Unclassified

PyShark

The PyShark library is a Python wrapper for the popular packet analysis tool called Wireshark. It allows you to capture, dissect, and analyze network packets programmatically using Python. Here's a brief introduction and explanation of PyShark:

1. Packet Analysis:
    - PyShark provides a high-level interface to capture, analyze, and process network packets.
    - It allows you to read packet capture files (PCAP or PCAPNG format) or capture live packets from network interfaces.
    - PyShark supports various network protocols and provides access to packet headers, fields, and data.
2. Features and Functionality:
    - PyShark provides several functionalities for packet analysis, including:
        - Packet filtering: You can define filters to capture specific packets based on protocol, source/destination IP addresses, ports, etc.
        - Packet dissection: PyShark allows you to dissect packets and access individual protocol layers, fields, and their values.
        - Statistics and metrics: You can extract statistical information from captured packets, such as packet counts, bandwidth usage, and protocol distribution.
        - Stream reassembly: PyShark can reassemble fragmented packets into complete streams for further analysis.
        - Decryption: It supports decrypting encrypted network traffic using decryption keys or certificates.
3. Usage Example:
    - Here's a simple example that demonstrates how to use PyShark to capture and analyze network packets:

```
import pyshark

# Capture packets on a network interface
capture = pyshark.LiveCapture(interface='eth0')

# Set a display filter to capture specific packets
capture.set_display_filter('http')

# Start capturing packets
capture.sniff(timeout=10)

# Process captured packets
for packet in capture:
```

```
    # Access packet information
    print(packet.tcp.srcport, packet.http.request_uri)

# Close the capture
capture.close()
```

4. Benefits:

    - PyShark provides a Pythonic interface for packet analysis, allowing you to leverage Python's flexibility and functionality for network traffic analysis.
    - It enables automation and scripting of packet analysis tasks, making it suitable for network monitoring, security analysis, and troubleshooting.
    - PyShark integrates well with other Python libraries and tools, enabling seamless integration into larger projects or workflows.

PyShark simplifies the process of capturing and analyzing network packets, making it a valuable tool for network engineers, security analysts, and developers working with network-related applications. It combines the power of Wireshark with the ease of Python programming, providing a convenient way to work with network packet data. Unclassified