, manageability are difficult to achieve, it is important to find the appropriate combination supporting the trust framework. Some of the countermeasures include:

- Authentication of VNF images.
- Authentication of users requesting access to NFV MANO function blocks.
- Updates to authorized users and managers in the suspended/offline images.
- Authorization on interfaces/APIs between different function blocks.
- Support for real-time monitoring, logging, and reporting on SLAs, reliability, and performance.
- Traffic packet acquisition at full line rate and traffic classification and accounting per-subscriber, per user and per application.
• Policy decision functions that raise alarm when a specific threshold has been reached according to the detected policies and traffic.

4.2.4.5 Dynamic State Management and Integrity Protection

• **Dynamic State Management**: Online and offline security operations such as securely suspending a VM image, updating the access control lists (ACLs) in suspended images, secured live migration of VNFs. VNFC should be provisioned with the initial root of trust. All communication between the initial root of trust and VNFC should be strictly monitored.

• **Dynamic Integrity Management**: During the normal functioning of VNF, the VNF volume should be encrypted and Cryptographic Keys should be stored in a secure location. Trusted Platform Module (TPM) volume is one way of securely storing the keys. A software or hardware misconfiguration can cause VNF to crash, rendering the VNF in an unexpected state, which can cause security concerns. The crash events should be properly analyzed to ensure the integrity of VNF keys and passwords is maintained during the crash. The analysis of crash event should also consider external influences that should be mitigated to restore service. In case of crash events, the Hypervisor should also be properly configured to wipe out virtual volume disk to prevent it from unauthorized access.

4.3 SDN Security

SDN finds many applications in enterprise cloud and data-center network. The adoption of SDN can provide benefits in not only cloud management and orchestration but also cloud security. Thus, the security of SDN itself is quite an important area of research. The centralized design of SDN can introduce security challenges such as distributed denial-of-service (DDoS) attacks against the SDN controller. The SDN functional architecture can be divided into three layers, i.e., the application layer, the control layer and the data layer as discussed in previous chapters. Each layer can have multiple attack vectors. Additionally, the communication channel between layers, e.g., an application-control interface can be targeted to traffic modification and eavesdropping attacks.

4.3.1 SDN Security Classification

The relationships between SDN elements can introduce new vulnerabilities, which are absent in the traditional network. For instance, the use of
transport layer security is optional in the OpenFlow network. The nature of the communication protocol can thus introduce security issues such as DoS, fraudulent flow rule insertion, and rule modification as discussed by Scott-Hayward et al [242].

The Figure 4.5 highlights different components in SDN (1) application plane, (2) control plane and (3) data plane tier that can be subjected to attacks. For instance, there can be software vulnerabilities in SDN controllers (Opendaylight, ONOS, Floodlight). Additionally, the communication paths between three tiers (4) and (5) can face security attacks. We discuss some of the attack vectors against targetable components in detail below:

- **Application Plane:** The applications developed for telemetry, orchestration and other SDN operations can have security vulnerabilities. All the security issues that can be present in a typical web application such as Cross Site Scripting (XSS), Cross Site Request Forgery (CSRF) also apply to SDN. The malicious/compromised applications can allow spread of attack in the entire network.

- **Control Plane:** The control plane consists of one or more controller, e.g., OpenDaylight, POX, ONOS and other applications and plugins for handling different kinds of protocols. The attacker can generate traffic from...
spoofed IP address and send a huge volume of traffic to the controller as discussed by Kalkan et al [143]. The communication between the switch and the controller can be saturated using this attack, thus increasing service latency or in the worst case bringing down the controller.

- **Data Plane**: The attackers can poison the global view of the network, by forging the Link Layer Discovery Protocol (LLDP) packages. The attackers can also observe the delay in communication between the control plane and data plane applications using specially crafted packets. This can help in identification of controller application logic [97]. The attackers can also target the switches. The switch responsible for data plane flow rule updates often have limited memory and can be overflowed by generating a large number of flow rules.

