Managing Risk from Software Defined Networking Controllers

Executive summary

Software Defined Networking (SDN) is a networking paradigm that enables enterprises to employ a centralized network management server to command and control network devices and control access to applications. This server is referred to as an SDN Controller (SDNC). Unlike traditional networks that require administrators to log in to each device, SDN allows administrators to scale device configuration and maintenance by only logging in to the SDNC to make changes to many devices at once. Often with little or no additional human interaction, SDN enables dynamic changes to switching and routing functions based on changing conditions detected in the network environment. Additionally, SDNCs may support integration with other servers and applications in an enterprise environment, typically via application programming interfaces (APIs). This integration can allow the SDNC to be part of an enterprise's greater automation and orchestration effort.

The SDNC benefits enterprise network management due to its centralized nature, but it also brings risk and could become a high priority target for adversaries. The SDNC's attack surface includes its management interface, the API it uses to communicate with other devices, the SDNC device itself, and the endpoints and switches that the SDNC manages. Malicious cyber actors could compromise these attack surfaces to perform management functions as if they were legitimate administrators, find sensitive configuration or authentication data, trick network devices into following a rogue SDNC's commands, or misconfigure the SDNC or SDN environment.

Given the critical nature of the SDNC, it requires additional oversight to prevent both malicious activity as well as unintentional changes to the network. The purpose of this cybersecurity information sheet (CSI) is to describe mitigations for SDNC risks.

What is software defined networking?

Software Defined Networking (SDN) allows networks to have a centralized network management server, also called the SDN Controller (SDNC), to control the network's devices automatically. Administrators configure policies in the SDNC that align with enterprise network segmentation requirements. As a result, the SDNC pushes configurations to network devices based on those policies. An SDNC may also include the capability to decide how to switch and route traffic when switches and routers receive network traffic.

Traditionally, when an administrator wants to make changes to network devices like switches and routers, the administrator has to log in to the network devices one-by-one to perform the necessary configuration changes. With SDN, the administrator needs only to log in to the SDNC. The administrator can configure policies in the SDNC that describe how the network devices must be configured under various conditions. In this way, SDN fosters dynamic configuration of multiple network devices to segment traffic to applications based on conditions in accordance with the policies. [1] The SDNC can detect changes in the network environment and autonomously update the configurations of the switches and routers, including to alter the segmentation of the network to enforce Zero Trust principles.

Finally, SDNCs often expose application programming interfaces (APIs), which allow enterprises to write custom scripts with API calls tailored to the operation and functions of the environment. This ability for programmatic administration and configuration of the SDNC and SDN environment further helps enterprises achieve their greater automation and orchestration efforts. APIs may also allow the SDNC to integrate with other existing services and applications, so that changes in the SDN environment can automatically be communicated with other critical components of the enterprise, and other applications can report external changes that could affect the network infrastructure to the SDNC, including cases where the SDNC should take action in response to security threats.

Control access to the management interface

SDNCs typically have a management interface for administrators to access and configure the SDNC and SDN environment. Unrestricted access to the management interface leaves the SDNC vulnerable to compromise.

NSA recommends enterprises only use SDNCs that support strict, fine-grained access control to the management interface. Enterprises should allow access to only a limited number of administrators in accordance with the principle of least privilege, using role-based access control (RBAC) to only allow SDN administrators to perform functions they are authorized to perform.

Ideally, enterprises should restrict access to the management interface to local access only, where an administrator must use a Privileged Access Workstation (PAW) connected directly to the management interface. If networked access must be used for remote administration, restrict access to a dedicated management network. [2] Only allow access to the SDNC's management interface from a dedicated PAW that only administrators use and is not used for high-risk activity (email, web surfing, etc.).

NSA strongly recommends multifactor authentication (MFA) for administrators managing critical devices. Using single factor authentication increases the risk of device exploitation. If unable to implement MFA, use unique local accounts assigned on a per user basis, along with long and complex passwords to reduce the risk of using single factor authentication. Choosing strong password storage algorithms can also help reduce the risk of using single factor authentication.

NSA recommends following National Institute of Standards and Technology (NIST) password guidance in NIST Special Publication 800-63B version 3, section 5.1.1 Memorized Secrets. [3]

Finally, NSA recommends enterprises physically segment the management interfaces from the interfaces used for communication with network devices. Connect PAWs to a segmented, out-of-band management (OOBM) network — only manage the SDNC from PAWs on the OOBM network. [2] Carefully consider which connections to allow and create rulesets that follow a deny-by-default, permit-by-exception approach.

Prevent viewing of sensitive information in network traffic

The SDNC typically communicates using the following two separate network traffic flows:

1. Network traffic for managing the SDNC
2. Network traffic for configuring network devices

Network traffic for managing the SDNC occurs between the PAWs and SDNC. This traffic can also occur between the SDNC and external services, such as services used for remote authentication and remote logging.

