Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

# Implementation of Zero-Trust Security in Software-Defined Networks (SDN)

## An Organized Review of Architecture, Security Controls, Implementation Challenges and Future Directions.

**Samruddhi Bajirao Mandekar**
Research Student
SCMIRT College, Pune



(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

## Abstract

Software-Defined Networking (SDN) changes the way enterprise networks are designed and managed by separating the control plane from the data plane and enabling programmable, centrally coordinated network policies. At the same time, this architecture introduces security concerns around controller compromise, application interfaces, flow-rule manipulation, denial-of-service attacks, and lateral movement. Traditional perimeter-based security is not sufficient for environments that include cloud services, remote users, mobile devices, and distributed workloads. Zero-Trust Architecture (ZTA) addresses this challenge by removing implicit trust based on network location and requiring explicit authentication, authorization, and continuous evaluation of access to protected resources. This paper reviews how Zero-Trust principles can be implemented in SDN environments. It examines the main architectural components, including identity and device verification, policy decision and enforcement, SDN controllers, secure controller-to-switch communication, micro-segmentation, telemetry, and automated policy updates. The study uses a secondary-source review of standards, government guidance, and recent academic literature. It does not claim to have deployed or experimentally evaluated a live enterprise system. The review finds that SDN can provide a flexible enforcement layer for Zero Trust because network paths and flow rules can be adjusted centrally in response to policy decisions. However, successful implementation depends on securing the controller and APIs, maintaining accurate identity and asset information, avoiding overly broad rules, preserving service availability, and planning a phased migration. The paper concludes with a practical implementation model, recommendations, limitations, and future research directions.

**Keywords:** Zero Trust Architecture, Software-Defined Networking, SDN Controller, Micro-segmentation, Identity-based Access Control, Policy Enforcement, Network Security, Continuous Authentication.

## I. Introduction

Enterprise networking has changed considerably with the growth of cloud computing, remote work, virtualization, Internet of Things (IoT) devices, and distributed applications. Users and workloads no longer operate only inside a clearly defined corporate network. Employees may connect from home, branch offices, public networks, or mobile devices, while applications and data may be hosted in private data centres and public clouds. In this environment, the assumption that a device is trustworthy simply because it is connected to an internal network creates significant security risk.

Zero-Trust Architecture (ZTA) is based on the principle that no user, device, or application should receive implicit trust merely because of its network location or ownership. Access to a resource should be granted only after the subject and device have been authenticated, authorization policies have been evaluated, and relevant security information has been considered. Access may be restricted or revoked when risk conditions change [1].

Software-Defined Networking provides a programmable approach to network management. It separates the control plane, which makes forwarding decisions, from the data plane, which forwards packets. An SDN controller maintains a logical view of the network and communicates with switches through southbound interfaces, while network applications interact with the controller through northbound interfaces. This programmability can help enforce Zero-Trust policies dynamically, for example by placing devices into restricted segments or installing flow rules that permit only approved communication.

The combination of SDN and Zero Trust is therefore promising: identity and contextual policy determine whether access is allowed, while the SDN control layer can translate the decision into network-level enforcement. However, SDN itself introduces additional risks because a compromised controller or insecure API may affect a large portion of the network. This paper reviews the opportunities and challenges involved in implementing Zero Trust in SDN-based enterprise environments.

## II. Research Problem

Traditional network security commonly relies on perimeter firewalls, VPN access, and broad internal trust zones. Once a user or device passes the perimeter, it may have more network reachability than is necessary for its role. If an attacker steals valid credentials or compromises an endpoint, the attacker may attempt lateral movement to reach additional systems. Static access rules can also become difficult to maintain as users, devices, workloads, and network locations change.

SDN improves programmability and centralized network management, but the same centralization creates a high-value target. Threats may affect the application plane, control plane, data plane, and communication channels between them. Research



(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

identifies issues such as unauthorized applications, controller attacks, flow-rule manipulation, denial-of-service, insecure interfaces, and data leakage [2].

