Introduction to Cloud Computing Report

Lab 2: Nebula VPN Configuration and Connectivity

Name: Yin Yuang
Student ID: 202383930027
Class: Software Engineering Class 1

Introduction to Cloud Computing
(Autumn,2025)

Nanjing University of Information Science and Technology, China
Waterford Institute

I. Objective

The objective of this lab is to configure and verify a Nebula VPN, ensuring secure communication between multiple client nodes through a Lighthouse node. By completing this lab, students will achieve proficiency in:

- Configuring a Lighthouse node using a server provided by the instructor, setting its IP address, and generating the necessary certificates.
- Configuring multiple client nodes located behind NAT, enabling them to establish secure connections through the Lighthouse node.
- Ensuring secure node identity authentication through encryption certificates and configuration files.
- Testing network connectivity, including performing ICMP ping tests and verifying application layer connectivity via SSH.

II. Lab Content

1. Cryptographic Assets Acquisition
    - Obtain the necessary cryptographic assets provided by the instructor:
        - yinyuang.crt: Client certificate (public key) for identity verification.
        - yinyuang.key: Private key for decrypting data and signing outgoing messages.
        - ca.crt: Root certificate to establish the trust domain in the network.
2. Configuration and Deployment
    - Customize the configuration files using YAML format to define each node's role, IP address, and point to the cryptographic assets. The configuration should include the following:
        - 'pki.cert': Path to the client certificate (e.g., **'yinyuang.crt'**).
        - 'pki.key': Path to the private key (e.g., **'yinyuang.key'**).
        - 'pki.ca': Path to the root certificate (e.g., **'ca.crt'**).
        - 'static$_{h}ost_{m}ap$' : $PublicIPaddressoftheLighthousenode(providedbytheinstructor)$.
        - 'firewall': Rules to allow traffic such as ICMP/Ping and SSH.
    - Configure the Lighthouse Node:
        - Use the server provided by the instructor as the Lighthouse node, setting the IP address and generating necessary certificates.
        - Install Nebula and generate the certificates using the following commands:
            - 'nebula-cert ca –name "LeiNetwork"'
            - 'nebula-cert sign -name "lighthouse" -ip "192.168.100.1/24"'
            - 'nebula-cert sign -name "MBP15" -ip "192.168.100.15/24"'
            - 'nebula-cert sign -name "mini" -ip "192.168.100.10/24"'
        - Configure the 'config.yaml' file for the Lighthouse node, ensuring it points to the correct certificate paths and IP addresses.
        - Start the Nebula Daemon to enable the Lighthouse node to accept connections from client nodes:
            - 'sudo nebula -config /.nebula/config.yaml'
    - Configure Client Nodes:
        - Similar to the Lighthouse node, each client node needs to configure the certificate paths and the Lighthouse IP address in the 'config.yaml' file. The 'static$_{h}ost_{m}ap$'$shouldpointtotheLighthousenode'sIPaddre$ '192.168.100.1' $**$).
3. Connectivity Verification
    - ICMP Connectivity Test (Ping Test):
        - Use the 'ping' command to test if the client node can establish basic network connectivity to the Lighthouse node:

    - ‘ping 192.168.100.1‘
- Application Layer Connectivity Test (SSH Login):
  - Use SSH to confirm that the secure tunnel established by Nebula VPN can reliably pass application layer traffic, verifying node identity:
    - ‘ssh user@192.168.100.1‘

III. Results

1.Cryptographic Assets Received

The security and identity of the node rely on three essential files provided by the lecturer (Certificate Authority).

File Name	Role & Type	Core Function in Nebula	Security Requirement
yinyuang.crt	Client Certificate (Public Key)	Proves your identity to the network peers and Lighthouse. It contains the public key necessary for others to encrypt data for you.	Publicly shareable (but used internally for identity).
yinyuang.key	Private Key	Used to decrypt data sent to your node and to create digital signatures to authenticate your outgoing messages.	MUST be kept absolutely confidential.
ca.crt	Root CA Certificate	Establishes the network’s trust domain. It verifies that your ‘yourname.crt‘ (and all other peer certificates) were issued by the trusted CA.	Publicly shareable, needed by all nodes.