Network traffic for configuring network devices occurs when the SDNC needs to make changes on network devices. Here, the SDNC sends network traffic containing configuration information to the network devices.

These two network traffic flows can contain authentication and configuration information that could be vulnerable to man-in-the-middle techniques or passive viewing if the traffic or information is plaintext.

NSA recommends enterprises use SDNCs that support network traffic protocols that protect the authentication and configuration information transmitted over the network. For the management traffic flow, network traffic between administrator workstations and the SDNC should use strong encryption, such as transport layer security (TLS) version 1.2 or better and secure shell (SSH) version 2 or better. If remote authentication is used, ensure the network traffic to and from the remote services is encrypted, and ensure any authentication information transmitted over the network, including passwords, tokens, hashes, tickets, and challenge-responses, is protected from inspection. Traffic containing network device configurations transiting between the SDNC and network devices should also be encrypted. [4]

Some traffic in the SDN environment may not be able to be encrypted; NSA views having a centralized network management system that does not support encrypted protocols as doing more harm than good to the enterprise environment, and urges organizations to only use SDNCs that support encrypted protocols.

Protect critical data at rest

The SDNC must store sensitive authentication and cryptographic information, such as administrator credentials and encryption keys. Rather than try to intercept network traffic or leverage vulnerabilities to break into network devices, malicious actors could recover the passwords and keys stored on SDNCs that do not properly protect this sensitive information when stored at rest.

NSA recommends enterprises only use SDNCs that protect critical data at rest inside the controller: protect authentication credentials using strong hashing and encryption and limit access to stored cryptographic materials. [4] If possible, SDNCs should ensure data at rest inside the controller is restricted to internal services and processes needed to perform functions that use the data, and prevent unnecessary internal services and processes from accessing it at rest.

Only allow the authorized SDN controller to configure network devices

The network devices in the SDN environment only switch and route traffic in accordance with the configurations sent from the SDNC. If the SDNC-to-network device communication does not use protocols that are strongly authenticated, network devices awaiting configurations from the SDNCs may be vulnerable to receiving configurations from unauthorized sources claiming to be SDNCs.

Ensure that the network devices in the SDN environment only accept configurations from the legitimate SDNCs by only using strong cryptographic mechanisms for authentication, such as TLS certificates or SSH keys, to verify the SDNC is legitimate. [4] All attempts from other endpoints or malicious SDNCs to take control of network devices should be blocked and logged by the network devices.

Do not let unauthorized or unauthenticated devices join the SDN environment

Malicious actors could connect workstations or additional network devices to the SDN environment. The malicious workstations could craft packets to exploit vulnerabilities in the SDNC or network devices. The additional network devices could send fake data to the SDNC to cause it to make changes to the network. The actors could also compromise workstations already on the network to exploit devices or move laterally.

Configure the network to block or ignore unknown devices. Do not allow “autoconnection.” Block and log all unauthorized traffic to the SDNC and network device management interfaces.

Control access to APIs

SDNCs typically provide APIs to programmatically make changes through the controller and allow for interoperability with other software. While APIs are essential for automation and orchestration of networks, administrators should be aware of the additional attack surface that APIs create. SDNCs could be left vulnerable to API misuse, where malicious cyber actors could collect sensitive data or perform malicious actions. In addition, an API function making one legitimate configuration change could also disrupt, override, or cause conflicts with existing configurations or other SDN processes. This allows malicious actors and authorized users alike to cause negative impacts, such as disabling of security controls, misrouting of network traffic, opening previously closed segments, and so on.

Note: Configuring routing equipment can often be complex and issues can be difficult to diagnose when network configurations are static. This complexity can often be exacerbated by adding dynamically changing SDN policies, especially if they are not very carefully written.

Enforce separation of duties and least privilege when providing access to APIs. Create a dedicated API administrator role, and restrict privileges so this role does not have the same level of access as SDN administrators.

An API interface provides access to configurable items and functions commonly referred to as API “objects” or “endpoints.” The request that an administrator makes to perform an action on an API object is commonly referred to as an API “call.” The SDNC should only accept API calls from authorized API administrators. Restrict API calls to only those API objects relevant to the administrator’s role. Only API objects that can be used to safely and securely manage the SDN environment should be used: allowlist the API objects needed to make authorized changes, and block access to all other unsafe and irrelevant API objects.

NSA recommends enterprises enforce auditing and logging of all authorized and unauthorized calls to SDNC API objects. Creating and enforcing policy that determines how administrators are authorized to manage SDN environments through APIs will aid in identifying unusual activity. Feed API usage logs through analytic platforms to aid in alerting on unusual API usage.

