Secure Shell Protocol

Introduction

Secure Shell (SSH) protocol is a cryptographic network protocol for operating network services securely over an unsecured network. Its primary application is for remote login to computer systems by users. SSH provides a secure channel over an unsecured network by using a client-server architecture, connecting an SSH client application with an SSH server.

Aim

The aim of this document is to provide a comprehensive understanding of the SSH protocol, covering both practical usage and theoretical underpinnings. This includes the objectives, procedural steps, and concluding remarks on its importance and implementation.

Objectives

1. Understand the fundamental principles of SSH protocol.
2. Learn the theoretical concepts behind SSH security mechanisms.
3. Gain practical knowledge on how to use SSH for secure communication.
4. Understand the configuration and management of SSH servers and clients.
5. Appreciate the importance of SSH in network security and its application in various scenarios.

Theory

1. Principles of SSH Protocol

- Encryption: SSH uses strong encryption to protect the data being transmitted over the network. The most common algorithms include AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adleman).
- Authentication: SSH provides robust authentication mechanisms to verify the identities of the client and server. Methods include password-based, key-based, and more advanced mechanisms like two-factor authentication.
- Integrity: To ensure data integrity, SSH employs hashing algorithms like SHA-2 to verify that data has not been altered during transmission.

- Forward Secrecy: Ensures that session keys are not compromised even if the private key of the server or client is compromised in the future.

2. Components of SSH

- SSH Client: The software that initiates the connection.
- SSH Server: The software that accepts and manages the connection.
- SSH Daemon: The background process running on the server that handles incoming SSH connections.
- SSH Keys: Pairs of cryptographic keys used for securing communications and authenticating the parties involved.

3. SSH Key Management

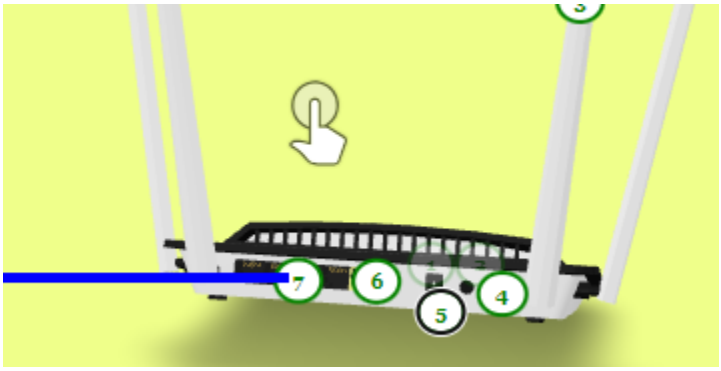
- Public/Private Key Pair: The public key is shared, while the private key is kept secret. They are used for establishing a secure connection.
- Key Generation: Tools like `ssh-keygen` are used to create key pairs.
- Key Storage: Keys are typically stored in `~/.ssh/` directory with permissions set to ensure security.

Conclusion

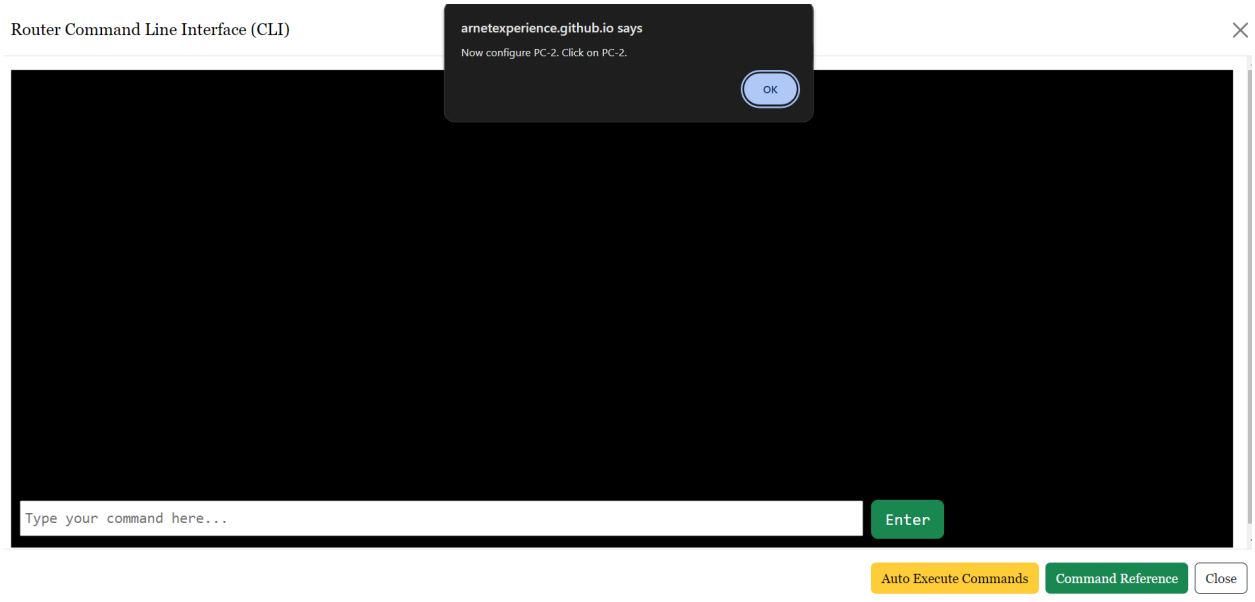
SSH is an essential protocol for secure communication over unsecured networks. Its robust encryption, authentication, and integrity mechanisms ensure the confidentiality and security of data. Practical knowledge of SSH, from key generation to configuring the SSH daemon, is crucial for network administrators and anyone involved in managing secure communications.

