

WireGuard

0826 – Redes - Instalação e Configuração

DANIEL AMADOR – T0133582
IAN REBELO - T0130772



WIREGUARD
FAST. MODERN. SECURE VPN TUNNEL



project by:

BLACKPEGASUS

Introdução

- ▶ Alguém sabe o que é uma **VPN**?
- ▶ Com o crescimento do trabalho remoto e para este seja seguro o uso de uma **VPN** é essencial!
- ▶ No nosso trabalho vamos usar uma **VPN** chamada **WireGuard**, sendo esta uma solução moderna de **VPN**.

O que é o WireGuard?

- ▶ O **WireGuard** é um protocolo e software **VPN** moderno, focado em:
 - Simplicidade de código;
 - Alta performance (implementado ao nível do kernel em Linux);
 - Criptografia de última geração;
 - Baixa latência e baixo overhead;
 - Configuração minimalista.
- ▶ Utiliza pacotes **UDP** para encapsular o tráfego encriptado, permitindo uma comunicação mais rápida e eficiente, **sem comprometer a segurança** do túnel.
- ▶ O **WireGuard** é **Open Source** e é suportado nativamente por vários sistemas operativos, como **Linux, Windows, macOS, iOS e Android**.

Arquitetura

- **Criptografia moderna:** Curve25519, ChaCha20, Poly1305, BLAKE2s, SipHash24, HKDF;
- **Modelo peer-to-peer:** cada nó atua como cliente e servidor;
- **Interface virtual:** cria uma interface de rede virtual;
- **Roaming integrado:** permite mudar de IP sem perder a sessão;
- **Cryptokey Routing:** cada chave pública está associada a IPs permitidos.

Funcionamento

- ▶ O **WireGuard** opera na **Layer 3 do modelo OSI (Network Layer)**.
- ▶ Cada dispositivo possui um **par de Chaves Criptográficas (Pública e Privada)** para **autenticação**.

Chave Pública - é partilhada com outro dispositivo;

Chave Privada – nunca é partilhada.

Os dispositivos reconhecem-se pela **Chave Pública** e provam quem são pela **Chave Privada**. A **Chave Privada** é simplesmente **usada localmente** para **autenticar** e assim **encriptar a comunicação**.

Casos Uso e Vantagens

Casos Uso

- ▶ **VPN** corporativa simples, rápida e segura;
- ▶ **VPN** pessoal para segurança e privacidade;
- ▶ Acesso remoto seguro a servidores;
- ▶ Túneis seguros entre datacenters;
- ▶ **VPN** para dispositivos móveis (roaming eficiente);
- ▶ Proteção e privacidade em redes públicas.

Vantagens

- ▶ Desempenho superior ao **OpenVPN** e **IPsec**;
- ▶ Configuração simples;
- ▶ Código reduzido, facilitando auditoria;
- ▶ Segurança moderna com criptografia atualizada;
- ▶ Roaming automático, ideal para dispositivos móveis;
- ▶ Compatível com vários sistemas operativos.

Comparação com Alternativas

- ▶ O **WireGuard** destaca-se pela **simplicidade** e **eficiência**, enquanto o **OpenVPN** e o **IPsec** oferecem **mais flexibilidade** mas com **maior complexidade**.
- ▶ Comparado com **OpenVPN** e **IPSec**, o **WireGuard** tende a ser **mais rápido**, **mais seguro**, e muito **mais simples de configurar**!

Critério	WireGuard	OpenVPN	IPsec
Desempenho	Muito alto	Médio	Alto
Complexidade	Baixa	Média	Alta
Criptografia	Moderna	Variável	Forte mas complexa
Código-fonte	Pequeno	Grande	Muito grande
Facilidade de configuração	Muito fácil	Média	Difícil
Suporte móvel	Excelente	Bom	Médio
Kernel Linux	Integrado	Não	Parcial

Laboratório Realizado

- ▶ Para a nossa demonstração estabelecemos uma ligação com o **WireGuard** entre um **Cliente** e um **Servidor** e mostramos que esta ligação só funciona se tivermos o túnel do **WireGuard** ativo entre o **Cliente** e o **Servidor**.

