**Abstract.** This report documents data exchange process between two machines using a VPN

tunneling application.

# Purpose of the environment

From this experiment, our team wants to understand the mechanism of IPsec & VPN and their applications in reality.

# Experimental setup

* 1. Hardware requirements
* Two laptops running on Linux OS.
  1. Software setup
* Turn on Wireshark and start capturing packets.
* The remaining steps are demonstrated by the tables below:

|  |  |  |
| --- | --- | --- |
| **Step** | **Laptop A** | **Laptop B** |
| Step 1 | Download “Wireguard” | Download “Wireguard” |
| Step 2 | Type “sudo bash” | Type “sudo bash” |
| Step 3 | Type “wg genkey > private” | Type “wg genkey > private” |
| Step 3.1 |  | Type “cat private” |
| Step 3.2 |  | Type “wg pubkey < private” |

Table 1: Step 1, 2, and 3

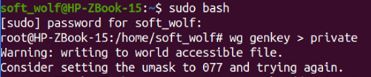


Figure 1: Laptop A’s terminal after step 1,2 and 3

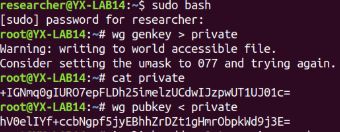


Figure 2: Laptop B’s terminal after step 1,2 and 3

|  |  |  |
| --- | --- | --- |
| **Step** | **Laptop A** | **Laptop B** |
| Step 4 | Type “ip link add wg0 type wireguard” | Type “ip link add wg0 type wireguard” |
| Step 5 | Type “ip addr add 10.0.0.1/24 dev wg0” | Type “ip addr add 10.0.0.1/24 dev wg0” |
| Step 6 | Type “wg set wg0 private-key  ./private” | Type “wg set wg0 private-key  ./private” |
| Step 7 | Type “ip link set wg0 up” | Type “ip link set wg0 up” |
| Step 8 | Type “ip addr” | Type “ip addr” |