Table 1: Cryptographic Assets in Nebula VPN

The screenshot below displays the core files required for Nebula VPN deployment: - `nebula.exe` and `nebula-cert.exe`: Nebula’s main executable program and certificate management utility. - `ca.crt`, `yinyuang.crt`, `yinyuang.key`: Cryptographic assets (root certificate, node-specific certificate, and private key) that correspond to the `pki` section in Table 1. - `config.yaml`: The configuration file to be edited, which maps to the parameters outlined in the above table.

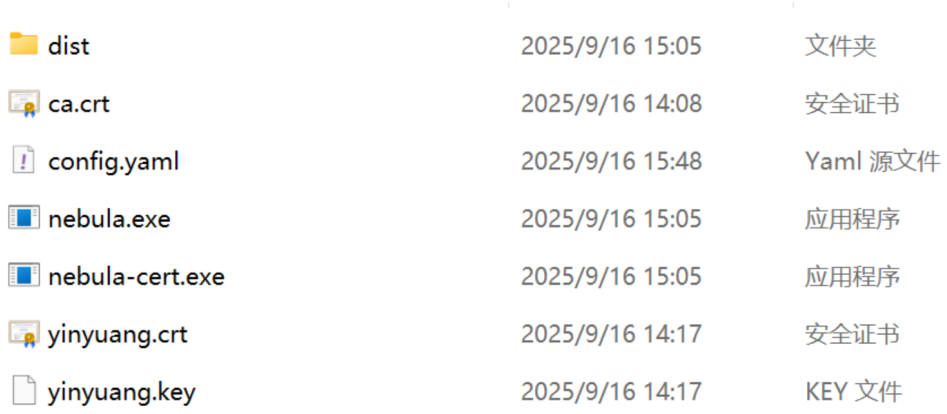

Figure 1: List of Nebula VPN Core Files

2.Configuration and Deployment

(1)Customizing the Configuration File

A YAML configuration file must be created or modified to define the node’s role, IP address, and point to the cryptographic assets.

Section	Parameter	Customization
pki	cert	Path to your `yourname.crt`
pki	key	Path to your `yinyuang.key`
pki	ca	Path to the root `ca.crt`
static_host_map	192.168.100.1	The Lighthouse's public IP/port for initial contact.
firewall	inbound/outbound	Custom rules to allow traffic (e.g., ICMP/Ping, SSH).

Table 2: Key Configuration Parameters for Nebula VPN

The screenshot below displays the core files required for Nebula VPN deployment: - `nebula.exe` and `nebula-cert.exe`: Nebula's main executable program and certificate management utility. - `ca.crt`, `yinyuang.crt`, `yinyuang.key`: Cryptographic assets (root certificate, node-specific certificate, and private key) that correspond to the `pki` section in Table 1. - `config.yaml`: The configuration file to be edited, which maps to the parameters outlined in the above table.

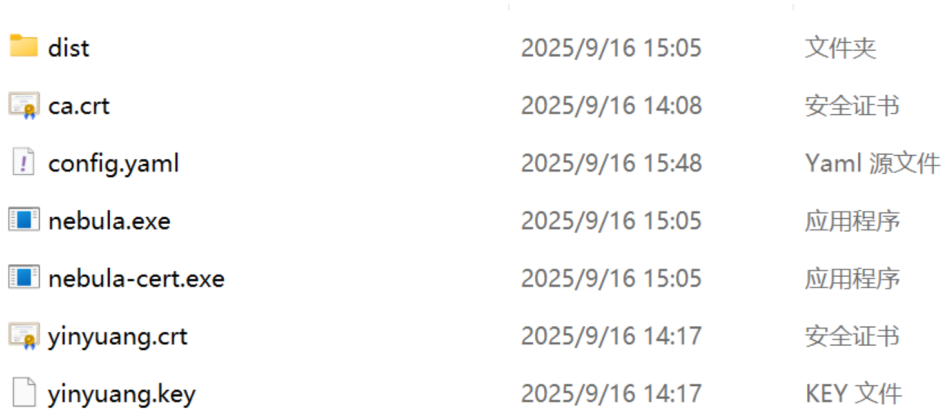

Figure 2: List of Nebula VPN Core Files

(2) Running the Nebula Daemon

The screenshot below presents the startup log of the Nebula VPN client on Windows systems, which records the full process: *Loading configuration → Starting firewall → Creating virtual network adapter → Listening on port → Connecting to the Lighthouse node*. Eventually, a handshake connection with the Lighthouse node (VPN IP: `192.168.100.1`, public address: `104.243.20.247:554`) is successfully established.