Cliente:

Edit tunnel

Name: Minecraft

Public key: yktfIr7QfFsW0HrsbvqDIbwqF8wTIYVTAXUA9plARs=

[Interface]
PrivateKey = lJoCFOW1M+vxTynnQnZkVab2NpBs/qeiqVzi00KP1E=
ListenPort = 51820
Address = 2.0.0.2/28

[Peer]
PublicKey = 3lG02Xo0ePIStIa3DdNSjeThd8F5UGWCjZK4JOJR58=
AllowedIPs = 2.0.0.0/28, 192.168.0.0/24
Endpoint = 192.168.0.1:51820

Save Cancel

Server:

Edit tunnel

Name: Minecraft

Public key: 3lG02Xo0ePIStIa3DdNSjeThd8F5UGWCjZK4JOJR58=

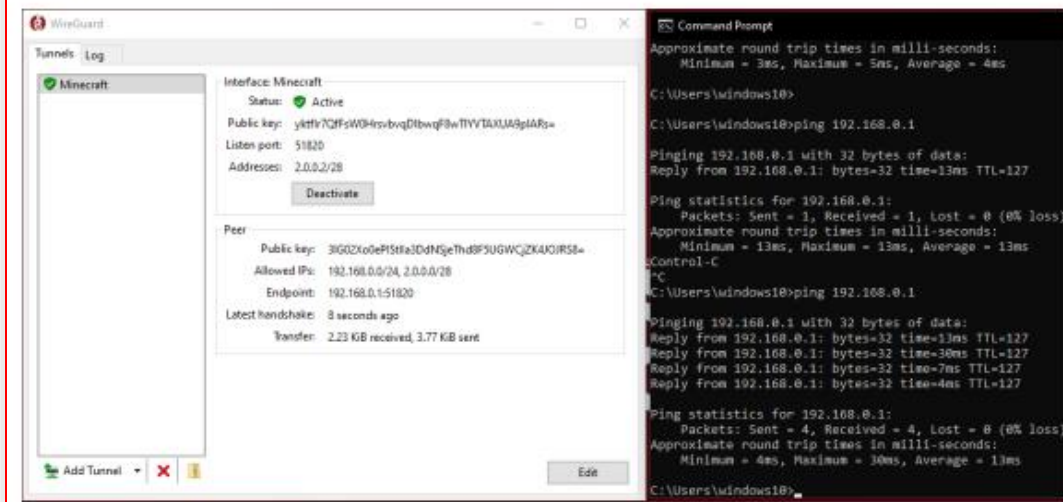
[Interface]
PrivateKey = YPm3ioFDPSwvcDNq6LeSIHPry92DnWN2BJ+2BvPheFY=
ListenPort = 51820
Address = 2.0.0.1/28

[Peer]
PublicKey = yktfIr7QfFsW0HrsbvqDIbwqF8wTIYVTAXUA9plARs=
AllowedIPs = 2.0.0.0/28
Endpoint = 192.168.1.1:51820

Save Cancel

Laboratório Realizado

Cliente:



The image shows the WireGuard Client application window and a Command Prompt. The WireGuard window displays the 'Minecraft' tunnel as active. The Command Prompt shows the results of a ping test to 192.168.0.1.

WireGuard Client Interface:

- Tunnels: Log
- Minecraft (Active)
- Interface: Minecraft
 - Status: Active
 - Public key: ykth7GffsW0HrsbvqDtbwqF0wTIVVTAUUA0pIA8=
 - Listen port: 51820
 - Addresses: 2.0.0.2/28
 - Deactivate
- Peer
 - Public key: 3K0Zx0eP1St0a3DdMgeThd8F9UGWQjZK4U0IR58=
 - Allowed IPs: 192.168.0.0/24, 2.0.0.0/28
 - Endpoint: 192.168.0.1:51820
 - Latest handshake: 8 seconds ago
 - Transfer: 2.23 KB received, 3.77 KB sent
- Add Tunnel
- Edit