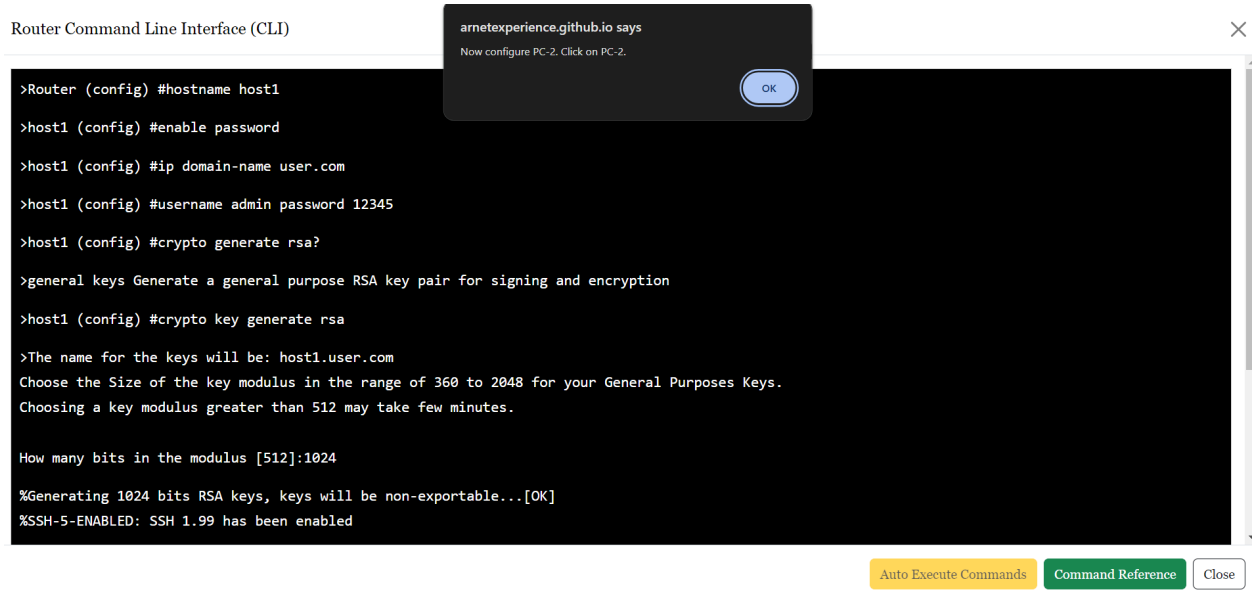
By understanding both the theoretical aspects and practical implementations of SSH, users can effectively secure remote connections and manage networked systems with confidence. The ongoing advancements and updates to SSH protocols and tools underscore the importance of staying informed and vigilant in maintaining network security.

