“pip install psutil nmap paramiko pycryptodome”

1. Analyze and Visualize IP Address Allocation and Subnetting

Code:

```
from scapy.all import IP, ICMP, sr1
import matplotlib.pyplot as plt
import ipaddress

def calculate_subnets(base_ip, subnet_mask, num_subnets):
    network = ipaddress.ip_network(f"{base_ip}/{subnet_mask}", strict=False)
    subnets = list(network.subnets(new_prefix=num_subnets))
    return subnets

def visualize_subnets(subnets):
    fig, ax = plt.subplots()
    for i, subnet in enumerate(subnets):
        ax.barh(i, subnet.num_addresses, label=f"{subnet}")
    ax.set_yticks(range(len(subnets)))
    ax.set_yticklabels([str(subnet) for subnet in subnets])
    ax.set_xlabel("Number of Addresses")
    ax.set_title("Subnet Allocation")
    plt.legend()
    plt.show()

# Example usage
```

```
base_ip = "192.168.1.0"
subnet_mask = 24
num_subnets = 4
subnets = calculate_subnets(base_ip, subnet_mask, num_subnets)
visualize_subnets(subnets)
```

2. Read and Modify Linux Network Configuration Files

Code:

```
import os


def read_file(file_path):
    with open(file_path, 'r') as file:
        return file.readlines()


def write_file(file_path, lines):
    with open(file_path, 'w') as file:
        file.writelines(lines)


def modify_hosts(ip, hostname):
    file_path = "/etc/hosts"
    lines = read_file(file_path)
    new_entry = f"{ip}\t{hostname}\n"
    if new_entry not in lines:
        lines.append(new_entry)
        write_file(file_path, lines)
        print(f"Added {ip} {hostname} to {file_path}")
    else:
        print(f"{ip} {hostname} already exists in {file_path}")

def modify_resolv_conf(nameserver):
    file_path = "/etc/resolv.conf"
    lines = read_file(file_path)
    new_entry = f"nameserver {nameserver}\n"
    if new_entry not in lines:
        lines.append(new_entry)
        write_file(file_path, lines)
        print(f"Added nameserver {nameserver} to {file_path}")
    else:
        print(f"Nameserver {nameserver} already exists in {file_path}")

def modify_interfaces(interface, config):
    file_path = "/etc/network/interfaces"
    lines = read_file(file_path)
    if config not in lines:
        lines.append(f"\n{config}\n")
        write_file(file_path, lines)
        print(f"Added {config} to {file_path}")
    else:
        print(f"{config} already exists in {file_path}")

# Example usage
modify_hosts("192.168.1.100", "myserver")
modify_resolv_conf("8.8.8.8")
modify_interfaces("eth0", "auto eth0\niface eth0 inet dhcp")
```

3. Capture and Analyze Active TCP/IP Daemons

Code:

```
import psutil
import subprocess

def get_active_connections():
    connections = psutil.net_connections(kind='tcp')
    active_daemons = {}
    for conn in connections:
        if conn.status == psutil.CONN_ESTABLISHED:
            pid = conn.pid
            if pid:
                process = psutil.Process(pid)
                daemon_name = process.name()
                if daemon_name not in active_daemons:
                    active_daemons[daemon_name] = 0
                active_daemons[daemon_name] += 1
    return active_daemons

def visualize_daemons(active_daemons):
    names = list(active_daemons.keys())
    counts = list(active_daemons.values())
    plt.bar(names, counts)
    plt.xlabel("Daemon Name")
    plt.ylabel("Number of Connections")
    plt.title("Active TCP/IP Daemons")
    plt.xticks(rotation=45)
    plt.show()

# Example usage
active_daemons = get_active_connections()
print("Active TCP/IP Daemons:", active_daemons)
visualize_daemons(active_daemons)
```

4. Simple Python-Based Network Daemon

Code:

```
import socket
import logging

def start_daemon(host='0.0.0.0', port=9999):
    logging.basicConfig(filename='network_daemon.log', level=logging.INFO,
                        format='%(asctime)s - %(message)s')
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind((host, port))
    server_socket.listen(5)
    logging.info(f"Daemon started on {host}:{port}")

    while True:
        client_socket, addr = server_socket.accept()
        logging.info(f"Connection from {addr}")
        client_socket.send(b"Connection logged.\n")
        client_socket.close()
```

```
# Example usage
start_daemon()
```

5. Scan and List Open Ports

Code:

```
import socket
import nmap


def scan_ports(target, port_range):
    nm = nmap.PortScanner()
    nm.scan(target, port_range)
    open_ports = []
    for host in nm.all_hosts():
        for proto in nm[host].all_protocols():
            ports = nm[host][proto].keys()
            for port in ports:
                if nm[host][proto][port]['state'] == 'open':
                    open_ports.append(port)
    return open_ports


# Example usage
target = "192.168.1.1"
port_range = "1-1024"
open_ports = scan_ports(target, port_range)
print(f"Open ports on {target}: {open_ports}")
```