Command Prompt:

```
Approximate round trip times in milli-seconds:
  Minimum = 3ms, Maximum = 5ms, Average = 4ms

C:\Users\windows10>
C:\Users\windows10>ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time=13ms TTL=127

Ping statistics for 192.168.0.1:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 13ms, Maximum = 13ms, Average = 13ms

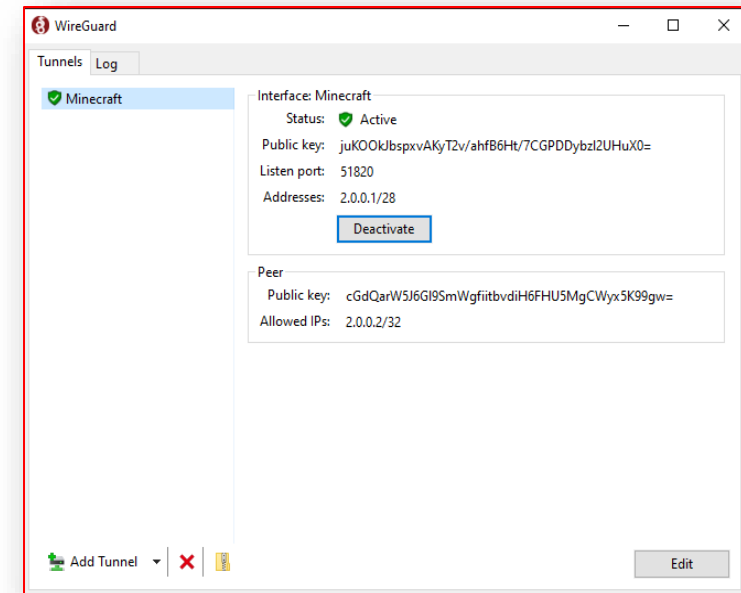
C:\Users\windows10>Control-C
^C
C:\Users\windows10>ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time=13ms TTL=127
Reply from 192.168.0.1: bytes=32 time=30ms TTL=127
Reply from 192.168.0.1: bytes=32 time=7ms TTL=127
Reply from 192.168.0.1: bytes=32 time=4ms TTL=127

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 30ms, Average = 13ms

C:\Users\windows10>
```

Servidor:

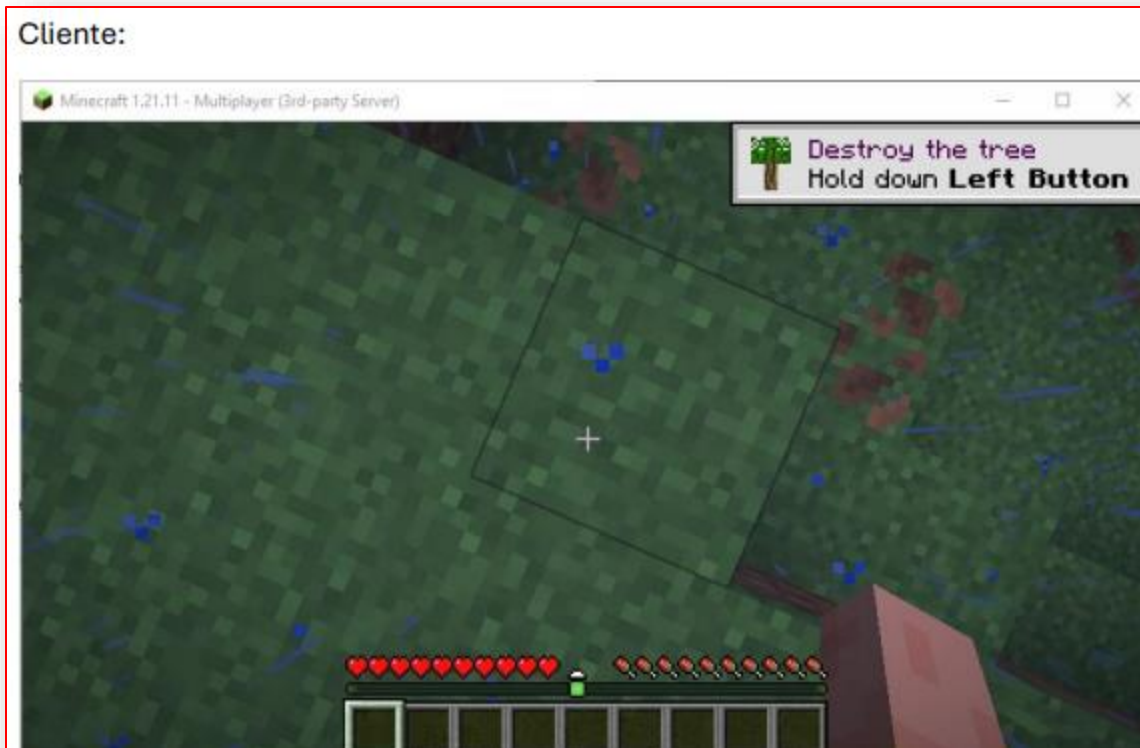


The image shows the WireGuard Server application window. The 'Minecraft' tunnel is listed as active. The interface displays the status, public key, listen port, and addresses for the tunnel.