The research problem addressed in this paper is how Zero-Trust principles can be translated into practical SDN enforcement mechanisms without creating unacceptable complexity, latency, operational risk, or dependence on a single controller. The paper also considers the gap between high-level Zero-Trust principles and the detailed configuration, monitoring, and governance required in an enterprise network.

## III. Objectives

O1: To explain the fundamental concepts and components of Zero-Trust Architecture and Software-Defined Networking.

O2: To examine how identity verification, device posture, least-privilege access, and continuous evaluation can be applied in SDN environments.

O3: To review SDN security threats and identify how Zero-Trust controls can reduce unauthorized access and lateral movement.

O4: To develop a conceptual implementation framework linking policy decisions with SDN-based network enforcement.

O5: To identify implementation challenges relating to controller security, interoperability, performance, availability, and legacy infrastructure.

O6: To provide practical recommendations and future research directions for enterprise adoption.

## IV. Research Methodology

This study uses a desk-based secondary-source review method. It draws on the NIST Zero Trust Architecture publication, the Cybersecurity and Infrastructure Security Agency (CISA) Zero Trust Maturity Model, and academic research concerning SDN architecture, SDN security, and Zero-Trust-enabled network frameworks. Sources were selected for relevance to the research objectives, technical authority, and publication recency. The review gives preference to standards and peer-reviewed publications, while conference papers are used where they directly discuss SDN and Zero-Trust integration.

The material is organized thematically into Zero-Trust principles, SDN architecture, integration and enforcement mechanisms, implementation workflow, security threats, operational challenges, and future scope. The paper is conceptual and review-based: it does not report original packet captures, laboratory measurements, production deployment data, or a controlled experiment. Therefore, performance benefits are discussed as expected design outcomes or findings reported by cited sources, not as results independently measured by this study.

## V. Literature Review

### A. Zero-Trust Architecture

NIST Special Publication 800-207 describes Zero Trust as an evolving cybersecurity approach that shifts protection away from static network perimeters and focuses on users, assets, and resources. It states that trust should not be granted solely because of a device's physical or network location or because the asset is enterprise-owned. Authentication and authorization are performed before a session to an enterprise resource is established [1].

A common logical model includes a Policy Engine (PE), Policy Administrator (PA), and Policy Enforcement Point (PEP). The Policy Engine makes the access decision using enterprise policy and contextual information. The Policy Administrator carries out the decision by establishing or terminating the communication path. The Policy Enforcement Point controls access to the protected resource. Supporting information may come from identity systems, device management, security monitoring, and threat intelligence [1].

Zero Trust is not a single product or a one-time authentication step. It is an architecture and operating model that combines identity, device health, resource policy, monitoring, and governance. CISA's Zero Trust Maturity Model organizes capabilities around five pillars: identity, devices, networks, applications and workloads, and data. It also emphasizes cross-cutting capabilities such as visibility and analytics, automation and orchestration, and governance [3].


(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

## B. Software-Defined Networking Architecture

SDN separates network control logic from packet forwarding. Its architecture is commonly described through three layers. The application layer contains services such as security policy applications, monitoring, load balancing, and access control. The control layer contains the SDN controller, which maintains network state and programs forwarding devices. The infrastructure or data layer contains physical and virtual switches that forward packets according to installed flow rules. Northbound APIs connect applications to the controller, while southbound protocols such as OpenFlow can connect the controller to forwarding devices [2].

The controller's logical centralization offers a network-wide view and allows policies to be changed programmatically. This can support dynamic segmentation and rapid isolation of compromised endpoints. However, it also creates risks: a vulnerable controller, weak API authentication, malicious application, or incorrect flow-rule update may affect multiple network devices. A systematic literature review of SDN security identifies threats across the application, control, and data planes and highlights the need for layered protection [2].

## C. Integration of Zero Trust with SDN

The integration model connects a Zero-Trust policy decision process to the SDN controller. An identity provider or access-management service authenticates the user. An endpoint-management or security tool supplies device posture information. The policy engine evaluates identity, device status, resource sensitivity, location, and risk. If access is approved, the policy administrator instructs the SDN controller or enforcement gateway to apply the required network policy. The controller then installs narrowly scoped forwarding rules or assigns the endpoint to an appropriate segment.

