```
┌──[✗]─[user@parrot]─[~]
└─── $ssh -L 8080:remote-server-IP:80 studentuser@10.138.16.84
ssh: connect to host 10.138.16.84 port 22: Connection refused
┌──[✗]─[user@parrot]─[~]
└─── $
```

```
└─── $sudo adduser adminuser
Adding user `adminuser' ...
Adding new group `adminuser' (1002) ...
Adding new user `adminuser' (1002) with group `adminuser (1002)' ...
Creating home directory `/home/adminuser' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for adminuser
Enter the new value, or press ENTER for the default
        Full Name []: 1
        Room Number []: 1
        Work Phone []: 1
        Home Phone []: 1
        Other []: 1
Is the information correct? [Y/n] Y
Adding new user `adminuser' to supplemental / extra groups `users' ...
Adding user `adminuser' to group `users' ...
┌──[user@parrot]─[~]
└─── $
```

Advanced Network Architecture Implementation

1. VLAN Segmentation Implementation

Configuration Screenshots & Notes

Screenshot 1: VLAN Interface Configuration

- Created VLAN 10 (Admin) and VLAN 20 (Users)
- Configuration commands:
- bash

sudo apt install vlan
sudo modprobe 8021q
sudo vconfig add eth0 10
sudo vconfig add eth0 20
sudo ip addr add 192.168.10.1/24 dev eth0.10

- sudo ip addr add 192.168.20.1/24 dev eth0.20

Screenshot 2: VLAN Tagging Verification

- Verified with:
- bash
- sudo tcpdump -i eth0 -nn -e vlan

2. Zero Trust Implementation

Device Verification Before Access

Screenshot 3: Certificate-Based Authentication

- Required certificates for all devices
- Configuration:
- bash

```
# On firewall:
sudo apt install openssl
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/ztna.key -out /etc/ssl/certs/ztna.crt

# In sshd_config:
PasswordAuthentication no
```

- PubkeyAuthentication yes

3. Secure Routing Configuration

Screenshot 4: Route Table with ACLs

- Implemented route filtering:
- bash

```
sudo iptables -A FORWARD -i eth0.10 -o eth0.20 -j DROP
```

- sudo iptables -A FORWARD -i eth0.20 -o eth0.10 -m state --state NEW -j DROP

Screenshot 5: Logged Routing Attempts

- Logging commands:
- bash

sudo iptables -A INPUT -j LOG --log-prefix "NET_ACCESS: "

- sudo tail -f /var/log/syslog | grep NET_ACCESS

4. Security Tool Integration Workflow

Snort + Fail2Ban Integration

Screenshot 6: Snort Alert Generation

- Snort configuration:
- bash
- sudo snort -A console -q -c /etc/snort/snort.conf -i eth0

Screenshot 7: Fail2Ban Blocking

- Integrated with Snort via:
- bash

In jail.local:
[snort]
enabled = true
filter = snort

- logpath = /var/log/snort/alert

Implementation Notes

VLAN Configuration Details

- Used Linux VLAN packages
- Separate subnets for different trust zones
- Enabled VLAN tagging on virtual switch

Zero Trust Components

1. Device certificates required
2. User MFA enforced
3. Microsegmentation between VLANs

Routing Security

- Implemented antispoofing rules
- Logged all inter-VLAN traffic attempts
- Used stateful inspection firewall

Security Workflow

1. Snort detects intrusion attempt
2. Alerts sent to syslog
3. Fail2Ban parses logs and updates firewall rules
4. Offending IP gets blocked

Verification Tests

Screenshot 8: VLAN Connectivity Test

- Verification commands:
- bash

```
# From VLAN 10:
ping 192.168.20.1 # Should fail
```

- ping 192.168.10.1 # Should succeed

Screenshot 9: Zero Trust Access Denial

- Shows rejected connection attempt without proper certificate

Maintenance Procedures

1. **Daily Checks**:
2. bash

```
sudo vconfig list
sudo iptables -L -n -v
```

3. sudo tail -n 50 /var/log/snort/alert
4. **Certificate Renewal** (Monthly):
5. bash

```
6. sudo openssl x509 -in /etc/ssl/certs/ztna.crt -noout -dates
```

Configuration Files

Attached in submission:

1. `/etc/network/interfaces.d/vlan.conf`
2. `/etc/snort/snort.conf`
3. `/etc/fail2ban/jail.local`
4. `/etc/iptables/rules.v4`

This implementation demonstrates enterprise-grade network segmentation and Zero Trust principles in a lab environment, with integrated security monitoring and automated response.