Table 2: Step 4, 5, 6 ,7, and 8

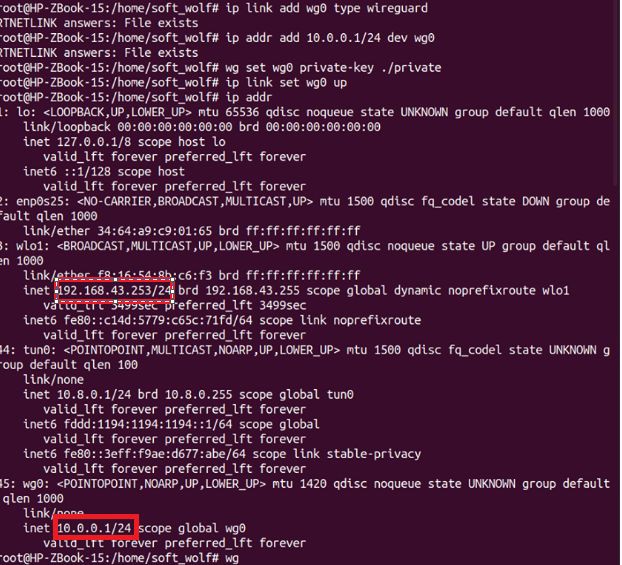


Figure 3: Laptop A’s terminal and IP addresses after step 4, 5, 6, 7, and 8

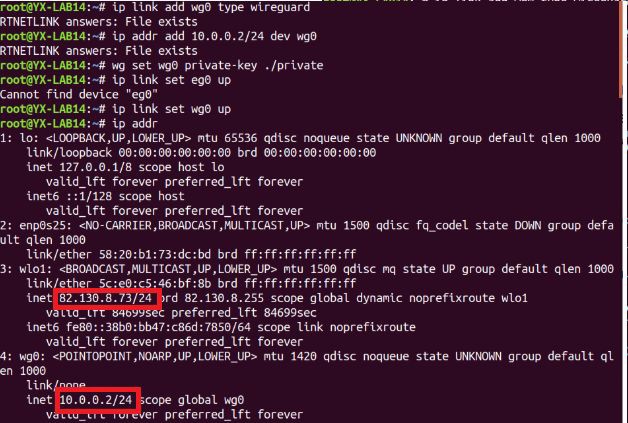


Figure 4: Laptop B’s terminal and IP addresses after step 4, 5, 6, 7, and 8

|  |  |  |
| --- | --- | --- |
| **Step** | **Laptop A** | **Laptop B** |
| Step 9 | Type “wg” | Type “wg” |
| Step 10 | Type “wg set wg0 peer [Laptop B’s public key] allowed-ips 10.0.0.2/32 endpoint [Laptop B’s IP address]:[Laptop B’s port]” | Type “wg set wg0 peer [Laptop A’s public key] allowed-ips 10.0.0.1/32 endpoint [Laptop A’s IP address]:[Laptop A’s port]” |
| Step 11 | Type “ping 10.0.0.2” |  |

Table 3: Step 9, 10, and 11

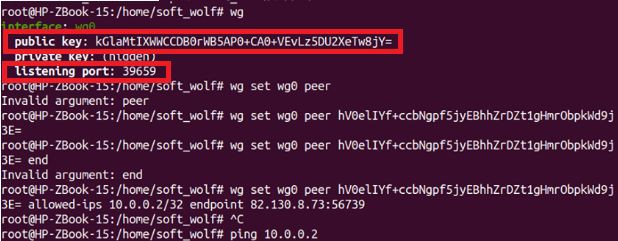


Figure 5: Laptop A’s terminal, public key and listening port after step 9, 10, and 11.



Figure 6: Laptop B’s terminal, public key and listening port after step 9, 10, and 11

# Result

* 1. Laptop A ping Laptop B

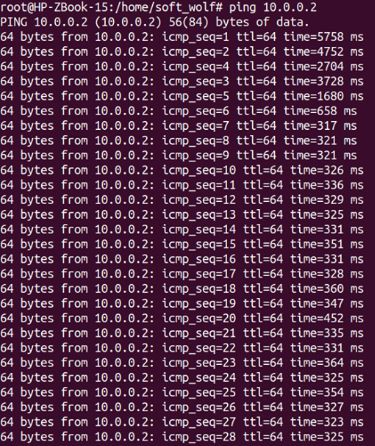


Figure 7: Laptop A ping to Laptop B

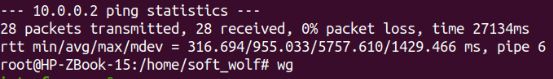


Figure 8: Statistics of the ping process from Laptop A to Laptop B

* 1. The connection between Laptop A and Laptop B

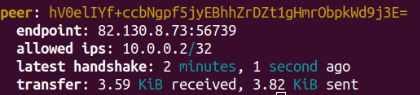


Figure 9: The connection of Laptop A to Laptop B

* “Peer” of Laptop A is expressed by the public key of Laptop B

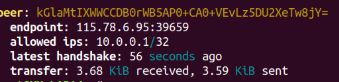


Figure 10: The connection of Laptop B to Laptop A

* “Peer” of Laptop B is expressed by the public key of Laptop A

⇒ The number of data Laptop A received was equal to the number of data Laptop B sent

⇒ Laptop A and Laptop B are “Peer” of each other

* 1. VPN traffic



Figure 11: A VPN packet

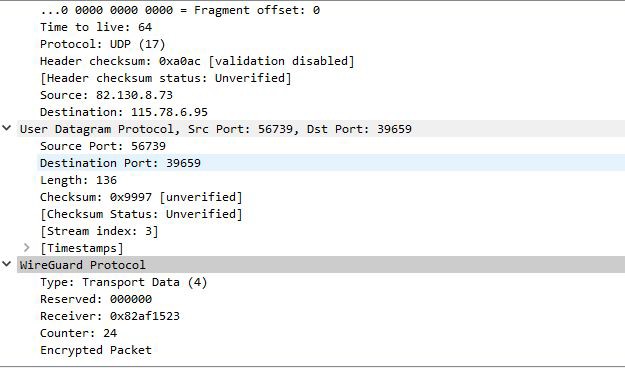


Figure 12: Configuration of the VPN packet

# Conclusion

In the experiment, we created a tunnel that goes both ways. On one end, we have a server and a client on the other end. Wireguard does not care which machine is the server or which is client.