Recent research has explored machine-learning-assisted Zero-Trust policy generation for SDN. The ZT-SDN work by Katsis and Bertino addresses the difficulty of generating suitable least-privilege rules when normal communication requirements are not fully known [4]. Other work has proposed combining SDN and Software-Defined Perimeter mechanisms for fine-grained access control and micro-segmentation [5]. These studies indicate active research interest, but their proposed approaches should not be treated as universal production standards.

## D. Security Threats in SDN Environments

SDN security concerns can be grouped by architectural layer. At the application layer, malicious or vulnerable applications may misuse controller APIs or request unauthorized changes. At the control layer, attackers may target the controller through denial-of-service, credential compromise, insecure management interfaces, or malicious control applications. At the data layer, threats include forged traffic, flow-table exhaustion, rule conflicts, packet manipulation, and unauthorized communication. Communication between the controller and switches must also be protected against interception and tampering [2].

Zero Trust can reduce the impact of these threats by applying strong identity checks to administrators and applications, limiting privileges, authenticating devices, encrypting management channels, and enforcing resource-specific access. It does not remove the need for conventional security controls such as secure configuration, patching, redundancy, firewalls, intrusion detection, backups, and incident response.

## E. Core Technologies and Controls

| Component | Role in Zero-Trust SDN | Example control |
|---|---|---|
| Identity provider | Verifies users and service identities before access. | MFA, SSO, certificates, role/group attributes. |
| Device posture | Checks whether endpoints meet security requirements. | EDR status, patch level, managed-device status. |
| Policy Engine | Evaluates requests and determines allow, deny, or restrict. | Least privilege, context and risk-based policy. |
| Policy Administrator | Coordinates session setup or termination after a decision. | Sends authorized policy instructions to enforcement. |


(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

| | | |
|---|---|---|
| SDN controller | Translates approved policy into network configuration. | Controller RBAC, API security, signed/validated changes. |
| Policy Enforcement Point | Enforces access at the network or resource boundary. | Switch ACLs, gateways, host agents, micro-segments. |
| Monitoring and analytics | Supplies events and context for ongoing evaluation. | SIEM, flow telemetry, alerts, anomaly detection. |

*Table 1. Main components required for Zero-Trust enforcement in an SDN environment.*

### F. Conceptual Architecture and Access Flow

A practical design can be represented as a sequence of identity verification, context evaluation, policy decision, and network enforcement. The SDN controller is not itself the source of trust; it acts as a programmable enforcement mechanism under the direction of an authorized policy service. The controller and its management interfaces must also be protected as critical infrastructure.

