

**PROJECT REPORT**

**Members:**

1. **Syed Abdul Rehman 21K-3156**
2. **Muhammad Maaz 21K-4559**
3. **Xabad Nadeem 21K-4612**

**Instructor:**

**Prof. Engr Abdul Rehman**

**1. Introduction**

In this project, we implemented an **SSH configuration audit** on a Cisco router using **Python scripting**.  
The goal was to **remotely access** the router via SSH, **retrieve the running configuration**, and **verify the existence of SSH security parameters**.

We utilized **Paramiko**, a Python library for SSH communication, to connect and automate interaction with the router.

**2. Tools and Technologies**

| **Tool/Technology** | **Purpose** |
| --- | --- |
| Cisco Packet Tracer | Network simulation |
| Python 3.x | Programming language |
| Paramiko | SSH client library for Python |
| Packet Tracer PTR Server (optional) | For future REST API interaction |

**3. Objectives**

* To automate the process of connecting to a router over SSH.
* To retrieve and store the **running configuration** (show running-config).
* To check if the router has **SSH security** properly configured.
* To create a scalable base for auditing multiple network devices.

**4. Methodology**

**4.1 Network Setup**

* Created a simple topology in **Cisco Packet Tracer**.
* Configured a Cisco router with:
  + An IP address.
  + SSH enabled.
  + Local login credentials.

**4.2 Python Script**

* Built a Python script using **Paramiko** to:
  1. Establish an SSH session with the router.
  2. Enter **privileged EXEC mode** (enable).
  3. Execute the following commands:
     + terminal length 0 (to disable output paging)
     + show running-config
  4. Capture and save the output to a local file router\_config.txt.
* Basic error handling was added for:
  1. SSH authentication failures.
  2. Socket errors.

**4.3 SSH Audit Logic**

* Analyzed the retrieved configuration file.
* Searched for key SSH-related settings:
  + ip domain-name
  + crypto key generate rsa
  + line vty settings like transport input ssh

**5. Script Overview**

import paramiko

# Router credentials and IP

# (example values)

router\_ip = '192.168.10.1'

username = 'admin'

password = 'cisco'

port = 22

# Create SSH client

ssh = paramiko.SSHClient()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

try:

ssh.connect(router\_ip, port=port, username=username, password=password, look\_for\_keys=False, allow\_agent=False)

remote\_conn = ssh.invoke\_shell()

remote\_conn.send('enable\n')

remote\_conn.send(password + '\n')

remote\_conn.send('terminal length 0\n')

remote\_conn.send('show running-config\n')

import time

time.sleep(2)

output = remote\_conn.recv(10000).decode('utf-8')

with open('router\_config.txt', 'w') as f:

f.write(output)

ssh.close()

except Exception as e:

print(f"Connection failed: {e}")

**6. Results**

* Successfully connected to the router via SSH.
* Retrieved the full running configuration.
* Saved the configuration in a human-readable text file.
* Confirmed SSH setup by inspecting:
  + SSH domain name (ip domain-name)
  + RSA encryption keys (crypto key generate rsa)
  + SSH input transport (transport input ssh under vty lines).

**7. Challenges**

| **Challenge** | **Resolution** |
| --- | --- |
| Packet Tracer SSH emulation limitations | Worked around by ensuring router configurations are compatible |
| SSH encryption complexity | Used Paramiko for simple yet effective SSH session handling |
| Packet paging (more prompts) | Disabled with terminal length 0 command |

**8. Conclusion**

This project demonstrates how **Python automation** can efficiently audit network device configurations over SSH.  
Using libraries like **Paramiko**, we can automate security audits, backups, and compliance checks in real-world environments.  
This serves as a foundation for **larger network automation projects** involving multiple routers, switches, or even firewalls.

**9. Future Work**

* Implement multi-device scanning and reporting.
* Integrate audit results into a database.
* Extend the script to automatically fix missing SSH configurations.
* Explore using **Packet Tracer PTR API** for full network automation inside simulation environments.

**Audit Check List:**

**1. SSH Configuration Checks**

| **Checkpoint** | **Description** |
| --- | --- |
| ip domain-name is configured | Required for RSA key generation. |
| crypto key generate rsa is used | Enables SSH support via RSA keys. |
| SSH version is set to 2 (ip ssh version 2) | More secure than default version 1. |
| line vty transport input is ssh only | Prevents Telnet access (transport input ssh). |
| login local is used on line vty | Enforces login via locally defined users. |

**2. User Account & Privilege Audit**

| **Checkpoint** | **Description** |
| --- | --- |
| Local username/password configured | Avoids reliance on default or no accounts. |
| Passwords are encrypted (service password-encryption) | Prevents plain-text password exposure. |
| enable secret is used instead of enable password | Uses stronger encryption (MD5). |
| Minimum privilege level for users is appropriate | For example, only trusted users have level 15 (admin). |

**3. Interface & Network Access Controls**

| **Checkpoint** | **Description** |
| --- | --- |
| SSH is allowed only on internal interfaces | Avoid exposing SSH to external/public networks. |
| Access-class (ACL) on vty lines is used | Limits who can SSH in. Example: access-class 10 in. |
| Timeout settings (exec-timeout) are configured | Limits idle SSH sessions. Example: exec-timeout 5 0. |

**4. Logging & Monitoring**

| **Checkpoint** | **Description** |
| --- | --- |
| Logging is enabled (logging buffered, logging console) | Ensures SSH login attempts are logged. |
| Login banners (banner motd) warn unauthorized users | Required for legal notification. |
| SSH authentication failures are monitored | Shows evidence of brute force attempts. |

**5. Other Security Best Practices**

| **Checkpoint** | **Description** |
| --- | --- |
| Router hostname is set properly | Prevents confusion in larger networks. |
| No unused interfaces are left active | Shut down interfaces not in use. |
| No default credentials | Ensure defaults (e.g., cisco/cisco) are removed. |
| Clock is set or NTP is configured | Important for logging and troubleshooting. |

- SSH Version: ✓ Version 2 configured

- Transport Protocol: ✓ Only SSH enabled (Telnet disabled)

- User Authentication: ✓ Local login with encrypted passwords

- RSA Key: ✓ 1024-bit key generated

- Idle Timeout: ✗ Not configured (Recommendation: exec-timeout 5)

- VTY Access Control: ✗ No access-class applied (Add ACL to restrict SSH)