HolePunch-UDPTunnel

An UDP tunnel system using NAT hole-punching to provide point-to-point tunnels between two clients across networks, through the use of an intermediatry information server for punching, then creating a VPN tunnel interface and forwarding all traffic through the hole-punched UDP ports.

Project Structure

The project consists of two linked golang projects, HolePunchUDPTunnel is the main project and it calls the executable produced by UDPTunnel:

- HolePunchUDPTunnel
  - Server mode: runs the hole-punch info exchange server.
  - Client mode: presents text user interface to select a client to connect to, performs hole-punch and launches the UDPTunnel.
- UDPTunnel
  - Runs a tunnel client/server with the given IP/ports, creates a vpn between clients when launched automatically by HolePunchUDPTunnel.

Directory Tree

Below is the directory tree of the project:

```
.
├── bin
│   └── HolePunch-UDPTunnel
├── holepunchudptunnel
│   ├── go.mod
│   ├── go.sum
│   ├── go.work
│   ├── main.go
│   ├── natholepunch
│   │   ├── clientdata.go
│   │   ├── client.go
│   │   ├── go.mod
│   │   └── server.go
│   ├── tui
│   │   ├── go.mod
│   │   ├── logging.go
│   │   └── ui.go
│   └── tunnelman
│       ├── go.mod
│       ├── tunneldata.go
│       ├── tunnelexec.go
│       ├── tunnelman.go
│       └── udptunnel
├── LICENSE
├── Makefile
├── README.md
```

```
└── udptunnel
    ├── filter.go
    ├── go.mod
    ├── go.sum
    ├── LICENSE.md
    ├── logger.go
    ├── main.go
    └── tunnel.go

7 directories, 27 files
```

Module Function Description

- ***HolePunchUDPTunnel***
  - Main program source directory, relies on compiled executable of support program UDPTunnel
  - **main.go**
    - Starts hole-punch server or hole-punch client plus text user interface (TUI) in client mode
  - ***natholepunch***
    - Sub-module for hole-punch server and client code
    - **server.go**
      - Code for running the hole-punch information exchange server
    - **client.go**
      - Code for running the hole-punch client and performing UDP hole-punching
    - **clientdata.go**
      - Defines structs that hold client data for both client and server
  - ***tui***
    - Sub-module for TUI code
    - **ui.go**
      - Code for displaying, rendering and updating the TUI with the information from the other sub-modules
    - **logging.go**
      - Code for redirecting StdOut output from other modules into a text view inside the TUI
  - ***tunnelman***
    - Sub-module for UDPTunnel manager code
    - **tunnelman.go**
      - Code for managing UDPTunnel configuration and creating UDPTunnel configuration based on hole-punch operation
    - **tunnelexec.go**
      - Code for executing the UDPTunnel program
      - The binary executable compiled from the UDPTunnel directory is embedded into our HolePunchUDPTunnel executable from this file.
    - **tunneldata.go**
      - Defines structs that hold UDPTunnel configuration data
- ***UDPTunnel***
  - Support program source directory, embbeded into and called by HolePunchUDPTunnel
  - **main.go**

        - Reads configuration file and launches a tunnel server/client
    - **tunnel.go**
        - Creates a new tunnel interface (such as tun0) and listens to trafic
        - Forwards raw packets to tunnel interface (local client -> UDPTunnel -> remote client)
        - Accepts raw packets from tunnel interface (local client <- UDPTunnel <- remote client)
    - **filter.go**
        - Security filter to only allow specified ports through the UDP tunnel
    - **logger.go**
        - Prints log and UDP tunnel traffic statistics

Count Lines of Code

Below is the result of counting the lines of code (generated by gocloc):

```
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
Go                              13            378            643           1789
Markdown                         3             36              0            176
Makefile                         1             11              8             34
-------------------------------------------------------------------------------
TOTAL                           17            425            651           1999
-------------------------------------------------------------------------------
```

Compilation

1. Begin by going to the project directory.
2. Compile the code by running `make all` (if compiling outside of China, change `prepare-cn` to `prepare` in `Makefile` line `12`).
3. Go to the `bin` directory by running `cd bin`.
4. The compiled project executable binary is found as `HolePunch-UDPTunnel` here.