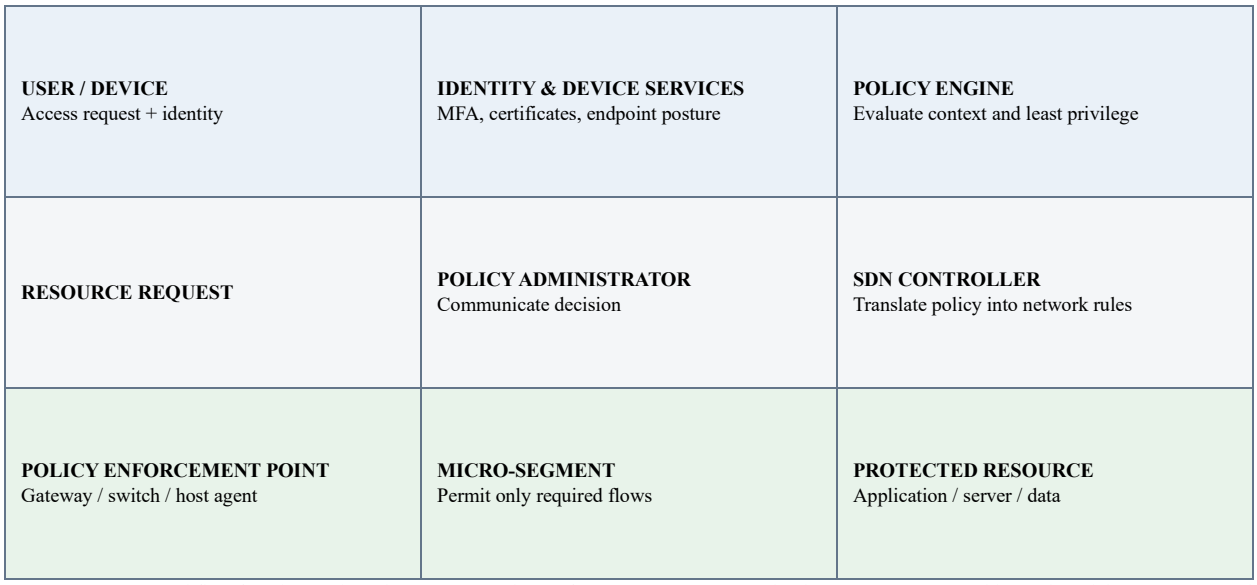



*Figure 1. Conceptual flow for identity-based Zero-Trust policy enforcement using SDN. The diagram is a proposed logical model, not a measured deployment.*

The sequence begins when a user or device requests access to an application. Identity and device services provide evidence to the policy engine. The policy engine evaluates the request and returns a decision. If approved, the policy administrator coordinates enforcement through the SDN controller. The controller updates the relevant network devices or enforcement points so that only the required traffic is permitted. Monitoring systems continue to supply telemetry, and access can be modified or revoked if the device becomes non-compliant or risk increases.

## VI. Discussion: A Practical Implementation Model

### A. Step 1 — Discover and Classify Assets

An organization should first identify users, endpoints, servers, applications, network devices, service accounts, and data flows. Each resource should be classified according to business importance and sensitivity. An accurate inventory is essential because access policies cannot be reliably designed when the organization does not know which assets exist or which systems communicate with each other. NIST recommends beginning migration by identifying enterprise actors, assets, key processes, and associated risks [1].

### B. Step 2 — Establish Identity and Device Verification

Integrate the network environment with a centralized identity provider and enforce multi-factor authentication for users, especially administrators and remote users. Devices should be identified using certificates or another strong mechanism



(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

where practical. Device posture checks can consider operating-system updates, endpoint detection status, disk encryption, and whether the device is managed. Unknown or non-compliant devices should be placed in a restricted network segment rather than receiving broad internal access.

### C. Step 3 — Define Least-Privilege Access Policies

Access policies should be based on the specific application or service required, not merely on a user's presence in an internal subnet. A policy may consider user role, device identity, resource sensitivity, time, connection context, and current risk. Default-deny rules should be used where operationally possible, with explicit exceptions documented and reviewed.
Policies should be tested against observed application dependencies to avoid interrupting legitimate business traffic.

### D. Step 4 — Integrate the Policy Engine with the SDN Controller

A controlled integration is required between the policy decision service and the SDN controller. The interface should use strong mutual authentication, encrypted transport, narrowly scoped API permissions, and audit logging. The controller should accept only authorized and validated policy updates. Administrative access to the controller should be separated from ordinary user access, and configuration changes should follow change-management procedures.

### E. Step 5 — Enforce Micro-segmentation

Micro-segmentation divides the network into small security zones and restricts communication between them. In an SDN environment, segmentation can be implemented through dynamically managed access-control lists, virtual network identifiers, security groups, or flow rules, depending on the platform. For example, a workstation may be allowed to reach a particular application server on a required port but denied access to unrelated server networks. A compromised endpoint can then be isolated by changing its policy or moving it into a quarantine segment.

### F. Step 6 — Monitor, Reassess, and Revoke

Zero Trust requires ongoing visibility. Network flow records, authentication events, endpoint alerts, controller logs, and application telemetry should be sent to a monitoring or SIEM platform. Policies should be reassessed when identity, device health, location, or threat conditions change. If a device is detected as compromised, the system may revoke its session, block specific flows, or move it to a restricted segment. Automated actions should include safeguards to reduce false positives and unintended service interruption.