- **Communication Channels**: The communication channel between switches and controllers (Southbound API), controllers and application plane tier (Northbound API) can be subjected to Man-in-the-Middle (MITM) attack as showcased by Romao et al [231], ARP Poisoning is one example of such security attacks. Other attacks showcased by authors that target communication channel include eavesdropping traffic between hosts, and stealthily modifying the traffic between hosts.

The table 4.2 summarizes some of the security issues associated with different components of SDN that we described above.

4.3.1.1 SDN Security Threat Vectors

In this section, we discuss some key Threats Vectors (TVs) in SDN in detail, and analyze if a better SDN platform design can help in dealing with security threats intrinsic and extrinsic to the SDN.

- **TV1 Fake Traffic Flows**: Faulty devices or malicious users can use DoS attack to target the TCAM switches in the SDN infrastructure, with the goal of exhausting the capacity of the TCAM switches. The problem can be mitigated by using simple authentication mechanism, but if the attacker is able to compromise the application server consisting of details of users, an attacker can use the same authenticated ports and source MAC addresses to inject forged authorized flows into the network.

- **TV2 Switch Specific Vulnerabilities**: The switches present in the SDN environment can have vulnerabilities. For instance, a vulnerability in Juniper OS (CVE-2018-0019) SNMP MIB-II subagent daemon (mib2d) allows a remote network-based attacker to cause the mib2d process to crash resulting in a denial of service condition (DoS) for the SNMP subsystem. A switch can be used to slow down the traffic in SDN environment, deviate the network traffic to steal information, or can be used to insert forged traffic requests with the goal of overloading the controller or the neighboring switches.
TABLE 4.2: Security Issues Associated with different layers of SDN

<table>
<thead>
<tr>
<th>Security Attack</th>
<th>SDN Layer Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>App Layer</td>
</tr>
<tr>
<td>Unauthorized Access</td>
<td></td>
</tr>
<tr>
<td>Unauthorized Controller Access</td>
<td></td>
</tr>
<tr>
<td>Unauthenticated Application</td>
<td>✓</td>
</tr>
<tr>
<td>Data Leakage</td>
<td></td>
</tr>
<tr>
<td>Flow Rule Discovery</td>
<td></td>
</tr>
<tr>
<td>Forwarding Channel Discovery</td>
<td></td>
</tr>
<tr>
<td>Data Modification</td>
<td></td>
</tr>
<tr>
<td>Flow Rule Modification</td>
<td></td>
</tr>
<tr>
<td>Malicious Applications</td>
<td></td>
</tr>
<tr>
<td>Fraudulent Rule Insertion</td>
<td>✓</td>
</tr>
<tr>
<td>Controller Hijacking</td>
<td></td>
</tr>
<tr>
<td>Denial of Service</td>
<td></td>
</tr>
<tr>
<td>Controller Switch Flooding</td>
<td></td>
</tr>
<tr>
<td>Switch Flow table Flooding</td>
<td></td>
</tr>
<tr>
<td>Configuration Issues</td>
<td></td>
</tr>
<tr>
<td>Lack of TLS</td>
<td></td>
</tr>
<tr>
<td>Policy Enforcement Issues</td>
<td>✓</td>
</tr>
</tbody>
</table>
• **TV3 Control Plane Communication Attack:** The control-data plane communication doesn’t require the presence of TLS/SSL security. Even if Public Key Infrastructure (PKI) is present in an SDN environment, complete security is not guaranteed for the communication channel communication. Research works highlight security issues with TLS/SSL [114]. A compromised Certificate Authority (CA), s vulnerable application can lead to an attacker gaining access in control plane channel of the SDN. The attacker can launch DDoS by using switches that are controlled by the control plane.

• **TV4 Controller Vulnerabilities:** The controller is the most important component in the SDN environment. A compromised controller can bring down the entire network. For example, an old version of SDN controller ONOS suffers from remote denial of service attack (CVE-2015-7516). The attacker can cause NULL pointer dereference and switch disconnect by sending two Ethernet frames with ether_type Jumbo Frame (0x8870) to ONOS controller v1.5.0. A combination of signature-based intrusion detection tools may not be able to find the exact combination of events that triggered a particular behavior and deem it malicious or benign.

• **TV5 Lack of Trust between controller and management applications:** Controller and management plane applications lack a built-in mechanism to establish trust. The certificate creation and trust verification between network devices in the SDN environment can be different from the trust framework between normal applications.

### 4.3.2 Design of Secure and Dependable SDN Platform

A secured and dependable SDN architecture as shown in Figure 4.6, having features such as fault-tolerance, self-healing, trusted framework and dynamic service provisioning capabilities can be used to deal with threat vectors discussed in the previous subsection. In this section, we discuss each of the security mechanisms that can be embedded into the design of the SDN framework.

1. **Replication:** Application and controller replication can help in dealing with cases of controller or application failures due to a high volume of traffic or software vulnerabilities. As shown in the Figure 4.5, there are three versions of the SDN controller providing replication. Additionally, the application B has been replicated on each controller. This approach can help in dealing with both hardware and software failure issues (accidental or malicious). Another advantage of replication is the isolation of malicious application while keeping the service consistency.

2. **Diversity:** The utilization of only one kind of software or operating system makes it easier for attackers to exploit a target. Diversity improves
FIGURE 4.6: Design of Secure and Dependable SDN.

the robustness and intrusion tolerance. As discussed by Garcia et al [98], utilization of a diverse set of OS makes a system less susceptible to intrusions. Diversity helps in avoiding the common faults and vulnerabilities since there are only a few intersecting vulnerabilities among diverse software or OS. In SDN management plane use of diverse controllers can help reduce lateral movement of an attacker and cascading system failures caused by common vulnerabilities.

3. **Automated Recovery:** In the case of security attacks, leading to service disruption, the proactive and reactive security recovery mechanisms can help in maintaining optimal service availability. When replacing a software, e.g., SDN controller, it is necessary to perform the replacement with new and diverse versions of the component. For example, if we plan to switch SDN controller OpenDaylight, we can consider an alternate version of controller software such as Floodlight, ONOS or Ryu providing similar functionality.

4. **Dynamic Device Association:** The association between the controller and devices such as OpenFlow switch should be dynamic in nature. For instance, if one instance of the controller fails, the switch should be able to dynamically associate with the backup controller in a secured fashion (proper authentication mechanism to detect good controller from mali-
Dynamic Device association feature helps in dealing with faults (crash or Byzantine). Other advantages include load balancing feature provided by diverse controllers (reduced service latency).

5. **Controller-Switch Trust:** A trust establishment mechanism between the controller and switch is important to deal with cases of fake flows being inserted by malicious switches. The controller can in basic trust establishment scenario maintain a whitelist of switch devices that are allowed to send control plane specific messages to the controller. In a more complex scenario, public key infrastructure (PKI) can be used to establish trust between the control plane and data plane devices. The behavior devices controlled by the controller can also be used to create a trust framework. The devices showcasing anomalous behavior can be put in quarantine mode by the controller.

6. **Controller-App Plane Trust:** The software components change behavior because of change in the environment. Additionally, the software aging can introduce security vulnerabilities. Controller and application plane components should use autonomic trust management mechanisms based on mutual-trust, delegated trust (3rd part such as the Certificate Authority to establish trust). The controller can utilize autonomic trust management for component-based software systems as discussed by Zheng et al [287]. Qualitative metrics such as confidentiality, integrity, and availability can also be leveraged to establish the trustworthiness of an application in the SDN framework.

7. **Security Domains:** Security domains help in segmenting the network into different levels of trust, and containment of the threats to only the affected section in the SDN framework. A security domain-based isolation can be incorporated to provide defense-in-depth for SDN environment. For example, the web-server application on one physical server should only interact with database back-end applications, and not any other application running in the same network. A white-list-based security policy composition with appropriate policy conflict checking mechanism can be utilized to achieve such security objectives. We discuss the segmentation policy creation, its key benefits, and real-world applications in detail in the next chapter.

