

Unclassified

```
import pyshark
import sys
from ipaddress import ip_network, IPv4Address

def parse_pcap(pcap_file, attribute, ip_range=None):
    # Open the PCAP file for parsing
    capture = pyshark.FileCapture(pcap_file)

    # Iterate over each packet in the capture
    for packet in capture:
        # Check if the desired attribute is present in the packet
        if hasattr(packet, attribute):
            # Check if filtering by IP range is enabled
            if ip_range is not None:
                # Extract the source IP address
                source_ip = packet.ip.src

                # Check if the source IP is within the specified range
                if is_ip_in_range(source_ip, ip_range):
                    # Extract and print the attribute value
                    value = getattr(packet, attribute)
                    print(value)
            else:
                # Extract and print the attribute value without filtering
                value = getattr(packet, attribute)
                print(value)

    # Close the capture
    capture.close()

def is_ip_in_range(ip, ip_range):
    # Convert the IP address to IPv4Address object
    ip_address = IPv4Address(ip)

    # Iterate over the IP range networks
    for network in ip_range:
        if ip_address in network:
            return True

    return False

if __name__ == "__main__":
    # Check if the PCAP file path and attribute are provided as command-line
    arguments
    if len(sys.argv) < 3:
        print("Usage: python pcap_parser.py <pcap_file> <attribute> [ip_range]")
        sys.exit(1)

    pcap_file = sys.argv[1]
    attribute = sys.argv[2]
```

```
# Check if IP range is provided as command-line argument
ip_range = None
if len(sys.argv) == 4:
    ip_range = [ip_network(sys.argv[3])]

# Call the parse_pcap function
parse_pcap(pcap_file, attribute, ip_range)
```

In this expanded version, we added the `is_ip_in_range` function that checks if an IP address is within the specified IP range. We convert the IP address to an `IPv4Address` object and iterate over the IP range networks to determine if the IP address falls within any of the networks.

The `parse_pcap` function has been modified to include the `ip_range` parameter. If an IP range is provided, it extracts the source IP address from each packet and checks if it is within the specified range using the `is_ip_in_range` function. Only packets with source IP addresses within the range will have their attribute values printed.

You can run the program as before, providing the PCAP file path, the desired attribute, and an optional IP address range as command-line arguments. For example:

```
python pcap_parser.py my_capture.pcap ip.src 192.168.0.0/24
```

This version of the program allows you to filter packets based on a specific IP address range, in addition to extracting the desired attribute. Unclassified