### G. Step 7 — Pilot, Measure, and Expand

Implementation should begin with a limited pilot, such as a sensitive application or a small group of managed endpoints. The organization should measure policy accuracy, access-denial rates, authentication delays, controller load, service availability, and the time required to isolate a compromised endpoint. Lessons from the pilot can be used to refine policies before expanding to other business units, branches, data centres, and cloud environments.

### Table 2. Suggested phased implementation plan

| Phase | Main activity | Expected output |
|---|---|---|
| 1. Discover | Inventory assets, identities, applications and traffic flows. | Asset map and prioritized risk list. |
| 2. Prepare | Integrate identity, MFA, device posture and logging. | Verified identity and device signals. |
| 3. Design | Define resource-level policies and segmentation boundaries. | Documented least-privilege policy set. |
| 4. Integrate | Secure policy-to-controller API and validate changes. | Controlled policy enforcement path. |
| 5. Pilot | Apply rules to selected resources and monitor impact. | Pilot results and corrected policies. |
| 6. Expand | Scale to additional users, networks, workloads and sites. | Phased enterprise adoption. |
| 7. Improve | Review telemetry, incidents, exceptions and policy drift. | Continuous policy improvement. |


(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

## H. Security and Operational Considerations

The SDN controller should be deployed with high availability where required, secure administrative access, role-based access control, configuration backups, timely patching, and monitoring. Controller-to-switch communication should be protected using supported secure transport and certificate validation. API credentials should be stored securely and rotated. Organizations should also plan for controller failure, network partition, policy-service unavailability, and emergency break-glass access. The fail-open or fail-closed behavior must be selected according to the business impact and risk of each resource.

A further concern is policy consistency. A policy update may reach switches at different times, and conflicting or stale flow rules can create unexpected access paths. Change validation, version control, staged deployment, rollback procedures, and periodic rule audits are therefore important. Multi-controller deployments require coordination to avoid inconsistent policy state across network domains.

## I. SWOT-Style Assessment

| Strengths | Weaknesses |
|---|---|
| • Centralized and programmable enforcement.<br>• Fine-grained segmentation can reduce unnecessary reachability.<br>• Policies can respond dynamically to identity and device context.<br>• Better visibility when telemetry is integrated. | • Complex integration across identity, SDN, endpoint and SIEMsystems.<br>• Controller and API become high-value security assets.<br>• Accurate asset and traffic inventories are required.<br>• Poorly designed rules can disrupt applications. |
| **Opportunities** | **Threats** |
| • Support for hybrid cloud and remote access security.<br>• Automated isolation of compromised devices.<br>• Policy analytics and risk-based access decisions.<br>• Gradual migration from broad network zones. | • Controller compromise or denial-of-service.<br>• Stolen credentials and compromised endpoints.<br>• Misconfiguration, policy conflicts and vendor lock-in.<br>• Latency, outages and interoperability problems. |

*Table 3. SWOT-style assessment based on the literature reviewed in this paper.*

Overall, SDN can make network-level enforcement more adaptable, but it does not automatically make a network Zero Trust. The quality of the result depends on identity assurance, policy design, controller security, accurate telemetry, and operational governance. The most realistic approach is gradual adoption, starting with high-value applications and well-understood traffic flows.


(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

## VI.A. SDA Network Setup and Its Role in Zero-Trust Implementation

Software-Defined Access (SDA), commonly called Cisco SD-Access, is an enterprise campus networking approach that uses a fabric-based architecture to simplify network access, segmentation, and policy management. In a Zero-Trust design, SDA can provide a programmable network foundation for applying identity-based access policies and limiting communication between users, devices, and applications. The supplied diagram shows an example enterprise network layout and the surrounding WAN, security, wireless, and server infrastructure.

