SOFTWARE-DEFINED NETWORKING (SDN) USING NETWORK VIRTUALIZATION (NV) SECURITY TECHNICAL IMPLEMENTATION GUIDE (STIG) OVERVIEW

Version 1, Release 1

27 February 2017

Developed by DISA for the DoD

Trademark Information

Names, products, and services referenced within this document may be the trade names, trademarks, or service marks of their respective owners. References to commercial vendors and their products or services are provided strictly as a convenience to our users, and do not constitute or imply endorsement by DISA of any non-Federal entity, event, product, service, or enterprise.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

1. INTRODUCTION

1.1 Executive Summary

The Software-Defined Networking (SDN) using Network Virtualization (NV) Security Technical Implementation Guide (STIG) provides the technical security policies and requirements for the implementation of an SDN and NV architecture. Security must be integrated with every component within the SDN/NV infrastructure. Securing the infrastructure encompasses everything from access control of SDN controllers and orchestration systems to control plane messages between the SDN controllers and physical and virtual network nodes, a secured path between these components, and providing a secure platform to house the SDN components.

This STIG does not include security guidelines for the development of northbound or southbound application program interfaces (APIs) that may be integrated with SDN/NV architectures. Those requirements will be found in the appropriate application STIG or SRG. The SDN STIG does contain several configuration requirements for SDN-enabled physical network elements that provide the necessary plumbing for the implementation of the SDN/NV framework. All other security requirements for physical or virtual network elements will be provided via the Network Infrastructure and vendor STIGs.

1.2 Authority

DoD Instruction (DoDI) 8500.01 requires that “all IT that receives, processes, stores, displays, or transmits DoD information will be […] configured […] consistent with applicable DoD cybersecurity policies, standards, and architectures” and tasks that Defense Information Systems Agency (DISA) “develops and maintains control correlation identifiers (CCIs), security requirements guides (SRGs), security technical implementation guides (STIGs), and mobile code risk categories and usage guides that implement and are consistent with DoD cybersecurity policies, standards, architectures, security controls, and validation procedures, with the support of the NSA/CSS, using input from stakeholders, and using automation whenever possible.” This document is provided under the authority of DoDI 8500.01.

Although the use of the principles and guidelines in these SRGs/STIGs provides an environment that contributes to the security requirements of DoD systems, applicable NIST SP 800-53 cybersecurity controls need to be applied to all systems and architectures based on the Committee on National Security Systems (CNSS) Instruction (CNSSI) 1253.

1.3 Vulnerability Severity Category Code Definitions

Severity Category Codes (referred to as CAT) are a measure of vulnerabilities used to assess a facility or system security posture. Each security policy specified in this document is assigned a Severity Category Code of CAT I, II, or III.

Table 1-1: Vulnerability Severity Category Code Definitions

	DISA Category Code Guidelines
CAT I	Any vulnerability, the exploitation of which will **directly and immediately** result in loss of Confidentiality, Availability, or Integrity.
CAT II	Any vulnerability, the exploitation of which **has a potential** to result in loss of Confidentiality, Availability, or Integrity.
CAT III	Any vulnerability, the existence of which **degrades measures** to protect against loss of Confidentiality, Availability, or Integrity.

1.4 STIG Distribution

Parties within the DoD and Federal Government's computing environments can obtain the applicable STIG from the Information Assurance Support Environment (IASE) website. This site contains the latest copies of any STIGs, SRGs, and other related security information. The address for the IASE site is http://iase.disa.mil/.

1.5 Document Revisions

Comments or proposed revisions to this document should be sent via email to the following address: disa.stig_spt@mail.mil. DISA will coordinate all change requests with the relevant DoD organizations before inclusion in this document. Approved changes will be made in accordance with the DISA maintenance release schedule.

1.6 Other Considerations

DISA accepts no liability for the consequences of applying specific configuration settings made on the basis of the SRGs/STIGs. It must be noted that the configuration settings specified should be evaluated in a local, representative test environment before implementation in a production environment, especially within large user populations. The extensive variety of environments makes it impossible to test these configuration settings for all potential software configurations.

For some production environments, failure to test before implementation may lead to a loss of required functionality. Evaluating the risks and benefits to a system's particular circumstances and requirements is the system owner's responsibility. The evaluated risks resulting from not applying specified configuration settings must be approved by the responsible Authorizing Official. Furthermore, DISA implies no warranty that the application of all specified configurations will make a system 100 percent secure.

