learn..evolve

WIPRO NGA Program – Connectivity Datacomm Developer Batch

Capstone Project Presentation – 13th Aug 2024

Project Title - **C++ Monitoring and Compliance Checking**

Presented by - **Team C** : Arfath Basha(TL)
Anuj Jaiswal
Alpita Mandavkar
Kowshik Goli
Logesh Kumar
Murali Prasath
Praveen Rajamoorthy

Agenda

1. Project Overview
2. Monitoring Device Status
3. Compliance Checking
4. Monitoring Device Status and Compliance Checking implementation
5. Network Topology Management
6. Test Scenarios
7. Conclusion

Project Overview

- This project is a Monitoring and Compliance Checking System implemented in C++.
- The system is designed to help manage and oversee a network of devices, such as routers, switches, or other network devices.

Objective:

- We provides essential functionality to monitor the status of these devices (e.g., whether they are online or offline) and to verify that their configurations match predefined requirements.

Why this Monitoring and Compliance Checking ?

Implementing monitoring and compliance checking features is crucial for maintaining the health, security, and efficiency of IT systems and infrastructure.

- Ensuring Operational Continuity
- Efficient Management
- Enhancing Security
- Enhance network reliability by early detection of issues.
- Ensure security policies are followed through compliance checks.

Ping Method

Description:ICMP (Internet Control Message Protocol):

- It provides a quick and simple way to check if a device is reachable on the network

Command: ping -c 1 <IP_Address>

Advantages: Simple, widely supported

SNMP Method

SNMP (Simple Network Management Protocol):

- SNMP is a protocol used for network management, monitoring, and configuration.
- It allows for querying and managing network devices like routers, switches, and servers.

Advantages: Provides detailed device information

Limitations: Requires SNMP configuration on devices

What is Monitoring ?

- Monitoring refers to the continuous observation of IT systems, networks, or devices to ensure they are functioning correctly and to detect any issues or anomalies in real-time.

Key Aspects:

- **Device Status:** Tracking whether devices (servers, routers, etc.) are operational (up) or not (down).
- **Performance Metrics:** Measuring various performance indicators like CPU usage, memory usage, network bandwidth, etc.
- **Configuration Changes:** Observing and recording any changes to device settings or configurations.

Device Status Monitoring:

- The system can monitor the real-time status of network devices, reporting whether they are “UP” (online) or “DOWN” (offline).
- This is crucial for network administrators to ensure that all critical devices are operational.

What is Compliance Checking ?

- Compliance checking involves ensuring that systems and devices adhere to predefined standards, regulations, or policies
- **Key Aspects:**
  - **Configuration Standards:** Defining acceptable configuration settings and policies for devices.
  - **Automated Checks:** Using tools to automatically compare current configurations against these standards.
  - **Audit Trails:** Keeping records of compliance checks, including results and any actions taken.
  - **Reports:** Generating reports that detail compliance status, issues, and remediation efforts.
  - **Remediation:** Addressing and correcting any deviations from the compliance standards.

Compliance Checking:

- The system compares each device's configuration against an expected configuration.
- This helps in ensuring that all devices are correctly configured according to the network's standards, which is vital for maintaining network security and performance.

```
// Function to check compliance
bool checkCompliance(const std::unordered_map<std::string, std::string>& deviceData,
                     const std::unordered_map<std::string, std::string>& expectedConfig)
  {
    for (const auto& [key, value] : expectedConfig)
    {
        auto it = deviceData.find(key);
        if (it == deviceData.end() || it->second != value)
        {
            return false;
        }
    }
    return true;
  }
```

Device monitoring and Compliance Checking Implementation

```
// Devices to monitor
std::string devices[] = {"device1", "device2"};

for (const auto& device : devices)
{
    // Check if device is up
    bool isUp = isDeviceUp(device);

    if (isUp)
    {
        // Fetch data from device (mocked)
        auto deviceData = mockSnmpGet(device);

        // Check compliance
        bool isCompliant = false;
        if (device == "device1")
        {
            isCompliant = checkCompliance(deviceData, expectedConfig1);
        } else if (device == "device2") {
            isCompliant = checkCompliance(deviceData, expectedConfig2);
        }

        // Print results
        std::cout << "Device " << device << " is " << (isUp ? "Up" : "Down") << std::endl;
        if (isUp)
        {
            std::cout << "Compliance: " << (isCompliant ? "Compliant" : "Not Compliant") << std::endl;
        }
    } else
    {
        std::cout << "Device " << device << " is Down" << std::endl;
    }
}
```

Mock SNMP Function

```cpp
// Mock function to simulate SNMP data retrieval
std::unordered_map<std::string, std::string> mockSnmpGet(const std::string& device) {
    std::unordered_map<std::string, std::string> data;
    if (device == "device1") {
        data["hostname"] = "device1";
        data["interface"] = "f0/0";
        data["ip_address"] = "192.168.1.1";
    } else if (device == "device2") {
        data["hostname"] = "device2";
        data["interface"] = "f0/1";
        data["ip_address"] = "192.168.2.1";
    } else {
        data["hostname"] = "unknown";
    }

    return data;
}

// Mock function to simulate device reachability
bool isDeviceUp(const std::string& device) {
    // Simulate a basic check
    return device == "device1" || device == "device2"; // Simulate that both devices are up
}
```

Network Topology Management:

Network Topology:

- Network topology management is nothing but it involves the planning, design, deployment, and ongoing maintenance of the physical and logical structure of a network.