**Key points of the illustrated SDA setup:**

1. **WAN and Internet connectivity:** Primary and secondary Internet routers connect through redundant Internet firewalls. Primary and secondary DMPLS routers provide connectivity to the MPLS cloud and applications.
2. **Firewall protection:** Internet firewalls and server firewalls are shown in high-availability pairs. These security layers help control traffic entering or leaving the enterprise and protect server-side resources.
3. **Campus fabric and switching:** The SDA portion contains paired Fusion, Border, and Intermediate switches, shown as Cisco Nexus 9600 devices, connected redundantly. A Cisco C9300 Hub Room switch provides downstream connectivity to end devices and wireless infrastructure.
4. **Identity and access services:** Cisco ISE is shown as an identity/security service, while DMAC and the WLAN controller support network access and wireless management. Identity and device context can be used to assign appropriate access policies and segments.
5. **Location services:** The server segment includes AD, DNS, DHCP, and SCCM. These services support authentication and directory functions, name resolution, IP address allocation, and endpoint management.
6. **Zero-Trust relevance:** The architecture can support least-privilege access by using identity and device information to apply segmentation policies. Access should be limited to required resources, continuously monitored, and adjusted when device posture or risk changes. SDA alone does not guarantee Zero Trust; policy design, authentication, monitoring, and secure management remain necessary.

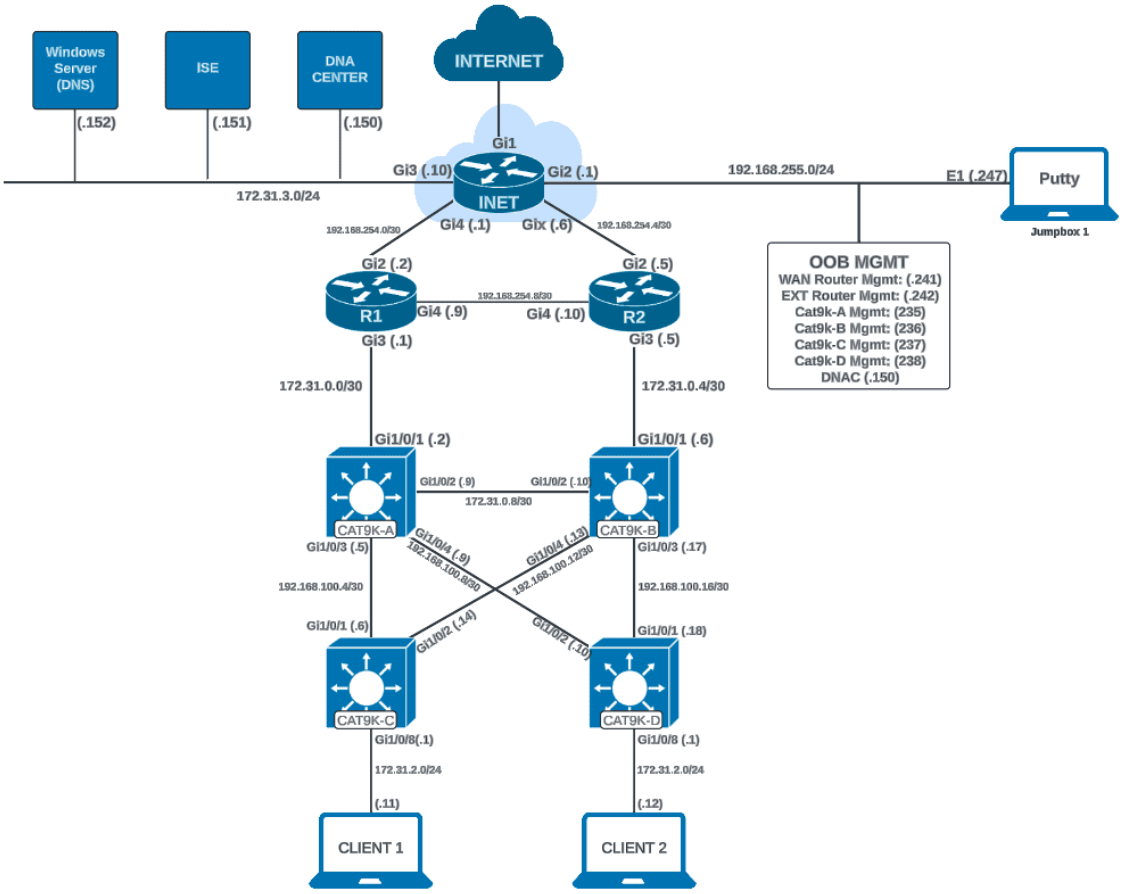



*Figure 2. Example SDA enterprise network setup supplied for inclusion in this research paper.*



(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

## VII. Limitations of the Study

This paper is based on secondary sources and a conceptual implementation model. It does not include a live SDN testbed, production network deployment, packet-level measurements, or interviews with enterprise network administrators. Consequently, it cannot quantify the effect of the proposed model on throughput, latency, controller CPU utilization, or incident response time.

Zero-Trust and SDN implementations vary across vendors, controller platforms, cloud environments, and enterprise architectures. The exact enforcement method may differ between physical switches, virtual switches, host agents, firewalls, and cloud-native policy systems. The paper therefore describes a vendor-neutral model rather than a configuration guide for a particular product.

Some recent research on SDN and Zero-Trust integration is still emerging and may present prototypes or proposed architectures rather than independently validated, large-scale deployments. Findings from such work should be interpreted cautiously. The review also focuses mainly on enterprise network environments and does not examine every specialized setting, such as operational technology, industrial control systems, or safety-critical networks in depth.

## VIII. Recommendations

1. Begin with asset and traffic discovery. Build and maintain an accurate inventory of users, devices, applications, servers,service accounts, and communication dependencies before writing enforcement rules.

2. Use strong identity verification. Enforce multi-factor authentication for users and administrators, use certificates orequivalent strong identity for devices and services, and remove shared or unnecessary accounts.

3. Apply least-privilege access. Permit only the specific application flows and services required for a business function.Review temporary exceptions and remove them when no longer needed.

4. Secure the SDN controller and APIs. Use role-based administration, mutual authentication, encrypted managementchannels, secure credential storage, audit logging, patching, and controlled change processes.

5. Introduce micro-segmentation gradually. Start with a limited application or network zone, validate dependencies,monitor denied traffic, and expand only after policies have been tested.

6. Integrate monitoring and response. Correlate identity events, device posture, network flow data, controller logs, andendpoint alerts. Define safe automated actions for quarantine and access revocation.

7. Plan for failure and recovery. Design controller redundancy where needed, back up configurations, test rollbackprocedures, and document emergency access methods. Define resource-specific fail-open or fail-closed behavior.

8. Measure operational impact. Track access decision latency, policy accuracy, false-positive isolation, service availability,controller health, rule consistency, and time to contain a compromised endpoint.

9. Train network and security teams together. SDN automation, identity systems, and Zero-Trust policy requirecollaboration between network operations, cybersecurity, cloud, endpoint, and application teams.

10. Maintain governance. Assign policy owners, review access rights regularly, document exceptions, and align theimplementation with organizational risk, privacy, and compliance requirements.

## IX. Conclusion

Zero-Trust Security and Software-Defined Networking address two important needs in modern enterprise environments. Zero Trust changes the basis of access decisions from implicit network location to verified identity, device condition, resource context, and least privilege. SDN provides a programmable network control layer that can translate those decisions into dynamic forwarding and segmentation policies. When integrated carefully, the two approaches can limit unnecessary communication, reduce opportunities for lateral movement, and support faster containment of compromised devices.


(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

However, implementation is not simply a matter of connecting a policy engine to an SDN controller. The controller, APIs, management interfaces, and flow-rule update process must themselves be secured. Organizations need reliable asset inventories, accurate traffic baselines, interoperable identity and monitoring systems, careful policy testing, and resilient operations. A phased deployment that begins with high-value resources and expands through measured pilots is more practical than attempting to redesign the entire network at once.