1) Click on router's button number 6 and open CLI



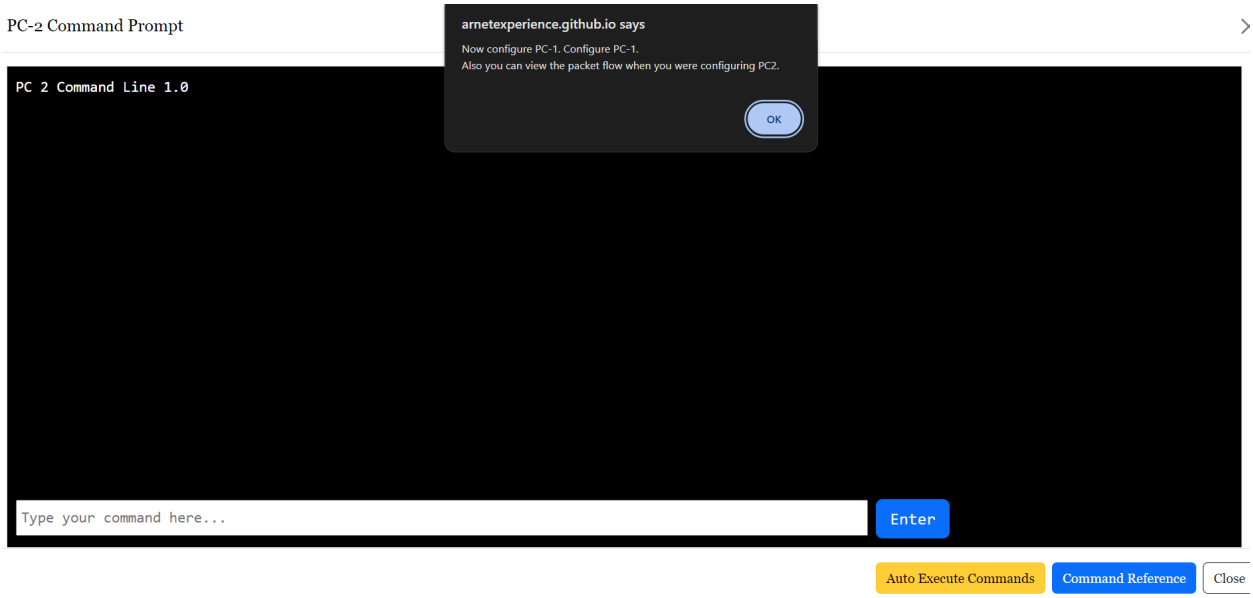
2) Click on auto execute command, you can also execute command one by one





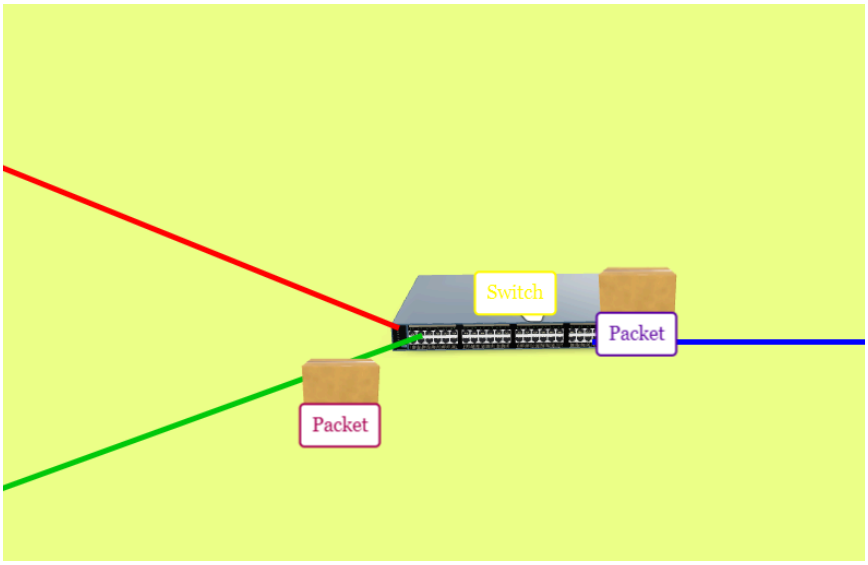
3) Now click on PC2, open command prompt

4) Auto execute command, you can also execute commands one by one

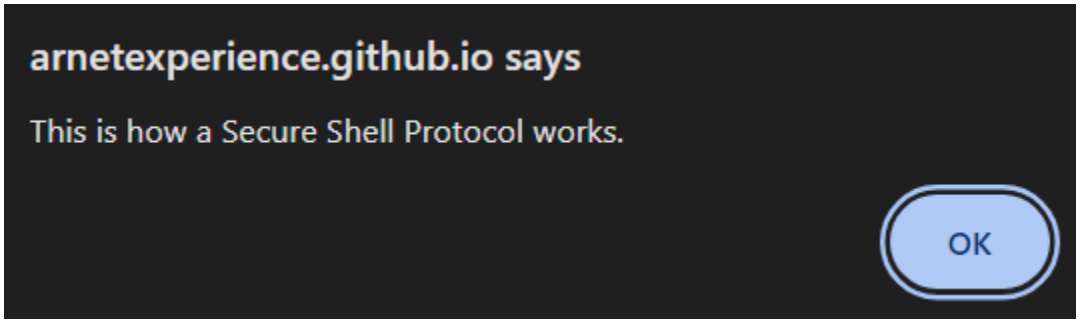



```
PC 2 Command Line 1.0
>ssh -l Admin 192.168.10.1

>Password:12345

host1>en

Password:12345

host1 #conf t

host1(config) #hostname PC2-SSH

PC2-SSH(config) #
```

5) Now packets will start flowing PC's to router and router to PC's



6) After the operation is successfully completed, you will get a pop-up message



Bonus!!! You can zoom in and zoom out the devices and also can read about each in detail.