

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

I. Purpose of the environment

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

II. Experimental setup

1. Hardware requirements

- Two laptops running on Linux OS.

2. 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

```

soft_wolf@HP-ZBook-15:~$ sudo bash
[sudo] password for soft_wolf:
root@HP-ZBook-15:/home/soft_wolf# wg genkey > private
Warning: writing to world accessible file.
Consider setting the umask to 077 and trying again.

```

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

```

researcher@YX-LAB14:~$ sudo bash
[sudo] password for researcher:
root@YX-LAB14:~# wg genkey > private
Warning: writing to world accessible file.
Consider setting the umask to 077 and trying again.
root@YX-LAB14:~# cat private
+IGNmq0gIURO7epFLDh25imelzUCdwIJzpwUT1UJ01c=
root@YX-LAB14:~# wg pubkey < private
hV0elIYf+ccbNgpf5jyEBhhZrDZt1gHmrObpkWd9j3E=

```

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

```

root@HP-ZBook-15:/home/soft_wolf# ip link add wg0 type wireguard
RTNETLINK answers: File exists
root@HP-ZBook-15:/home/soft_wolf# ip addr add 10.0.0.1/24 dev wg0
RTNETLINK answers: File exists
root@HP-ZBook-15:/home/soft_wolf# wg set wg0 private-key ./private
root@HP-ZBook-15:/home/soft_wolf# ip link set wg0 up
root@HP-ZBook-15:/home/soft_wolf# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s25: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 34:64:a9:c9:01:65 brd ff:ff:ff:ff:ff:ff
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether f8:16:54:8b:c6:f3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.43.253/24 brd 192.168.43.255 scope global dynamic noprefixroute wlo1
        valid_lft 3499sec preferred_lft 3499sec
    inet6 fe80::c14d:5779:c65c:71fd/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 100
    link/none
    inet 10.8.0.1/24 brd 10.8.0.255 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fddd:1194:1194::1/64 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::3eff:f9ae:d677:abe/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
5: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group default qlen 1000
    link/none
    inet 10.0.0.1/24 scope global wg0
        valid_lft forever preferred_lft forever
root@HP-ZBook-15:/home/soft_wolf# wg

```

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

```

root@YX-LAB14:~# ip link add wg0 type wireguard
RTNETLINK answers: File exists
root@YX-LAB14:~# ip addr add 10.0.0.2/24 dev wg0
RTNETLINK answers: File exists
root@YX-LAB14:~# wg set wg0 private-key ./private
root@YX-LAB14:~# ip link set eg0 up
Cannot find device "eg0"
root@YX-LAB14:~# ip link set wg0 up
root@YX-LAB14:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s25: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 58:20:b1:73:dc:bd brd ff:ff:ff:ff:ff:ff
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 5c:e0:c5:46:bf:8b brd ff:ff:ff:ff:ff:ff
    inet 82.130.8.73/24 brd 82.130.8.255 scope global dynamic noprefixroute wlo1
        valid_lft 84699sec preferred_lft 84699sec
    inet6 fe80::38b0:bb47:c86d:7850/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
4: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group default qlen 1000
    link/none
    inet 10.0.0.2/24 scope global wg0
        valid_lft forever preferred_lft forever

```

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


```

root@HP-ZBook-15:/home/soft_wolf# wg
interface: wg0
  public key: kGlaMtIXWCCDB0rWB5AP0+CA0+VEvLz5DU2XeTw8jY=
  private key: (hidden)
  listening port: 39659
root@HP-ZBook-15:/home/soft_wolf# wg set wg0 peer
Invalid argument: peer
root@HP-ZBook-15:/home/soft_wolf# wg set wg0 peer hV0eLIYf+ccbNgpf5jyEBhhZrDZt1gHmr0bpkWd9j3E=
root@HP-ZBook-15:/home/soft_wolf# wg set wg0 peer hV0eLIYf+ccbNgpf5jyEBhhZrDZt1gHmr0bpkWd9j3E= end
Invalid argument: end
root@HP-ZBook-15:/home/soft_wolf# wg set wg0 peer hV0eLIYf+ccbNgpf5jyEBhhZrDZt1gHmr0bpkWd9j3E= allowed-ips 10.0.0.2/32 endpoint 82.130.8.73:56739
root@HP-ZBook-15:/home/soft_wolf# ^C
root@HP-ZBook-15:/home/soft_wolf# ping 10.0.0.2