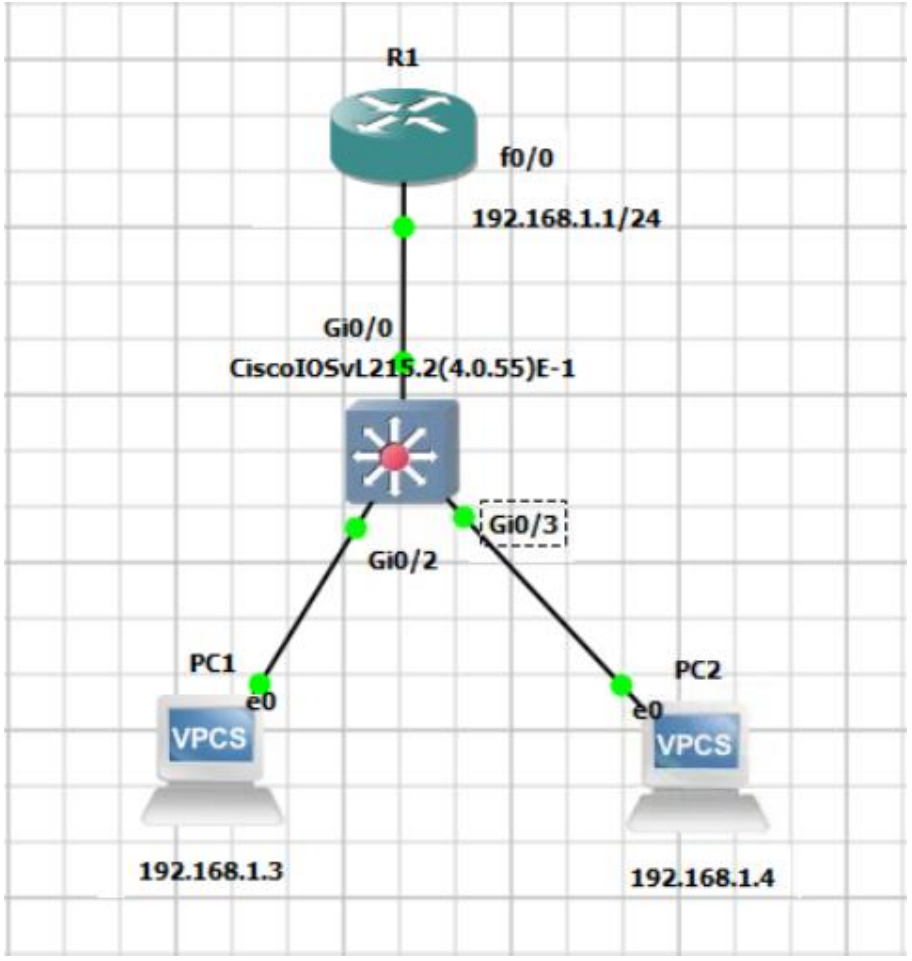
The Key aspects of Network topology management:

- Designing the topology
- Implementation
- Monitoring and maintenance

Testing Steps Overview

- Connect devices to the switch/router.
- Ensure correct IP assignments.
- Compile and run the code on a testing device.
- Ensure the output.

Topology



Sample Output

Test Scenario-1:

- If all devices are up and compliant

```
Device device1 is Up
Compliance: Compliant
Device device2 is Up
Compliance: Compliant
```

Test Scenario-2:

- If two devices are up and third device is down

```
Device device1 is Up
Compliance: Compliant
Device device2 is Up
Compliance: Compliant
Device device3 is Down
```

Troubleshooting

- No response from devices
- Incorrect IP address assignments
- Code not handling edge cases
- Verify device connections and configurations
- Check and debug code for errors
- Use network monitoring tools to diagnose issues

Benefits

- **Proactive Monitoring**: Continuous tracking ensures early detection of issues, reducing downtime.
- **Configuration Adherence**: Regular checks ensure devices comply with security policies and standards.
- **Performance Tracking**: Monitors operational metrics to help optimize device performance.
- **Detailed Logs**: Maintains comprehensive records of device statuses and compliance checks for auditing.

Conclusion

- Implementation of device status monitoring and compliance checking
- Testing scenarios to ensure functionality
- Additional considerations for robust deployment

Thank You