The overall conclusion is that SDN can serve as an effective enforcement foundation for Zero-Trust principles, but security outcomes depend on sound architecture, strong identity, least-privilege policy, continuous monitoring, and disciplined governance. Zero Trust should be treated as an ongoing improvement program rather than a one-time technology installation.

## X. Future Scope

Future work can implement the proposed architecture in a laboratory environment using an SDN controller, virtual switches, an identity provider, and a policy enforcement service. Experiments could compare conventional network access with identity-driven micro-segmentation under normal traffic and simulated attacks. Useful measurements would include access-decision latency, throughput, controller resource use, time to isolate an endpoint, and the number of unauthorized flows blocked.

Further research could investigate machine-learning techniques for discovering normal communication patterns and recommending least-privilege rules. Such systems would need explainable recommendations, human approval for high-impact changes, and safeguards against poisoned or misleading telemetry. Another direction is multi-controller Zero-Trust enforcement across hybrid cloud and geographically distributed enterprise networks, with a focus on policy consistency and resilience.

Additional study is needed on practical migration from legacy VLAN- and firewall-based networks, interoperability between vendors, management of IoT and unmanaged devices, privacy implications of continuous monitoring, and secure failover when identity or policy services are unavailable. Long-term field studies could help determine which implementation patterns provide measurable security improvement without creating unacceptable operational overhead.

## References

[1] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, “Zero Trust Architecture,” NIST Special Publication 800-207, National Institute ofStandards and Technology, Aug. 2020. https://doi.org/10.6028/NIST.SP.800-207

[2] M. S. Farooq, S. Riaz, and A. Alvi, “Security and Privacy Issues in Software-Defined Networking (SDN): A Systematic Literature Review,” *Electronics*, vol. 12, no. 14, article 3077, July 2023. https://doi.org/10.3390/electronics12143077

[3] Cybersecurity and Infrastructure Security Agency (CISA), “Zero Trust Maturity Model, Version 2.0,” Apr. 2023. https://www.cisa.gov/zero-trust-maturity-model

[4] C. Katsis and E. Bertino, “ZT-SDN: An ML-Powered Zero-Trust Architecture for Software-Defined Networks,” *ACM Transactions on Privacy and Security*, vol. 28, no. 2, article 23, 2025. https://doi.org/10.1145/3712262

[5] “A Software-Defined Zero Trust Framework for Secure Access Control and Microsegmentation using SDN and SDP,” in *Proceedings of the 2025 Fourth International Conference on Smart Technologies, Communication and Robotics (STCR)*, May 2025. https://doi.org/10.1109/STCR62650.2025.11020089

[6] J. Kim, M. Seo, S. Lee, J. Nam, V. Yegneswaran, P. Porras, G. Gu, and S. Shin, “Enhancing security in SDN: Systematizing attacks anddefenses from a penetration perspective,” *Computer Networks*, vol. 241, article 110203, Mar. 2024. https://doi.org/10.1016/j.comnet.2024.110203

[7] “Survey on security aspects of distributed software-defined networking controllers in an enterprise SD-WLAN,” *Digital Communications and Networks*, vol. 10, no. 6, pp. 1716–1731, Dec. 2024. https://doi.org/10.1016/j.dcan.2023.09.004

[8] National Institute of Standards and Technology, “Implementing a Zero Trust Architecture,” NIST NCCoE project documentation. https://www.nccoe.nist.gov/projects/implementing-zero-trust-architecture

[9] Cybersecurity and Infrastructure Security Agency (CISA), “Zero Trust Maturity Model,” official Zero Trust resources. https://www.cisa.gov/zero-trust

[10] Open Networking Foundation, “OpenFlow Switch Specification,” Open Networking Foundation. https://opennetworking.org/sdn-resources/openflow-switch-specification/

**Note:** This paper is a secondary-source review and conceptual implementation proposal. The architecture and phased plan are presented for academic discussion; no live enterprise deployment or original performance experiment is claimed.


(This work is licensed under a Creative Commons Attribution 4.0 International License.)