NSA recommends enterprises only employ an SDNC that strongly authenticates API calls. Ensure API calls are secured within authenticated and encrypted network traffic flows, such as via TLS v1.2 or better, or SSH v2 or better. If possible, use mutual authentication, such as requiring both client and server certificates via TLS. At a minimum, require that the SDNC presents a valid server certificate to the client performing API calls, and that the client properly validates the certificate before authenticating and performing any API calls.

Manage custom code use

APIs in SDNCs allow users to write custom code and scripts that perform calls to the API objects and automate repetitive SDNC configuration and management tasks. While developers can read the API documentation to know which API objects will be needed to automate the SDN configuration tasks without in-depth knowledge of the existing networking, security, and SDN infrastructure, enterprises must still put custom code through version control and vetting. Some API objects may have dependencies on other API objects. Calling an API object to make one configuration change could also trigger another object to make an unintended change. These unintended changes could make networking and security changes the administrators did not intend, potentially resulting in negative impacts to the environment, such as inadvertently subverting security controls and misrouting network traffic to unprotected networks.

Only allow security-aware developers trained on how the SDN environment works and its risks to send custom scripts to the SDNC. All scripts used to make API calls should be validated through change management platforms.

Select reputable vendors

Since more complex interfaces naturally tend to have more implementation bugs, some APIs may have vulnerabilities in their design which could allow malicious cyber actors to gain unauthenticated access, conduct privilege escalation, perform remote code execution, or otherwise bypass the security controls of the device. Given the complexity, power, and relative newness of such APIs, this could be a lucrative attack surface for adversaries, and various API vulnerabilities have already been disclosed and patched.

Enterprises should only deploy SDNCs from vendors with a reputation for API reliability and security, and regularly apply patches and software updates as they are released.

Follow latest API documentation

Sometimes, APIs could become outdated or deprecated, but still be accessible. Also, products could have APIs that are not documented, but still accessible. Deprecated and undocumented APIs could have vulnerabilities that go undetected by vendors, but are identified and leveraged by malicious actors.

When possible, identify the latest vendor-provided API documentation and ensure the API usage is conducted in accordance with the documentation. API calls that appear to not conform to the API documentation could be a sign of undocumented or deprecated APIs. Prevent access to any discovered undocumented or abnormal APIs to reduce the risk of unwanted functions or exploitation from unpatched vulnerabilities.

Centralized control: a trusted core

SDNCs enhance the network environment through dynamic command and control of network devices and access to applications, but they could be a high-priority target for malicious cyber actors. When done right, the SDNC becomes the core of the modern environment, trusted by network devices and other critical enterprise services. To this end, a fully integrated SDN implementation needs to be well thought out and securely implemented to provide network segmentation and the needed granular access control for advanced cybersecurity capabilities and enforcing Zero Trust principles. NSA will be partnering with the National Information Assurance Partnership (NIAP) to publish an SDN Controller Protection Profile that further defines the security requirements to harden SDNCs.▪

Works cited

[1] NSA. Segment Networks and Deploy Application-aware Defense. 2019. https://media.defense.gov/2019/Sep/09/2002180325/-1/-1/0/Segment%20Networks%20and%20Deploy%20Application%20Aware%20Defenses%20-%20Copy.pdf

[2] NSA. Performing Out of Band Network Management. 2020. https://media.defense.gov/2020/Sep/17/2002499616/-1/-1/0/PERFORMING_OUT_OF_BAND_NETWORK_MANAGEMENT20200911.PDF

[3] NIST. Special Publication 800-63B version 3, section 5.1.1 Memorized Secrets. 2020. https://pages.nist.gov/800-63-3/sp800-63b.html#sec5

[4] NSA. Network Infrastructure Security Guide. 2023. https://media.defense.gov/2022/Jun/15/2003018261/-1/-1/0/CTR_NSA_NETWORK_INFRASTRUCTURE_SECURITY_GUIDE_20220615.PDF

Disclaimer of endorsement

Purpose

This document was developed in furtherance of NSA's cybersecurity missions, including its responsibilities to identify and disseminate threats to National Security Systems, Department of Defense, and Defense Industrial Base information systems, and to develop and issue cybersecurity specifications and mitigations. This information may be shared broadly to reach all appropriate stakeholders.

Contact

Cybersecurity Report Feedback: CybersecurityReports@nsa.gov
General Cybersecurity Inquiries or Customer Requests: Cybersecurity_Requests@nsa.gov
Defense Industrial Base Inquiries and Cybersecurity Services: DIB_Defense@cyber.nsa.gov
Media Inquiries / Press Desk: NSA Media Relations: 443-634-0721, MediaRelations@nsa.gov