Usage

Make sure to launch the executable from the same folder as the executable (i.e. `cd bin && ./HolePunch-UDPTunnel`) To use this project, at least three devices are required:

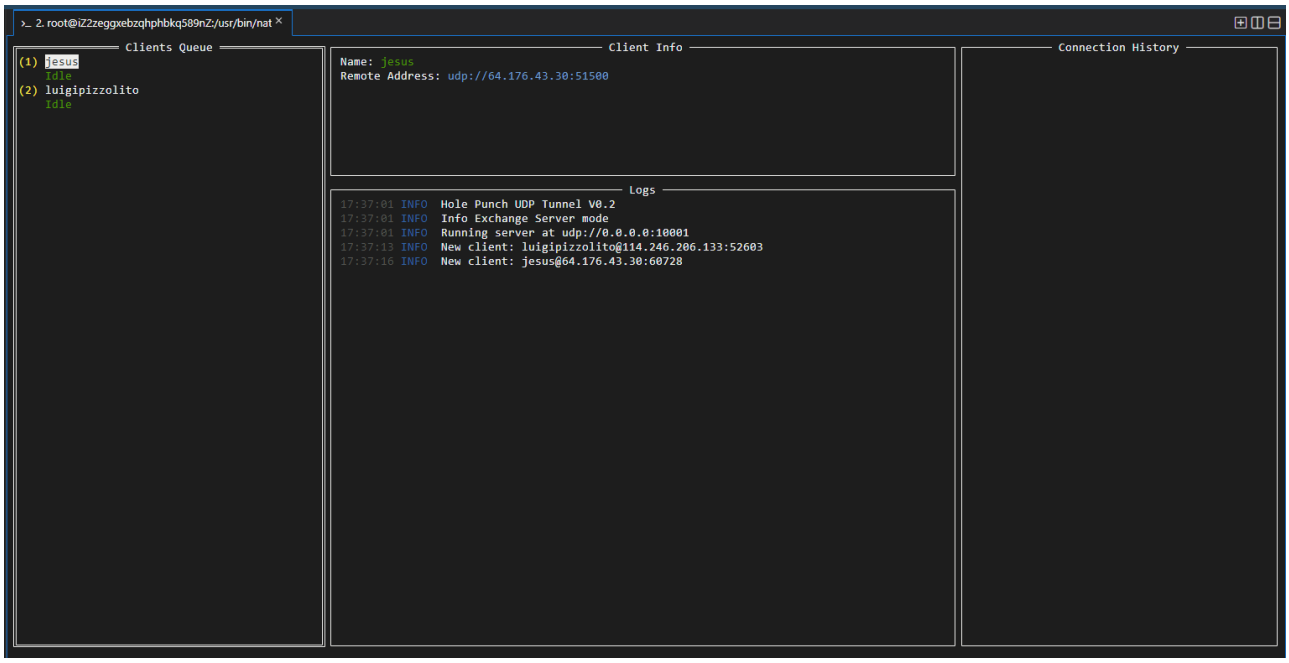
- Hole-punch info exchange server
    - Must be run on a server with a public IP
    - Run `./HolePunch-UDPTunnel --server`