Security guidance is provided for the Department of Defense. While other agencies and organizations are free to use it, care must be given to ensure that all applicable security guidance is applied both at the device hardening level as well as the architectural level due to the fact that some of the settings may not be able to be configured in environments outside the DoD architecture.

1.7 Product Approval Disclaimer

The existence of a STIG does not equate to DoD approval for the procurement or use of a product.

STIGs provide configurable operational security guidance for products being used by the DoD. STIGs, along with vendor confidential documentation, also provide a basis for assessing compliance with Cybersecurity controls/control enhancements, which supports system Assessment and Authorization (A&A) under the DoD Risk Management Framework (RMF). DoD Authorizing Officials (AOs) may request available vendor confidential documentation for a product that has a STIG for product evaluation and RMF purposes from disa.stig_spt@mail.mil. This documentation is not published for general access to protect the vendor's proprietary information.

AOs have the purview to determine product use/approval IAW DoD policy and through RMF risk acceptance. Inputs into acquisition or pre-acquisition product selection include such processes as:

- National Information Assurance Partnership (NIAP) evaluation for National Security Systems (NSS) (http://www.niap-ccevs.org/) IAW CNSSP #11
- National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) (http://csrc.nist.gov/groups/STM/cmvp/) IAW Federal/DoD mandated standards
- DoD Unified Capabilities (UC) Approved Products List (APL) (http://www.disa.mil/network-services/ucco) IAW DoDI 8100.04

2. ASSESSMENT CONSIDERATIONS

2.1 Security Assessment Information

This STIG contains the security guidance for the implementation of the SDN/NV framework regardless of the specific software and hardware products or solutions used for its deployment and operation. Hence, it is imperative that STIGs specific for those products as well as any applicable Network Infrastructure STIGs are used to further reduce the risk of the network virtualization platform being compromised.

3. SDN NETWORK ARCHITECTURE AND CONCEPTS

3.1 SDN Overview

Software-Defined Networking is an emerging network paradigm where the control plane is decoupled from the data plane to improve network flexibility and manageability. The control plane makes decisions as to which way traffic is sent. The control plane function includes the system configuration, management, and exchange of routing table information. The data plane, also known as the forwarding plane, forwards traffic to the next node along the path to the selected destination according to control plane logic. With SDN, forwarding decisions that traditionally are computed by individual network elements will migrate to a controller that abstracts a logical view of the network. Network intelligence and state are now centrally maintained in an SDN controller or cluster of controllers. In some SDN implementations, forwarding decisions can migrate to the control plane and the centralized controller.

3.2 SDN Controller

An SDN controller is the central repository for control instructions, data flow logic, security policies, and business policies required to deploy, configure, and manage network elements to obtain the desired network behavior. The controller provides a programmatic interface for the provisioning of network services using a consistent approach.

The SDN controller uses an open or proprietary protocol to control each network element and the traffic flow on the network. SDN relies heavily on control messages between a controller and the forwarding devices for reliable network operation. Thus, it is critical to ensure network reachability between a controller and the forwarding devices within the SDN architecture.

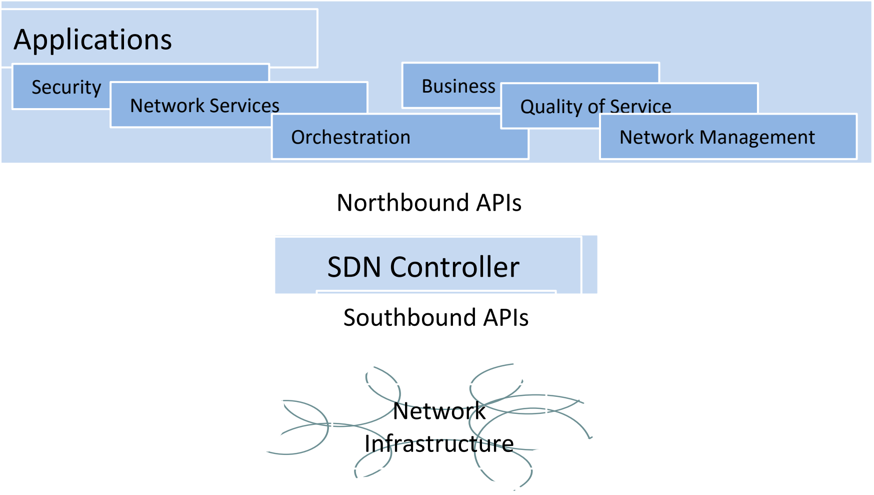
The controller also extracts state information about the network from the network elements and communicates that information back to the application with an abstract view of the network, including statistics and events about what is happening.

3.3 Application Program Interface (API)

SDN is a shift in networking that breaks traditional physical boundaries of network elements through well-defined APIs. An API is an interface presented by software that provides the capability to collect information from or make changes to an underlying resource. An API makes it possible to dynamically and programmatically provision and manage a network through software. The API is the control point for each component of the SDN-enabled network infrastructure, including switches, routers, SDN controllers, orchestration systems, network management systems, and network analytics. Virtualization enables existing physical network element constructs to be reused in a logical environment, while APIs enable resource abstraction.

APIs between the SDN controller and the application layer enable business applications to alter network behavior and deploy network services based on business requirements without the burden of network implementation details. As shown in Figure 3-1, the SDN architecture APIs, often referred to as northbound and southbound interfaces, provide the communication between the applications, controllers, and network elements.

Figure 3-1: APIs



3.3.1 Southbound API

Southbound APIs facilitate efficient management and control over the network and enable the SDN controller to dynamically make changes according to real-time demands and needs. Southbound API messages can be categorized as either control plane or management plane traffic. A southbound API implemented between the SDN controller and the network elements provides the mechanism to enable the controller to send forwarding table updates to the network elements such as switches, routers, and firewalls, both physical and virtual (hypervisor-based). These messages would all be considered control plane traffic, whereas management plane traffic consists of messages used by the controller to provision and configure network elements.

The API also provides the vehicle for the controller to receive new state information from the network elements. This can include the concept of flows to identify network traffic based on predefined match rules that can be statically or dynamically programmed by the SDN control software. The controller defines how traffic should flow through network elements based on policy, usage, applications, and available bandwidth.

3.3.2 Northbound API

Northbound API may be the most critical APIs within the SDN architecture since the value of SDN is tied to the innovation it can enable and support. Via communication with the SDN controller, the northbound API enables applications, management systems, and orchestration systems to program the network. A network abstraction (e.g., state information, configuration data, etc.) can be presented to applications and management systems via northbound API call to the controller. It enables developers to create network applications without the need to call the southbound API directly.

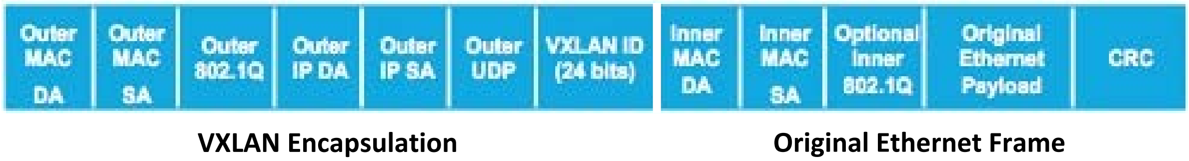
Northbound APIs are also used to integrate the SDN controller with automation stacks as well as orchestration platforms. The northbound APIs can be used to enable orchestration and automation of the network to align with the needs of different applications. Network services that can be deployed and optimized via this API include security services, load balancing, traffic engineering, and quality of service.

3.4 Overlay Transport Network

An overlay network can be defined as any logical network that is created on top of an existing physical network. Implementation of an overlay technology is enabled by encapsulating original packets with an outer header that defines the service, source, and destination. Communication is typically established between two tunnel endpoints. The overlay network provides the transport for communication between both physical and virtual workloads and integration between the physical and virtual infrastructures.

The Internet Engineering Task Force (IETF) standard Virtual Extensible LAN (VXLAN) is the predominant overlay technology to deploy network virtualization within an SDN framework. By encapsulating Ethernet frames with a routable IP header, VXLAN provides Layer 2 connectivity across an IP infrastructure, enabling virtual machines belonging to the same Layer 2 domain to communicate independent of hypervisor residency or location. Hence, VXLAN provides a scalable solution that enables data centers to provision many Layer 2 virtual networks in a multitenant environment.

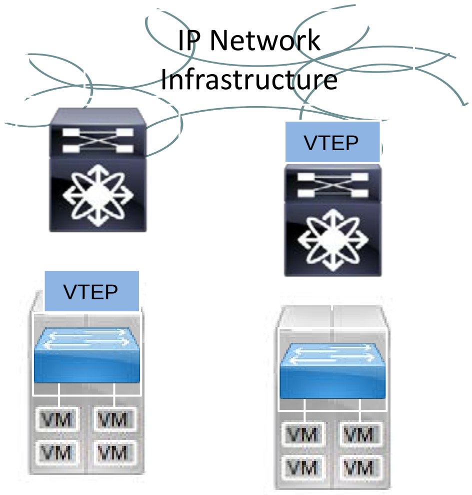
Figure 3-2: VXLAN Encapsulation



As shown in Figure 3-2, the VXLAN encapsulation includes a VXLAN ID to identify the VXLAN segment, source IP address of the sending VTEP, and destination IP address of the remote VTEP that has Layer 2 connectivity to the target virtual machine. The VXLAN ID is also referred to as the virtual network identifier (VNI), which essentially identifies a Layer 2 domain.

A VXLAN Tunnel Endpoint (VTEP) performs the frame encapsulation and de-encapsulation, as well as the forwarding and receiving of VTEP packets. As depicted in Figure 3-3, VTEP functionality can be implemented on hypervisor hosts or VXLAN-enabled switches. The latter is an enabler for Layer 2 integration with physical servers.

Figure 3-3: VXLAN Tunnel Endpoint



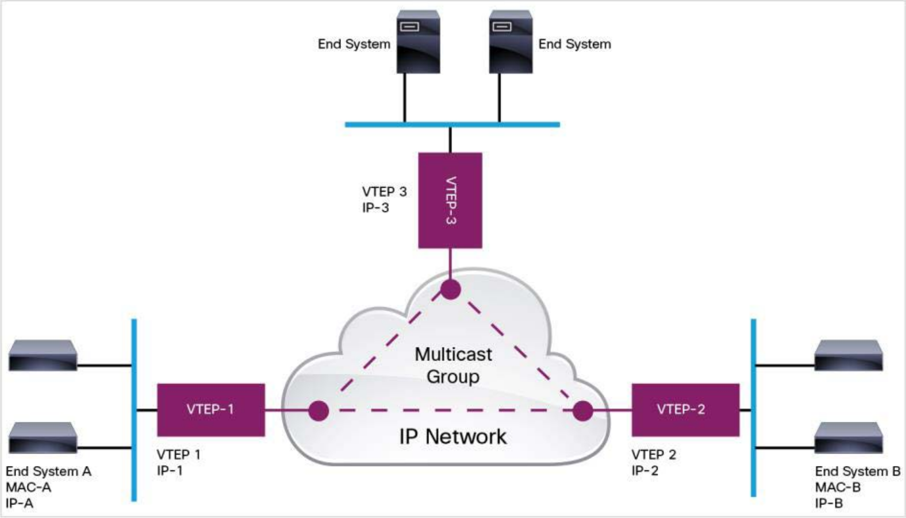
The VXLAN segments are independent of the underlying IP network topology; conversely, the underlying IP network between VTEPs is independent of the VXLAN overlay. VXLAN packets are forwarded based on the outer IP address header, with the initiating VTEP as the source IP address and the terminating VTEP as the destination IP address. VTEP-to-VTEP unicast reachability is provided through the routing protocol deployed on the IP network infrastructure.

Multicast is used by VTEPs to discover the IP address of remote VTEPs, as well to learn the IP and MAC addresses of end systems attached to those remote VTEPs. Multicast is also used to transmit VXLAN broadcast and unknown unicast packets, thereby limiting Layer 2 flooding to only those VTEPs that have end systems participating in the same VXLAN segment.

Each VXLAN segment (VNID) is mapped to an IP multicast group in the transport network. Each VTEP is independently configured to join this multicast group through the Internet Group Management Protocol (IGMP). The IGMP joins trigger Protocol Independent Multicast (PIM) joins through the transport network for the particular multicast group. The multicast distribution tree for this group is built through the transport network based on the locations of participating VTEPs. Figure 3-4 depicts an abstract of VTEPs that have joined a specific multicast group for the purpose of discovering each other and hence enable reachability for all physical and virtual servers belonging to the same Layer 2 domain. The end systems shown can be physical servers or virtual machines; henceforth, the VTEPs can be either hypervisor hosts or VXLAN-enabled switches.

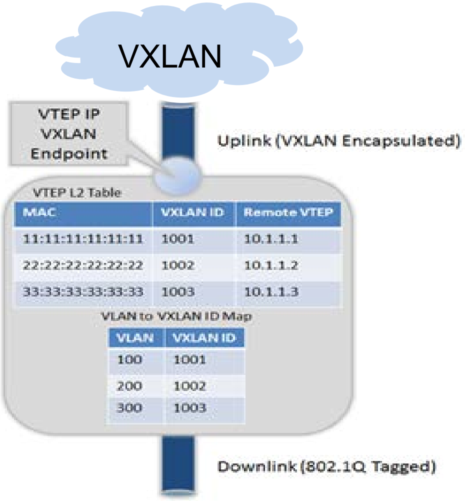
Note: Multicast is not the only method that can be implemented for the purpose of discovering remote VTEPs and MAC address learning.