The Table 4.3 summarizes the security solutions that we discussed in this sub-section and threat vectors TV1-5, that can be mitigated using these mechanisms. An important consideration while deploying the desired security solution or combination of solutions is cost-benefit analysis (delay introduced in the network, CPU/resource utilization) of these solutions in isolation as well as together.
### SDN Data Plane Attacks and Countermeasures

#### 4.3.3 SDN Data Plane Attacks

The communication between the SDN data plane and control plane opens the doors for highly programmable application development, it also introduces the possibility of new types of threat vectors in SDN data plane. The data plane attacks can dysfunction the anonymous communication in OpenFlow networks [294]. The attacker can also perform reconnaissance on the traffic channel between SDN data plane devices and controller to identify relevant information about controller software, e.g., version and type of controller (python or java application). This information can be utilized by the attacker to perform more targeted attacks against control plane software. Some key attacks that find origin in the SDN data plane include:

1. **Side-Channel Attacks**: The attacker can observe the processing time of the control plane in order to learn the network configuration as discussed by Sonchak et al [258]. The attacker craft probe requests corresponding to different layers of network protocol stack, e.g., ARP requests for MAC layer, TTL probe message for the IP layer. The requests can be sent to the controller along with some baseline traffic requests with a known response. By observing the response time and content difference between baseline traffic and the probe traffic, the attacker can observe version of OpenFlow, the size of switch flow table, network and host communication records, network monitoring policies, and the version of the controller software.

2. **Denial-of-Service (DoS)**: The devices send a connection request to the switching software in the data plane. If the switch consists of flow rule entry corresponding to the traffic pattern, traffic is forwarded out of the specific switch port. If the entry is missing (table-miss packets) the request is sent to the controller. A class of DoS attacks - data to control plane saturation attacks as discussed by Gao et al [97] can forge the OpenFlow fields with random values, that will lead to table-miss event in the switch. When a large volume of forged table-miss flows is sent to the controller as packet_in entries. The controller can be saturated since these packet_in

### Table 4.3: SDN Design Solution for dealing with Threat Vectors

<table>
<thead>
<tr>
<th>SDN Security Solution</th>
<th>Threat Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replication</td>
<td>TV1, TV4, TV5</td>
</tr>
<tr>
<td>Diversity</td>
<td>TV3, TV4</td>
</tr>
<tr>
<td>Automated Recovery</td>
<td>TV2, TV4</td>
</tr>
<tr>
<td>Dynamic Device Association</td>
<td>TV3, TV4</td>
</tr>
<tr>
<td>Controller-Switch Trust</td>
<td>TV1, TV2, TV3</td>
</tr>
<tr>
<td>Controller-App Plane Trust</td>
<td>TV4, TV5</td>
</tr>
<tr>
<td>Security Domains</td>
<td>TV4, TV5</td>
</tr>
</tbody>
</table>
messages will consume a large amount of switch-controller bandwidth and controller resources (CPU, memory). In case the controller decides to insert the flow entries in the switch, the TCAM table limit of the switch may be reached, which will prevent the legitimate traffic flows from being inserted in the switch flow table.

3. **Topology Poisoning Attacks**: This is a two-stage data plane attack, which utilizes forged control plane packets (LLDP packets) as a starting point. In the first attack, the attacker captures the OpenFlow LLDP packets and filters out the LLDP syntax. In the second step, the attacker sends the forged LLDP packets to the controller or replays them to other hosts in the network, in order to trigger the response from the connected switch to the controller. This can help an attacker in establishing a previously non-existent link between the switches. The attacker can utilize the modified topology to his advantage and launch MITM or a variation of DoS attack in the network.