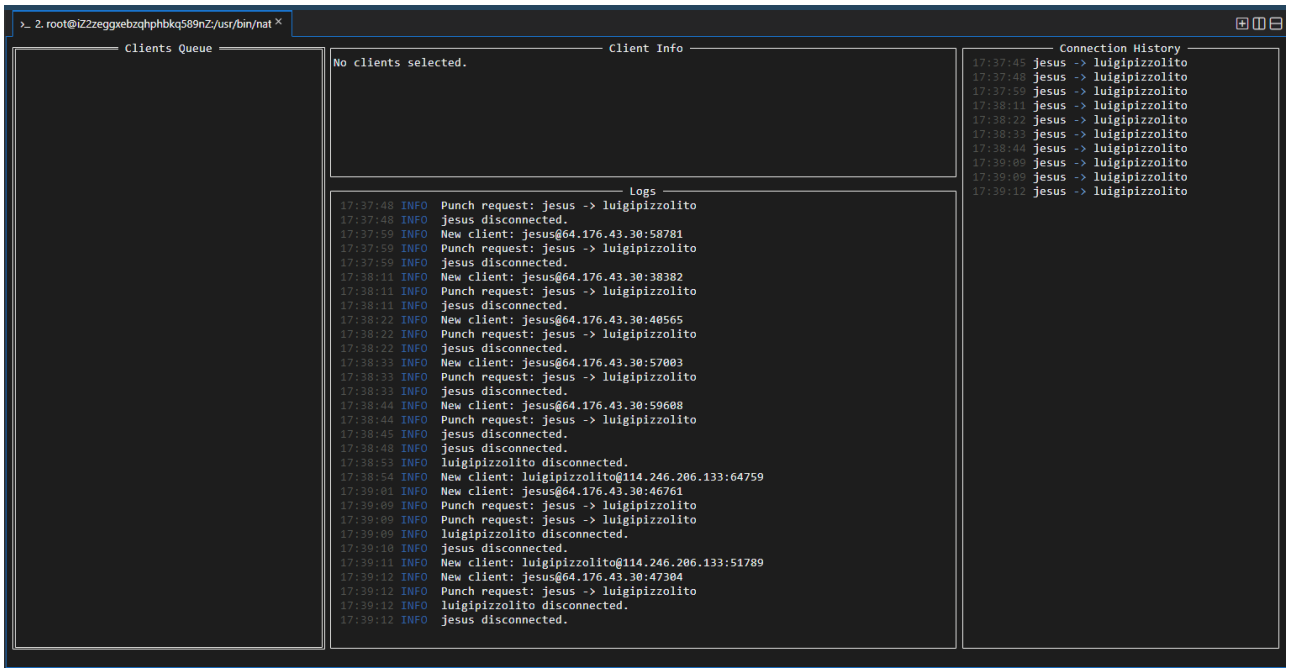
- Hole-punch client (2 or more)
  - Run on clients behind NATs
  - Shows text user interface (TUI) to user
  - Run `./HolePunch-UDPTunnel -a <Hole-Punch Server Public UP>`
    - replace `<Hole-Punch Server Public UP>` to the actual public IP of the server running the hole-punch info exchange.
    - Default username is taken to be the OS username
    - Other options available through `./HolePunch-UDPTunnel --help`:

```
Usage of bin/HolePunch-UDPTunnel:
  -a string
        Info exchange server IP address to connect to (default "127.0.0.1")
