

```
#!/bin/bash

# Function to check if the last command was successful
check_command() {
    if [ $? -ne 0 ]; then
        echo "Error: $1 failed to execute."
    fi
}

# Gather hostname information
echo "Hostname Information:"
hostname || check_command "hostname"
echo "-----"

# Gather network interfaces information
echo "Network Interfaces:"
ifconfig -a >/dev/null
if [ $? -ne 0 ]; then
    echo "Error: ifconfig command failed. Trying 'ip addr show' as an alternative."
    ip addr show || check_command "ip addr show"
else
    ifconfig -a
fi
echo "-----"

# Display routing table information
echo "Routing Table:"
netstat -rn >/dev/null
if [ $? -ne 0 ]; then
    echo "Error: netstat command failed. Trying 'ip route' as an alternative."
    ip route || check_command "ip route"
else
    netstat -rn
fi
echo "-----"

# Display current IP information
echo "IP Address Information:"
ip addr show || check_command "ip addr show"
echo "-----"
```

```

# Display current IP information
echo "IP Address Information:"
ip addr show || check_command "ip addr show"
echo "-----"

# Display DNS settings
echo "DNS Information:"
if [ -f /etc/resolv.conf ]; then
    cat /etc/resolv.conf || check_command "cat /etc/resolv.conf"
else
    echo "Error: /etc/resolv.conf file not found."
fi
echo "-----"

# Display default gateway
echo "Default Gateway:"
ip route | grep default &>/dev/null
if [ $? -ne 0 ]; then
    echo "Error: Unable to retrieve default gateway information."
else
    ip route | grep default
fi
echo "-----"

# Display open ports and listening services
echo "Open Ports and Listening Services:"
netstat -tuln &>/dev/null
if [ $? -ne 0 ]; then
    echo "Error: netstat command failed. Trying 'ss -tuln' as an alternative."
    ss -tuln || check_command "ss -tuln"
else
    netstat -tuln
fi
echo "-----"

# Display active network connections
echo "Active Network Connections:"
netstat -an &>/dev/null
if [ $? -ne 0 ]; then
    echo "Error: netstat command failed. Trying 'ss -an' as an alternative."
    ss -an || check_command "ss -an"
else

```

```

fi
echo "-----"

# Display active network connections
echo "Active Network Connections:"
netstat -an &>/dev/null
if [ $? -ne 0 ]; then
    echo "Error: netstat command failed. Trying 'ss -an' as an alternative."
    ss -an || check_command "ss -an"
else
    netstat -an
fi
echo "-----"

# Gather ARP table information
echo "ARP Table:"
arp -a &>/dev/null
if [ $? -ne 0 ]; then
    echo "Error: arp command failed. Trying 'ip neigh' as an alternative."
    ip neigh || check_command "ip neigh"
else
    arp -a
fi
echo "-----"
|

```

Explanation of the Error Handling

1. **check_command Function:** This function checks if the last command executed successfully (using \$?). If it didn't, it prints an error message.
2. **Alternative Commands:** If certain commands (ifconfig, netstat, arp) are unavailable, the script tries alternatives (ip addr show, ss, ip neigh). This ensures compatibility across different systems.
3. **File Existence Check:** For DNS information, it checks if /etc/resolv.conf exists before attempting to read it.
4. **Output Redirection:** &>/dev/null suppresses command output for error checking, avoiding unnecessary messages if the primary command fails.

This enhanced script provides more robust error handling and alternative commands, making it more adaptable across various Linux distributions.