4.3.3.2 SDN Data Plane Attack Countermeasures

1. **Side-Channel Countermeasure** The side-channel attacks rely on response time pattern, so a disruption in time-frequency can help counter side-channel attacks. Sonchak *et al* [258] propose timeout proxy on the data plane to normalize control plane delay. When the control plane fails to respond within a specified time duration, the timeout proxy sends default forwarding instruction to the request. The proxy reduces the long delay time of some long response packets to avoid side-channel attacks. The duration of response can also be randomized in order to counter the side-channel attacks.

2. **DoS Countermeasure** DoS attacks such as SYN flood can be inspected for existence of valid source and destination addresses. AvantGuard [253] extends the hardware of OpenFlow switches, and adds TCP proxy to send SYN-ACK as a reply for the TCP-based data to control plane starvation attacks. Another approach is to perform statistical analysis and flow classification to distinguish attack traffic from the benign traffic can help in the detection and prevention of DoS attacks.

3. **Topology Poisoning Countermeasure** The type of neighbor devices connected to OpenFlow switch can help in dealing with LLDP-packet-based Topology Poisoning Attacks as showcased by TopoGuard [120]. The control plane can identify neighbor device using packet hop distance, and other packet statistics. If a port first receives an LLDP packet, the neighboring device can be regarded as a switch. On the other hand, if a packet from first hop host is received, the neighboring device is regarded as host. The dynamic monitoring and probing can help in the reconstruction of neighbor topology. The drawback of this approach is
that it may allow attackers to forge neighbor device transfer from host to switch. Gao et al [97] propose more a granular classification of devices (host/switch/any/untested) to deal with the problem of topology poisoning faced by TopoGuard.

4.3.4 SDN-Specific Security Challenges

In addition to the treat vectors discussed in previous sub-sections, there are some security issues specific to SDN that are not inherently present in traditional networks. In this subsection, we highlight these security challenges and best practices to avoid them in SDN.

4.3.4.1 Programmability

SDN offers programmatic features to the clients who belong to different business entities and organizations. Traditional business entities follow a closed domain, administrative model. Thus the SDN business model makes it necessary to protect system integrity, open interfaces and 3rd party data across multiple administrative and business domains.

- **Traffic and Resource Isolation:** The business management and real-time control information of one application need to be fully isolated from other applications. Traffic and resource isolation across tenants must be ensured in the SDN environment. The SLA requirements and private addressing scheme induced dynamic interactions may create a need for more fine-grained isolation.

- **Trust between third-party applications and controller:** Authentication and authorization mechanisms should be enforced at the point of application registration to the controller in order to limit controller exposure.

4.3.4.2 Integration with Legacy Protocols

The advent of SDN solved some of the technical and process deficiencies in the legacy protocols. However retrofitting of the security capabilities into existing technologies, e.g., DNS, BGP may not be straightforward. It is critical to inspect compatibility of legacy protocols before incorporation into SDN.

4.3.4.3 Cross Domain Connection

SDN infrastructure allows connectivity across different physical servers, clusters and data centers. Each security domain can be under the control of one or many controllers. An appropriate mechanism for establishing a trust relationship between controllers should be present in SDN design. The trust framework should have the ability to prevent abuse and capability of establishing a secure channel.
### TABLE 4.4: OpenFlow protocol analysis breakdown.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Component</th>
<th>Sub-components/Scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch</td>
<td>Ports</td>
<td>• Physical Ports</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Logical Ports</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Reserved Ports</td>
</tr>
<tr>
<td></td>
<td>Tables</td>
<td>• Counters</td>
</tr>
<tr>
<td>OpenFlow Channel and Control Channel</td>
<td>Channel Connections</td>
<td>• Connection Setup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Encryption</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Multiple Controllers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Auxiliary Connections</td>
</tr>
</tbody>
</table>

4.3.5 OpenFlow Protocol and OpenFlow Switch Security Analysis

4.3.5.1 Attack Model

**Actors:** The threats against OpenFlow protocol can be internally initiated or externally initiated. A trusted insider can try privilege escalation to modify the implementation of OpenFlow protocol or perform unauthorized access request on the OpenFlow related reference data. On the other hand, an external attacker can control the dataplane devices directly attached to the OpenFlow switches, and try to generate malicious traffic request aimed at disrupting the communication or gaining privileges to OpenFlow devices remotely.