Figure 3-4: VXLAN Multicast Group



A VTEP has two logical interfaces: an uplink and a downlink. The uplink is responsible for receiving and forwarding VXLAN packets and acts as the tunnel endpoint with an IP address. VTEP IP addresses are infrastructure addresses used for nodes within the VXLAN fabric; they are separate from any tenant IP addressing. Packets received from the VXLAN fabric on the uplink are mapped from the VXLAN ID to a VLAN ID. The payload is sent as an 802.1Q Ethernet frame on the downlink toward the virtual switch. During this process, the inner MAC source address and VXLAN ID is learned in a local table as depicted in Figure 3-5. Packets received from a virtual switch on the downlink are mapped to a VXLAN ID using the VLAN ID of the original frame.

Figure 3-5: VTEP Layer 2 Forwarding Table



VTEP L2 Table

MAC	VXLAN ID	Remote VTEP
11:11:11:11:11:11	1001	10.1.1.1
22:22:22:22:22:22	1002	10.1.1.2
33:33:33:33:33:33	1003	10.1.1.3

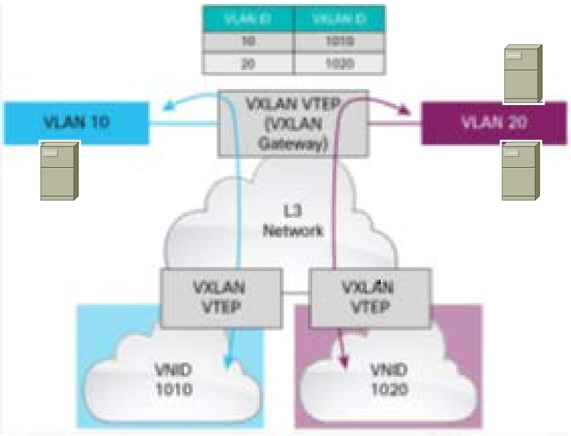
VLAN to VXLAN ID Map

VLAN	VXLAN ID
100	1001
200	1002
300	1003

When a frame is received on the downlink bound for an unknown destination, it is encapsulated using the assigned multicast group address as the outer destination IP address and is then sent out on the uplink. Any VTEP with nodes on that VXLAN ID will have joined the multicast group and therefore receive the frame. This maintains the traditional Ethernet flood and learn behavior required to build and maintain the Layer 2 forwarding table as shown in Figure 3-5.

As previously mentioned, VTEP functionality can be implemented on both hypervisor hosts and VXLAN-enabled switches. A VXLAN-enabled switch can also function as a VXLAN Layer 2 gateway, as well as a VXLAN Layer 3 gateway, to provide routing between different VXLAN segments. The VXLAN Layer 2 gateway bridges traffic between physical servers and virtual machines connected to virtual switches belonging to the same VNI. Traffic from the physical servers is mapped to the appropriate VNI based on VLAN membership, while traffic from virtual machines is encapsulated in VXLAN with VNI tagging. The logic mapping between IEEE 802.1Q VLAN and VXLAN on a VXLAN gateway is shown in Figure 3-6.

Figure 3-6: VXLAN Gateway



A typical data center will have physical hosts and service appliances coexisting with virtual machines. Virtual machines need to access services on physical hosts and appliances and vice versa, which creates the need of a gateway for virtual machines in a VXLAN segment to communicate with devices in a classic VLAN segment.

4. GENERAL SECURITY REQUIREMENTS

Security must be built into every component within the SDN framework, including the hardening of the SDN controller and the physical hypervisor servers hosting the SDN elements, as well as the underlying network infrastructure that provides transport for data plane, control plane, and VXLAN traffic. These Information Assurance (IA) controls will be found in the applicable OS and Network Infrastructure STIGs. API must be developed and implemented in accordance with applicable application STIGs and SRGs.

This STIG focuses on providing security requirements for the traffic between the controller and the applications and between the controller and SDN-enabled network elements. Authenticating and encrypting this traffic is imperative to mitigate any risk of the controller being compromised or the injection of a rogue controller. High availability of the controller is paramount to ensure service availability and the provisioning of intelligent routing. Guidelines are also provided for a secured implementation of the overlay transport network—specifically, configurations for the VXLAN tunnel endpoints.