Introduction

Traditional network security models rely on perimeter-based defenses, assuming that internal users and devices are trustworthy. However, modern cyber threats, remote work, and cloud computing have made these models ineffective. Zero Trust Architecture (ZTA) enforces strict identity verification, least-privilege access, and continuous monitoring—making it a strong security approach for SDN-based networks.

Why SDN with Zero Trust?

- **Centralized Security Control:** SDN separates the control plane from the data plane, allowing dynamic security policy enforcement.
- **Microsegmentation:** SDN enables fine-grained access control, reducing the attack surface within a network.
- **Automated Threat Response:** Zero Trust policies in SDN can dynamically react to threats based on real-time traffic analysis.
- **Scalability & Cloud Readiness:** SDN is widely used in cloud and data center environments, making Zero Trust integration highly relevant.

Project Goals:

1. **Design a Zero Trust model** for SDN, focusing on authentication, microsegmentation, and policy enforcement.
2. **Develop and test an SDN-based security framework** using Mininet, Ryu Controller, and OpenFlow.
3. **Simulate cyberattacks** (e.g., port scanning, DDoS) and analyze how Zero Trust mitigates threats.
4. **Evaluate performance** in terms of security, network efficiency, and attack prevention.

Steps to move forward

Next Steps

Now that your setup is working:

- Start designing your **Zero Trust SDN framework**.
- Implement **authentication, microsegmentation, and real-time monitoring** in Ryu.
- Simulate attacks using **Nmap, hping3, and Iperf3**.
- Measure the network security performance.

1. **Creating a basic SDN network with Mininet**
2. **Implementing Zero Trust policies using the Ryu controller**
3. **Simulating attacks and testing security mechanisms**

`wireshark`: Packet sniffer.
`nmap`: Network scanner.
`hping3`: Advanced network testing tool.
`iperf3`: Network performance measurement.

`metasploit-framework`: Exploitation framework.

sudo ovsdb-tool create /etc/openvswitch/conf.db /usr/share/openvswitch/vswitch.ovsschema
sudo systemctl restart openvswitch-switch