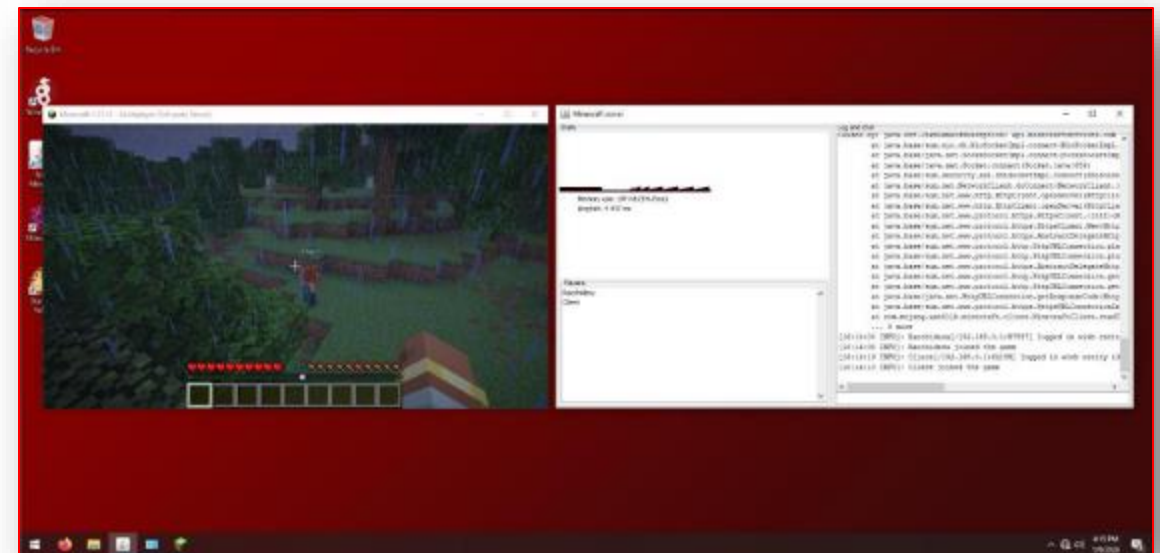
WireGuard Server Interface:

- Tunnels Log
- Minecraft (Active)
- Interface: Minecraft
 - Status: Active
 - Public key: juK0OKJbspvxAKyT2v/ahfB6Ht/7CGPDDybzI2UH-X0=
 - Listen port: 51820
 - Addresses: 2.0.0.1/28
 - Deactivate
- Peer
 - Public key: cGdQarW5J6G9SmWgfiitbvdIH6FHU5MgCWyx3K99gw=
 - Allowed IPs: 2.0.0.2/32
- Add Tunnel
- Edit

Laboratório Realizado



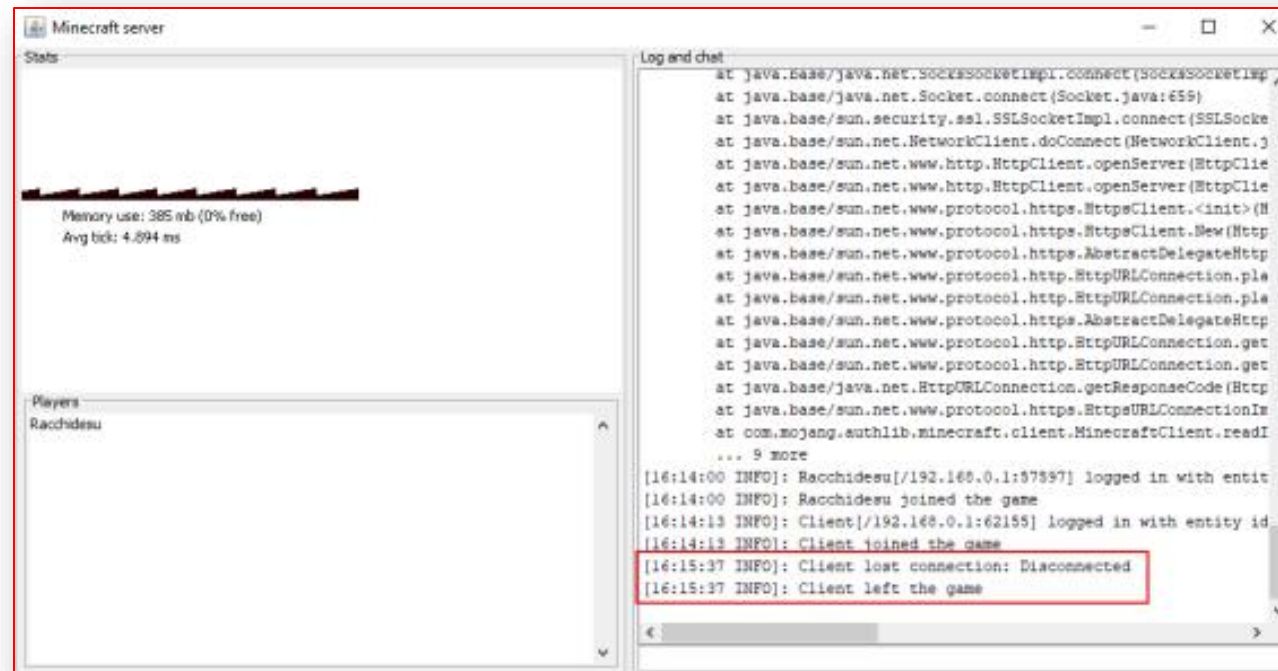
Servidor:Minecraft + Minecraft Sever Console