6. Java Program to Extract Network Settings

Code:

```
import java.net.*;
import java.util.*;

public class NetworkInfo {
    public static void main(String[] args) throws SocketException {
        Enumeration<NetworkInterface> interfaces =
NetworkInterface.getNetworkInterfaces();
        while (interfaces.hasMoreElements()) {
            NetworkInterface iface = interfaces.nextElement();
            if (iface.isUp() && !iface.isLoopback()) {
                System.out.println("Interface: " + iface.getDisplayName());
                for (InterfaceAddress addr : iface.getInterfaceAddresses()) {
                    InetAddress inetAddr = addr.getAddress();
                    if (inetAddr instanceof Inet4Address) {
                        System.out.println("IP Address: " + inetAddr.getHostAddress());
                        System.out.println("Subnet Mask: " +
getSubnetMask(addr.getNetworkPrefixLength()));
                        System.out.println("Default Gateway: " +
getDefaultGateway(iface));
                    }
                }
            }
        }
    }

    private static String getSubnetMask(short prefix) {
        int mask = 0xFFFFFFFF << (32 - prefix);
        return String.format("%d.%d.%d.%d",
                (mask >> 24) & 0xFF,
                (mask >> 16) & 0xFF,
                (mask >> 8) & 0xFF,
                mask & 0xFF);
    }

    private static String getDefaultGateway(NetworkInterface iface) {
        try {
            for (InterfaceAddress addr : iface.getInterfaceAddresses()) {
                InetAddress inetAddr = addr.getAddress();
                if (inetAddr instanceof Inet4Address) {
                    return inetAddr.getHostAddress();
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return "Unknown";
    }
}
```

7. Log Incoming and Outgoing Network Connections

Code:

```
import psutil
import logging
import time
```

```
logging.basicConfig(filename='network_connections.log', level=logging.INFO,
                    format='%(asctime)s - %(message)s')

def log_connections():
    while True:
        connections = psutil.net_connections(kind='inet')
        for conn in connections:
            if conn.status == psutil.CONN_ESTABLISHED:
                logging.info(f"{conn.laddr.ip}:{conn.laddr.port} ->
{conn.raddr.ip}:{conn.raddr.port}")
        time.sleep(10)  # Log every 10 seconds

# Example usage
log_connections()
```

8. Monitor Unauthorized Changes to Network Configuration Files

Code:

```
import os
import time
import hashlib

def get_file_hash(file_path):
    hasher = hashlib.md5()
    with open(file_path, 'rb') as f:
        buf = f.read()
        hasher.update(buf)
    return hasher.hexdigest()
```

```
def monitor_files(file_paths):
    hashes = {file_path: get_file_hash(file_path) for file_path in file_paths}
    while True:
        time.sleep(10)  # Check every 10 seconds
        for file_path in file_paths:
            current_hash = get_file_hash(file_path)
            if current_hash != hashes[file_path]:
                print(f"Unauthorized change detected in {file_path}")
                hashes[file_path] = current_hash

# Example usage
file_paths = ["/etc/network/interfaces", "/etc/resolv.conf"]
monitor_files(file_paths)
```

9. Encrypt and Decrypt Files for FTP Transfer

Code:

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad
import os

KEY = os.urandom(32)  # 256-bit key

def encrypt_file(file_path):
    cipher = AES.new(KEY, AES.MODE_CBC)
    with open(file_path, 'rb') as f:
        plaintext = f.read()
    ciphertext = cipher.encrypt(pad(plaintext, AES.block_size))
    with open(file_path + ".enc", 'wb') as f:
        f.write(cipher.iv + ciphertext)

def decrypt_file(encrypted_path):
    with open(encrypted_path, 'rb') as f:
        iv = f.read(16)
        ciphertext = f.read()
    cipher = AES.new(KEY, AES.MODE_CBC, iv)
    plaintext = unpad(cipher.decrypt(ciphertext), AES.block_size)
    with open(encrypted_path[:-4], 'wb') as f:
        f.write(plaintext)

# Example usage
encrypt_file("example.txt")
decrypt_file("example.txt.enc")
```

10. SSH Brute-Force Attack Detection

Code:

```
import paramiko
import logging

logging.basicConfig(filename='ssh_attempts.log', level=logging.INFO,
                    format='%(asctime)s - %(message)s')

def detect_brute_force(host, port=22):
    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    while True:
        try:
            ssh.connect(host, port=port, username='root',
password='wrongpassword')
        except paramiko.ssh_exception.AuthenticationException:
            logging.warning("Failed SSH login attempt detected")

# Example usage
detect_brute_force("192.168.1.1")
```