Figure 3: Nebula VPN Client Launch Log (Command-Line Output)

3.Connectivity Verification

(1) Testing ICMP Connectivity (Ping Test)

From a separate terminal window, the connectivity to the Lighthouse (IP 192.168.100.1) is tested using the ICMP protocol (Ping). ping 192.168.100.1

The screenshot below shows the result of the Ping test: it successfully receives 4 replies from the Lighthouse node (192.168.100.1) with 0

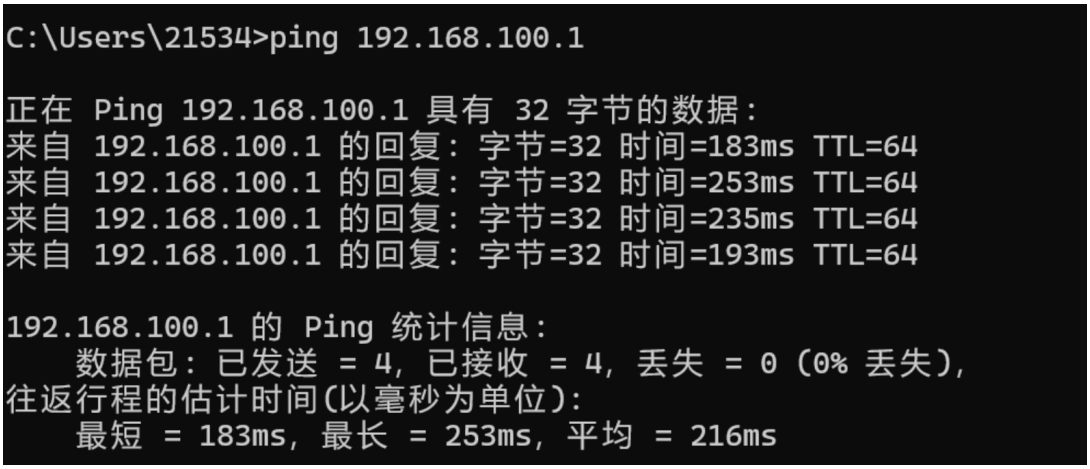

Figure 4: ICMP Connectivity Test (Ping Result to Lighthouse Node)

(2)Testing Application Layer Connectivity (SSH Login) To confirm that the secure tunnel can reliably pass application traffic and that the identity is fully verified, an SSH login attempt is made.

The screenshot below shows the successful SSH login result: after entering the password, the client connects to the Ubuntu 22.04 system of the Lighthouse node, and subsequent commands (e.g., `ls`, `pwd`) execute normally. This confirms that the application-layer connectivity of the Nebula VPN is fully functional.

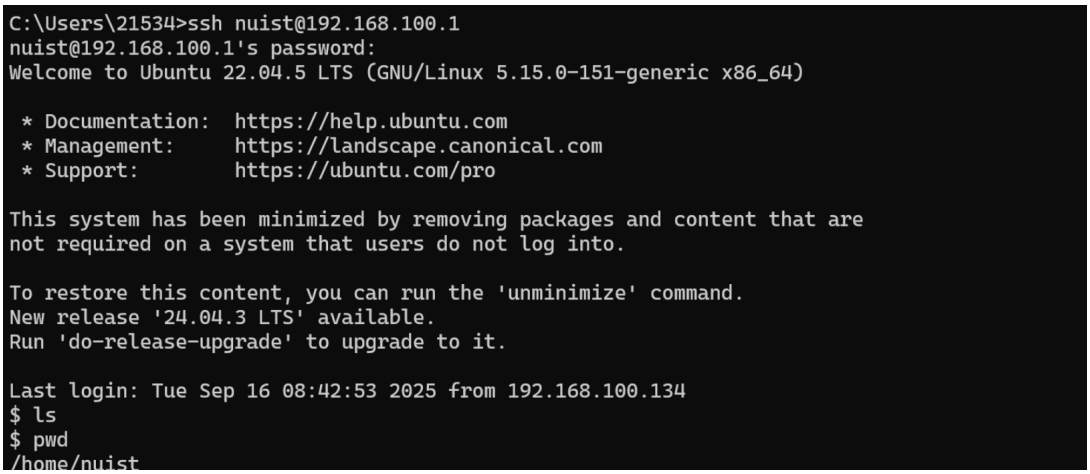

Figure 5: Application-Layer Connectivity Test (SSH Login to Lighthouse Node)