

tun_demo

In the tun_demo project, I explored the practical operation of a TUN virtual network interface in Linux. I implemented creation of a TUN interface from user space via /dev/net/tun, system-level configuration including IP address assignment, interface activation, and routing, as well as packet exchange between the Linux kernel and a user-space application. The project includes reading and writing raw IPv4 packets, demonstrating the operation of TUN as a Layer 3 (IP) virtual device. The project is intended for educational purposes and does not implement a full VPN, encryption, or a custom protocol.

Reference material:

<https://en.wikipedia.org/wiki/TUN/TAP>

<https://www.kernel.org/doc/html/latest/networking/tuntap.html>

udp_demo

In the udp_demo project, I studied low-level UDP socket communication in Linux. I implemented UDP socket creation and datagram transmission using the sendto() and recvfrom() system calls and experimentally analyzed the effects of packet size, MTU, fragmentation, and packet loss. The project demonstrates the unreliable and connectionless nature of UDP, which is commonly used as a transport layer in VPN systems.

Reference material:

https://en.wikipedia.org/wiki/User_Datagram_Protocol

<https://man7.org/linux/man-pages/man7/udp.7.html>

tun_userspace_demo

In the tun_userspace_demo project, I examined the interaction between the Linux kernel networking stack and a user-space application via a TUN interface. I implemented packet transfer using the read() and write() system calls, demonstrating the complete path of IP packets through a virtual network interface.

Reference material:

<https://en.wikipedia.org/wiki/TUN/TAP>

<https://man7.org/linux/man-pages/man2/read.2.html>

<https://man7.org/linux/man-pages/man2/write.2.html>

tun_udp_demo

In the tun_udp_demo project, I implemented a basic VPN-like packet forwarding model using a TUN interface and UDP transport. The project demonstrates IP packet encapsulation, transmission over UDP, and reinjection of packets into the Linux kernel on the remote side, focusing on the conceptual mechanisms of VPN operation.

Reference material:

https://en.wikipedia.org/wiki/Virtual_private_network

[https://en.wikipedia.org/wiki/Encapsulation_\(networking\)](https://en.wikipedia.org/wiki/Encapsulation_(networking))

crypto_demo

In the crypto_demo project, I studied symmetric cryptography using the libsodium library. I implemented authenticated encryption and decryption using the ChaCha20-Poly1305 AEAD algorithm and verified data integrity after decryption. The project demonstrates correct usage of cryptographic primitives without implementing a full cryptographic protocol.

Reference material:

https://en.wikipedia.org/wiki/Symmetric-key_algorithm

https://en.wikipedia.org/wiki/Authenticated_encryption

<https://en.wikipedia.org/wiki/Nonce>

<https://en.wikipedia.org/wiki/ChaCha20-Poly1305>

<https://libsodium.org/>

<https://docs.rs/libsodium/latest/libsodium/>

Overall outcome

The knowledge and practical experience gained from these demonstration projects form a foundation for implementing a full VPN system in the next semester. The projects collectively cover the essential building blocks of a VPN: virtual network interfaces (TUN), packet routing, unreliable transport (UDP), and cryptographic primitives, which can later be combined into a complete secure networking solution.