```

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

```

root@YX-LAB14:~# wg
interface: wg0
  public key: hV0eLIYf+ccbNgpf5jyEBhhZrDZt1gHmr0bpkWd9j3E=
  private key: (hidden)
  listening port: 56739

```

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

III. Result

1. Laptop A ping Laptop B

```

root@HP-ZBook-15:/home/soft_wolf# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=5758 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=4752 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=2704 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=3728 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=1680 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=658 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=317 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=321 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=321 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=326 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=336 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=329 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=325 ms
64 bytes from 10.0.0.2: icmp_seq=14 ttl=64 time=331 ms
64 bytes from 10.0.0.2: icmp_seq=15 ttl=64 time=351 ms
64 bytes from 10.0.0.2: icmp_seq=16 ttl=64 time=331 ms
64 bytes from 10.0.0.2: icmp_seq=17 ttl=64 time=328 ms
64 bytes from 10.0.0.2: icmp_seq=18 ttl=64 time=360 ms
64 bytes from 10.0.0.2: icmp_seq=19 ttl=64 time=347 ms
64 bytes from 10.0.0.2: icmp_seq=20 ttl=64 time=452 ms
64 bytes from 10.0.0.2: icmp_seq=21 ttl=64 time=335 ms
64 bytes from 10.0.0.2: icmp_seq=22 ttl=64 time=331 ms
64 bytes from 10.0.0.2: icmp_seq=23 ttl=64 time=364 ms
64 bytes from 10.0.0.2: icmp_seq=24 ttl=64 time=325 ms
64 bytes from 10.0.0.2: icmp_seq=25 ttl=64 time=354 ms
64 bytes from 10.0.0.2: icmp_seq=26 ttl=64 time=327 ms
64 bytes from 10.0.0.2: icmp_seq=27 ttl=64 time=323 ms
64 bytes from 10.0.0.2: icmp_seq=28 ttl=64 time=325 ms

```

Figure 7: Laptop A ping to Laptop B

```

--- 10.0.0.2 ping statistics ---
28 packets transmitted, 28 received, 0% packet loss, time 27134ms
rtt min/avg/max/mdev = 316.694/955.033/5757.610/1429.466 ms, pipe 6
root@HP-ZBook-15:/home/soft_wolf# wg

```

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

2. The connection between Laptop A and Laptop B

```

peer: hV0eLIYf+ccbNgpf5jyEBhhZrDZt1gHmrObpkWd9j3E=
endpoint: 82.130.8.73:56739
allowed ips: 10.0.0.2/32
latest handshake: 2 minutes, 1 second ago
transfer: 3.59 KiB received, 3.82 KiB sent

```

Figure 9: The connection of Laptop A to Laptop B

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

```
peer: kGlaMtIXWCCDB0rWBSAP0+CA0+VEvLz5DU2XeTw8jY=
endpoint: 115.78.6.95:39659
allowed ips: 10.0.0.1/32
latest handshake: 56 seconds ago
transfer: 3.68 KiB received, 3.59 KiB sent
```

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

3. VPN traffic

```
83... 18.5909... 82.130.8.73 115.78.6.95 WireG... 170 |Transport Data, receiver=0x82AF1523, counter=24, datalen=96
```

Figure 11: A VPN packet

```
...0 0000 0000 0000 = Fragment offset: 0
Time to live: 64
Protocol: UDP (17)
Header checksum: 0xa0ac [validation disabled]
[Header checksum status: Unverified]
Source: 82.130.8.73
Destination: 115.78.6.95
▼ User Datagram Protocol, Src Port: 56739, Dst Port: 39659
  Source Port: 56739
  Destination Port: 39659
  Length: 136
  Checksum: 0x9997 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 3]
  > [Timestamps]
▼ WireGuard Protocol
  Type: Transport Data (4)
  Reserved: 000000
  Receiver: 0x82af1523
  Counter: 24
  Encrypted Packet
```

Figure 12: Configuration of the VPN packet

IV. 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.