#!/usr/bin/env python3

"""

Network Disabler - Disable network interfaces on Linux and Windows

Author: zakia-23

Repository: https://github.com/zakia-23/network-disable

License: MIT

"""

import subprocess

import sys

import os

import platform

def detect\_os():

"""Detect operating system"""

system = platform.system().lower()

if 'linux' in system:

return 'linux'

elif 'windows' in system:

return 'windows'

elif 'darwin' in system:

return 'macos'

else:

return 'unknown'

def check\_privileges():

"""Check if program has necessary privileges"""

os\_type = detect\_os()

if os\_type == 'linux':

if os.geteuid() != 0:

return False, "Root access required on Linux. Run with: sudo python3 network\_disabler.py"

elif os\_type == 'windows':

try:

subprocess.run(['net', 'session'], check=True, capture\_output=True)

except subprocess.CalledProcessError:

return False, "Administrator privileges required on Windows"

return True, "Privileges OK"

def disable\_network\_linux():

"""Disable network on Linux"""

try:

# Get active interfaces

result = subprocess.run(['ip', 'link', 'show'], capture\_output=True, text=True)

interfaces = []

for line in result.stdout.split('\n'):

if 'state UP' in line:

interface = line.split(':')[1].strip()

interfaces.append(interface)

if not interfaces:

print("No active interfaces found")

return True

print(f"Active interfaces: {interfaces}")

# Disable each interface (skip loopback)

for interface in interfaces:

if interface != 'lo':

print(f"Disabling interface: {interface}")

subprocess.run(['ip', 'link', 'set', 'dev', interface, 'down'], check=True)

return True

except Exception as e:

print(f"Error disabling network: {e}")

return False

def disable\_network\_windows():

"""Disable network on Windows"""

try:

# Get list of enabled network adapters

result = subprocess.run(

['wmic', 'nic', 'where', "NetEnabled=True", 'get', 'Name,Index'],

capture\_output=True, text=True, check=True

)

lines = result.stdout.strip().split('\n')

interfaces = []

for line in lines[1:]: # Skip header

if line.strip():

parts = line.strip().rsplit(' ', 1)

if len(parts) == 2:

name, index = parts

interfaces.append((index.strip(), name.strip()))

if not interfaces:

print("No active interfaces found")

return True

print("Active interfaces:")

for idx, (index, name) in enumerate(interfaces, 1):

print(f" {idx}. {name}")

# Disable each interface

for index, name in interfaces:

print(f"Disabling: {name}")

subprocess.run(

['wmic', 'path', 'win32\_networkadapter', 'where', f"Index={index}",

'call', 'disable'],

check=True, capture\_output=True

)

return True

except Exception as e:

print(f"Error disabling network: {e}")

return False

def show\_reactivation\_instructions():

"""Show how to re-enable network"""

os\_type = detect\_os()

print("\n" + "="\*50)

print("NETWORK REACTIVATION INSTRUCTIONS")

print("="\*50)

if os\_type == 'linux':

print("To re-enable network, run:")

print(" sudo ip link set dev [interface] up")

print("\nTo see available interfaces:")

print(" ip link show")

elif os\_type == 'windows':

print("To re-enable network:")

print(" 1. Open Network Connections (ncpa.cpl)")

print(" 2. Right-click on disabled adapters")

print(" 3. Select 'Enable'")

print("="\*50)

def main():

"""Main function"""

print("=== NETWORK DISABLER ===")

print("⚠️ WARNING: This will disable ALL network connections!")

print("You will lose internet and local network access.")

print()

# Check OS compatibility

os\_type = detect\_os()

if os\_type not in ['linux', 'windows']:

print(f"Unsupported operating system: {os\_type}")

print("Currently supports Linux and Windows only.")

sys.exit(1)

# Check privileges

privileged, message = check\_privileges()

if not privileged:

print(f"Error: {message}")

sys.exit(1)

# Final confirmation

confirm = input("Are you sure you want to continue? (yes/NO): ")

if confirm.lower() != 'yes':

print("Operation cancelled")

return

print("\nDisabling network...")

# Disable network based on OS

success = False

if os\_type == 'linux':

success = disable\_network\_linux()

elif os\_type == 'windows':

success = disable\_network\_windows()

if success:

print("\n✅ Network successfully disabled!")

show\_reactivation\_instructions()

else:

print("\n❌ Failed to disable network")

if \_\_name\_\_ == "\_\_main\_\_":

main()