**Attack Vectors:** The following attack vectors can be employed by external and internal attackers who aim to target OpenFlow components.

- Passive eavesdropping on the data/control plane messages. This may help the attacker gain necessary information for subsequent attacks.
- Replay attacks in SDN network with non-authentic data/control messages, Man-in-the-Middle (MITM) attack, DoS/DDoS attacks or side-channel attacks.
TABLE 4.5: OpenFlow component security issues and candidate countermeasures.

<table>
<thead>
<tr>
<th>Component</th>
<th>Security Issue</th>
<th>Candidate Countermeasure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Port</td>
<td>Fake physical port may be inserted or changed in order to perform traffic analysis leading to network attack.</td>
<td>Enable link state monitoring and network change tracking capability in SDN controller.</td>
</tr>
<tr>
<td>Logical Ports</td>
<td>Port tunnel ID missing in port statistic messages.</td>
<td>Enable tunnel ID checking feature in the controller.</td>
</tr>
<tr>
<td>Reserved Ports</td>
<td>Controller unable to collect statistical information on reserved ports.</td>
<td>Enable APIs to allow the controller to query reserved ports.</td>
</tr>
<tr>
<td>Counters</td>
<td>Counter roll-back out of control</td>
<td>Ensure controller, flow table synchronization.</td>
</tr>
<tr>
<td>Connection Setup</td>
<td>TLS protection for TCP header missing or information to manage Cryptographic Keys and certificates missing.</td>
<td>Mechanisms providing TCP-AO for header protection, and switch management protocol for key and certificate management should be present.</td>
</tr>
<tr>
<td>Encryption</td>
<td>Authentication for message communication not present.</td>
<td>Support for multiple types of authentication and encryption protocols should be incorporated.</td>
</tr>
<tr>
<td>Multiple Controller</td>
<td>Security policy conflict between controllers, malicious controller attempting unauthorized access in the network</td>
<td>Mutual authentication and synchronization should be employed across controllers. Role-based authentication for each controller. Secure communication between the master controller and switches.</td>
</tr>
<tr>
<td>Auxiliary Connections</td>
<td>Lack of verification mechanisms against invalid DPID.</td>
<td>Alert mechanism in the controller when invalid DPID sends across a packet. Use different authentication for auxiliary and main connections.</td>
</tr>
</tbody>
</table>
Target/Goal: The attacker may aim at obtaining OpenFlow protocol assets/properties:

- Sensitive information in protocol messages.
- Tenant, network topology, SDN network availability or performance related information.
- Reference data on devices implementing OpenFlow switch flow table entries.
- Data and resource information of control and dataplane (e.g., bandwidth, latency, flow timeout duration).

4.3.5.2 Protocol-Specific Analysis

The protocol specific analysis should consider following entities, components and sub-components as shown in table 4.4. We discuss the candidate security countermeasures to deal with attacks against OpenFlow protocol, OpenFlow switch and associated components highlighted in table 4.5. The analysis assumes that each OpenFlow switch can be connected to one or more controllers within the trust boundary of cloud service provider. Also, TLS security can be employed between switch and controller to deal with message tampering and to perform mutual authentication.

Summary

The SDN and NFV platforms suffer from many threat vectors, some of which are introduced by weak authentication, authorization mechanisms, others because of the SDN/NFV design. Consideration of each threat vector in isolation is important for creating a secured cloud networking environment managed by SDN/NFV. This chapter explored security issues affecting confidentiality, integrity, and availability if SDN/NFV. The security design goals and best practices, security countermeasures have described in detail for NFV, SDN data plane, SDN control plane and OpenFlow protocol. The secured architecture design depends on many other factors apart from the mechanisms described in this chapter, such as latency and throughput impact because of a particular secured configuration. These factors, however, are beyond the scope of this chapter and should be considered before adopting the recommendations provided as a part of this chapter.