Laboratório Realizado

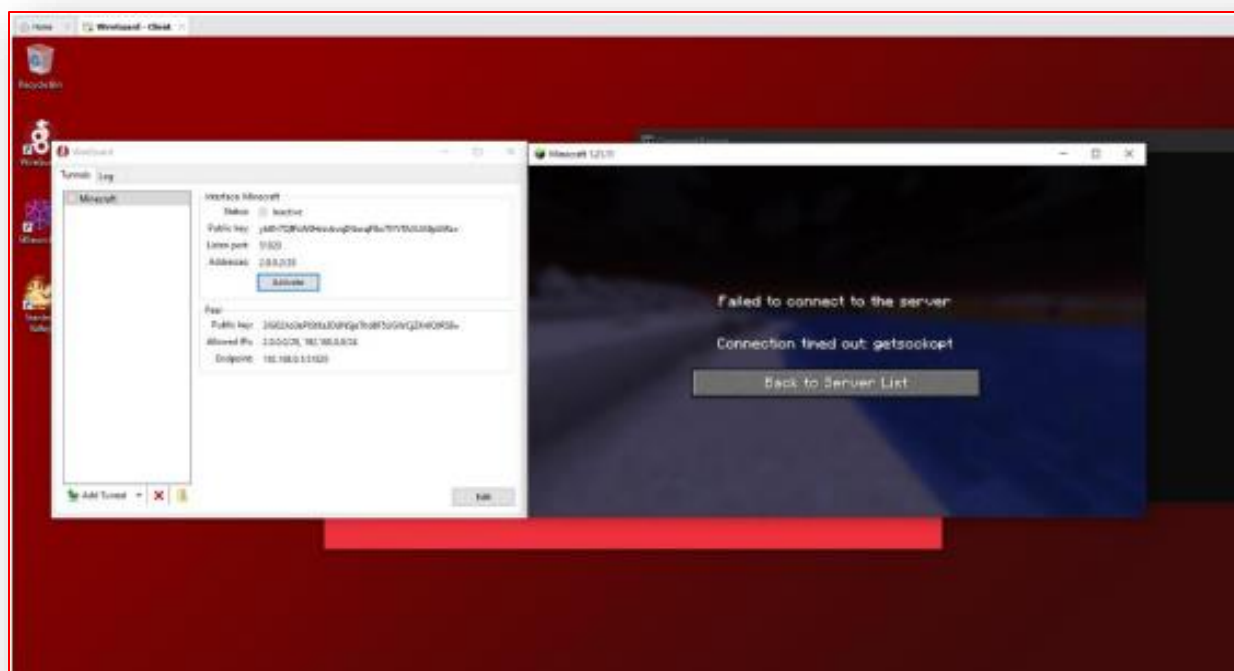
Desativamos o WireGuard no Cliente:

Servidor



Laboratório Realizado

Com o **WireGuard** desativado no **Cliente**, tentámos **aceder ao servidor** e vemos que **não conseguimos**:



Conclusão

- ▶ O **WireGuard** é uma solução **VPN** moderna.
- ▶ O trabalho permitiu compreender o seu **funcionamento, finalidade e vantagens** em relação a outras alternativas.
- ▶ O laboratório prático confirmou que o túnel garante **segurança e controlo de acesso**. A ligação ao servidor só era possível quando o **WireGuard** estava ativo.

QUIZ – O que é o WireGuard?

Kahoot!

Laboratório

► Passos para realização da tarefa:

- Instalar WireGuard numa OVA fornecida;
- Configurar o WireGuard do lado do Cliente;
- Testar Conexão ao Servidor Minecraft - [IP:192.168.0.1].

- Vão ser fornecidos os recursos necessários, assim como, o '**Guia de Laboratorio**' do WireGuard para a **realização do laboratório proposto**.

