COMP 7003

Assignment 2

Design

Aditya Singh Attri

A01276335

Oct 5th, 2024

# Purpose

This program captures network packets using **Scapy** based on user-defined filters (ARP, IPv4, TCP, UDP) and parses the relevant headers manually without using Scapy’s built-in functions for field extraction.

# Data Types

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| interface | string | The name of the network interface on which the program listens (e.g., Wi-Fi, Ethernet) |
| capture\_filter | string | The filter used to capture packets. This can be either arp, tcp, udp, or ip |
| packet\_count | int | The number of packets to capture before stopping the program |
| raw\_data | byte | The raw byte data of the captured packet |
| hex\_data | string | The hexadecimal representation of the packet |

# Pseudocode

## capture\_packets

### Parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| interface | string | The network interface to listen on (e.g., Wi-Fi, Ethernet) |
| capture\_filter | string | The protocol filter to apply (e.g., arp, tcp, udp, ip) |
| packet\_count | int | The number of packets to capture before exiting |

### Pseudocode

If capture\_filter is not valid:

Print error message

Exit program

Try to capture packets on the given interface with the provided filter:

If interface is invalid, catch the error and display a message.

Call packet\_callback function to process each captured packet.

## packet\_callback

### Parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| packet | string | the captured packet |
| capture\_filter | string | The protocol filter applied during capture |

### Pseudocode

Convert raw packet data to hexadecimal.

If EtherType is ARP (0x0806):

Call parse\_arp\_packet function.

Else if EtherType is IPv4 (0x0800):

If capture\_filter is 'ip', call parse\_ipv4\_header\_only function to show only IPv4 headers.

Else, call parse\_ipv4\_packet to extract full IPv4 as well as TCP or UDP headers.

## parse\_ethernet\_header

### Parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| hex\_data | string | the hexadecimal representation of the captured packet |

### Pseudocode

Extract destination MAC, source MAC, and EtherType from the Ethernet header.

Convert MAC addresses from hex to readable format.

Print MAC addresses and EtherType.

## parse\_arp\_packet

### Parameter

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| hex\_data | string | the hexadecimal representation of the ARP packet |

### Pseudocode

Extract ARP fields: hardware type, protocol type, sender MAC, sender IP, target MAC, target IP.

Convert IP addresses from hex to dotted-decimal format.

Print extracted ARP fields.

## parse\_ipv4\_header\_only

### Parameter

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| hex\_data | string | the hexadecimal representation of the IPv4 packet |

### Pseudocode

Extract IPv4 header fields: version, IHL, total length, protocol, source IP, destination IP.

Convert IP addresses from hex to dotted-decimal format.

Print extracted IPv4 fields.

## parse\_ipv4\_packet

### Parameter

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| hex\_data | string | the hexadecimal representation of the IPv4 packet |

### Pseudocode

Extract IPv4 header fields.

Determine protocol (TCP or UDP).

If protocol is TCP:

Call parse\_tcp\_packet to extract TCP header fields.

Else if protocol is UDP:

Call parse\_udp\_packet to extract UDP header fields.

## parse\_tcp\_packet

### Parameter

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| hex\_data | string | the hexadecimal representation of the TCP packet |

### Pseudocode

Extract TCP fields: source port, destination port, sequence number, acknowledgment number, flags, window size.

Print extracted TCP fields.

## parse\_udp\_packet

### Parameter

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| hex\_data | string | the hexadecimal representation of the UDP packet |

### Pseudocode

Extract UDP fields: source port, destination port, length, checksum.

Print extracted UDP fields.