  -l string
        Specify Local side ID (default "luigipizzolito")
  -p string
        Info exchange server port to listen on or connect to (default
"10001")
  -r string
        (optional) Specify remote side ID to tunnel to
  -server
        Run in info exchange server mode (run this on a publicly accesible
IP)
  -t duration
        Specify reconnection timeout (default 2s)
```

Server TUI

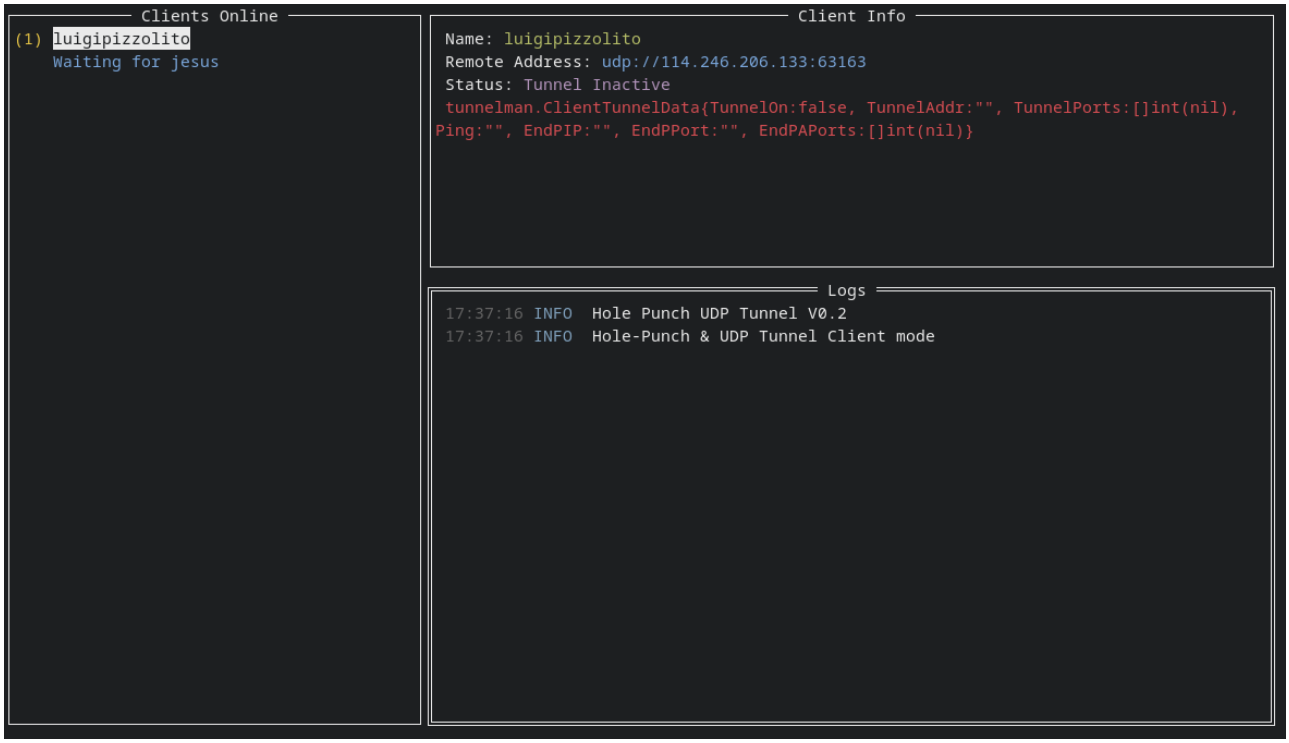
- After running a hole-punch server, the client info, logs and connection history can be seen:



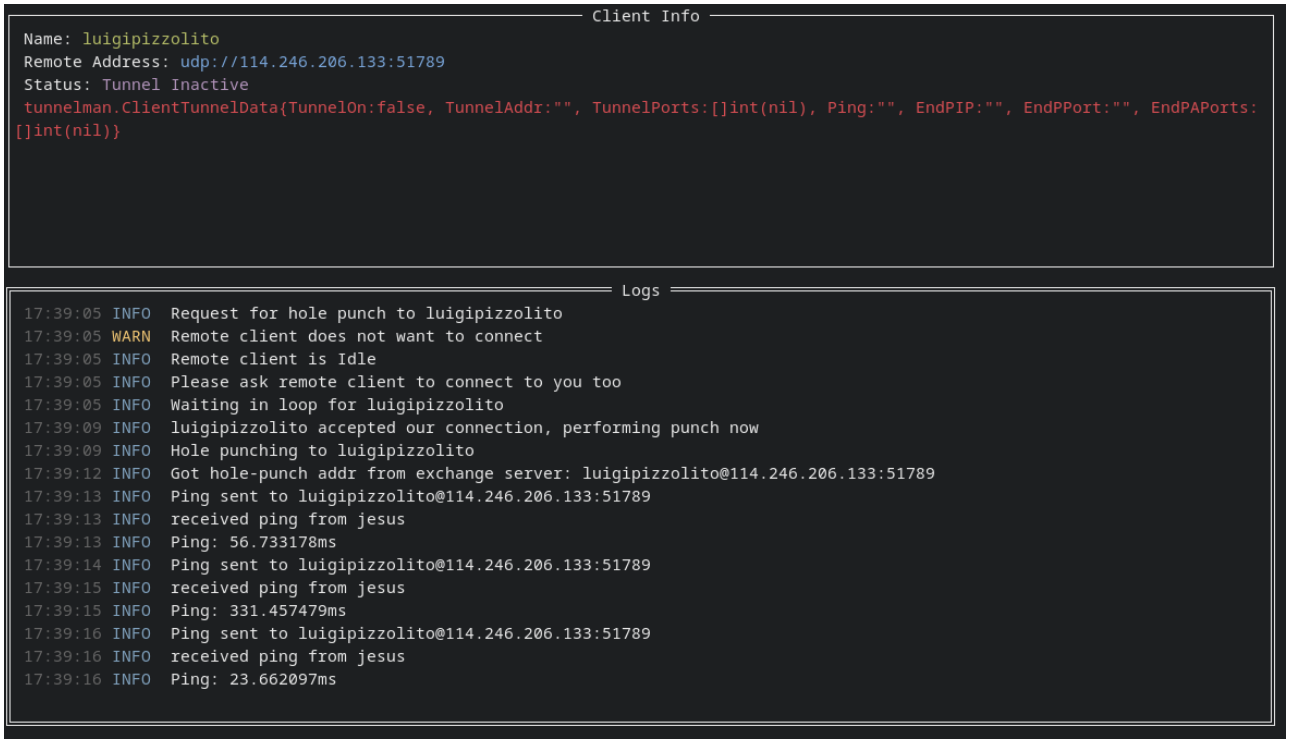


Client TUI

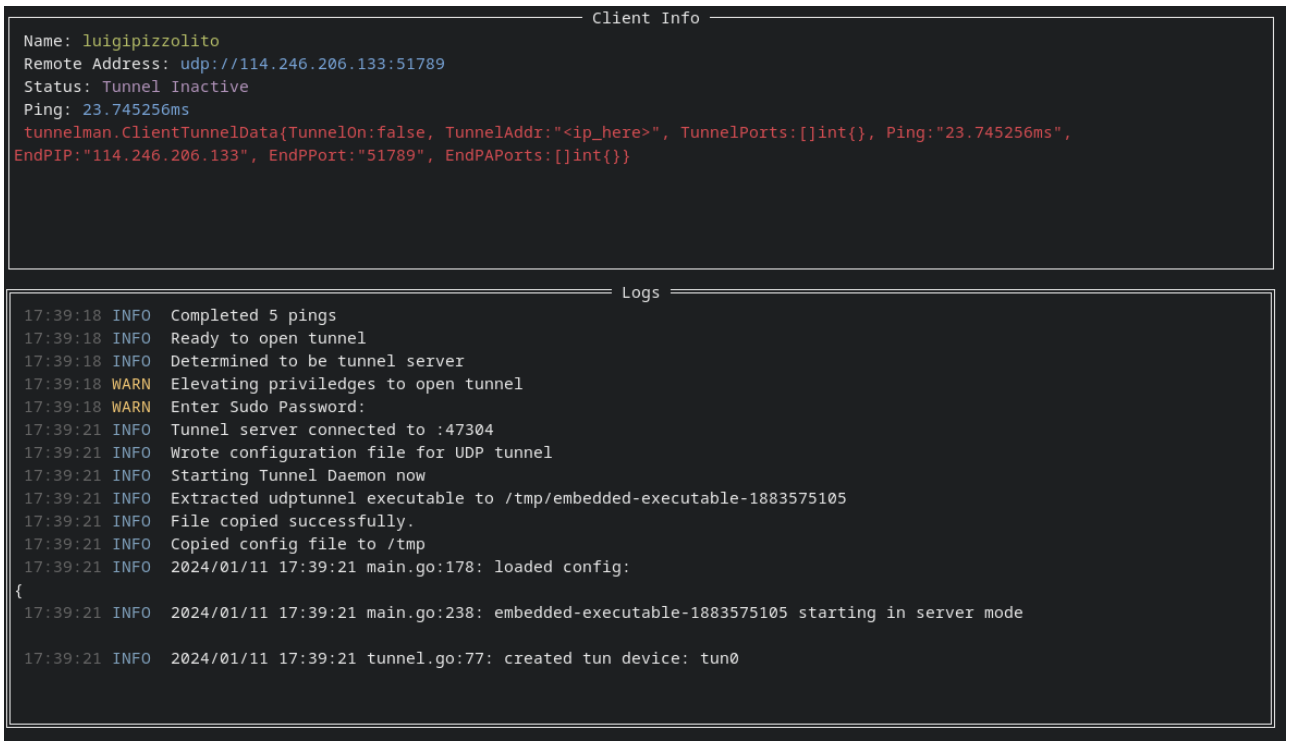
- After running a hole-punch client, the TUI can be seen:
- Click, press enter, or press a number key to select a client to connect to.



- The hole-punch operation will begin, and pings will be sent in between clients:



- After sucessful hole-punch, `sudo` password is prompted, enter sudo password to launch the UDP tunnel:



- After launching UDP tunnel, one client is assigned IP `10.0.0.1` and the other client is assigned IP `10.0.0.2`

- You may now access the other client with any application or protocol (including TCP) directly by using their IP.

- For example, SSHing into the other client: