Openport

Release 2.0

Jan De Bleser

Jun 13, 2022

CONTENTS

Openport is a free reverse ssh service. It allow you to connect to a port on your machine from the internet.

CHAPTER

ONE

CONTENTS

1.1 Usage

1.1.1 Installation

To use Openport, first download and install it:

```
$ wget https://openport.io/download/debian64/latest.deb
$ sudo dpkg -i latest.deb
```

Other platforms available from our Github page.

1.1.2 Sharing a port

Examples:

```
openport 22
openport 8080 --http-forward
openport 3389 --restart-on-reboot
```

1.1.3 Linking a key to your account

After creating an account, get your token from https://openport.io/user/keys

```
openport register-key <your-token>
```

After registration, your client and sessions will appear on the clients and sessions page.

Notes:

- If the client was already registered under a different account, it will be unregistered from that account
- Be careful when registering clients with sudo, in some cases this registers the wrong key because the $HOME variable is changed. (running openport as root is not recommended in any case).

1.1.4 Extra options

```
Usage: openport (<port> | forward | list | kill | kill-all | register-key | help )
Default: openport <port> [arguments]
  -d, --daemonize                     Start the app in the background.
      --exit-on-failure-timeout int   Specify in seconds if you want the app to exit if␣
↪it cannot properly connect. (default -1)
      --http-forward                  Request an http forward, so you can connect to␣
↪port 80 on the server.
      --ip-link-protection string     Let users click a secret link before they can␣
↪access this port. This overwrites the setting in your profile. choices=[True, False]
      --keep-alive int                The interval in between keep-alive messages in␣
↪seconds. (default 120)
      --port int                      The local port you want to expose. (default -1)
      --proxy string                  Socks5 proxy to use. Format: socks5://
↪user:pass@host:port
      --remote-port string            The remote port on the server. [openport.io:1234]␣
↪(default "-1")
  -R, --restart-on-reboot             Restart this session when 'restart-sessions' is␣
↪called (on boot for example).
  -v, --verbose                       Verbose logging
```

1.1.5 Files used

All log files and the local sqlite database are stored under the .openport folder in the user home (~/.openport).

1.1.6 Restarting after a reboot: Windows

In windows, only the user that installed openport shares can be restarted on reboot, and the option to install the service during installation must be checked.

1.1.7 Restarting after a reboot: Linux & MacOS

Openport will automatically restart the shares that the root user has started with –restart-on-reboot. If you want to start the shares from other users too, you'll need to create a file at /etc/openport/users.conf and list the users here (one per line).

1.2 Release notes

Download the client from Github

1.2.1 2.1.0: Latest

- Adding –exit-on-failure-timeout flag (Specify in seconds if you want the app to exit if it cannot properly connect. (default -1))
- Code refactoring

1.2.2 2.0.4

- Fixed issue with restarting sessions from a non-root user.
- Creating the /etc/openport/users.conf file at installation.

1.2.3 2.0.3

- Checking that both .ssh/id_rsa and .ssh/id_rsa.pub exist before using them
- Bugfix: server port was not reused after closing the app with ctrl-c

1.2.4 2.0.2

- Rewritten client in Go
- Updated commands: "openport –list" is now "openport list" Same for forward, list, kill, kill-all, register-key, version and help.
- Fixed issue with hanging clients (added timeout on http requests)
- Improved speed, size and memory consumption
- Terminology: replaced "share" with "session"

1.2.5 1.2.0

- Fixed ssl issue in ubuntu
- Python3 compatibility
- Dependency upgrades
- Added –keep-alive flag to modify the interval in between the keep-alive messages

1.2.6 1.1.0

- Added –daemonize flag
- Fixed issue with saving session without an internet connection
- Added open-for-ip link in –list output

1.2.7 1.0.2

- Moved openport to it's own package, so it can be installed as a regular python package.

1.2.8 1.0.1

- added –name to –register-key

1.2.9 1.0.0

- Added a Graphical User Interface (GUI)
- ip-link-protection can be switched on and off from the command line
- You can now create [forward tunnels](/wiki/recipes/create_a_forward_tunnel/)
- Many bugfixes
- Removed the manager
- Only one forward per port
- Can handle encrypted keys
- 64 bit version for ubuntu
- Better warnings and messages
- Better signal handling
- Show extra information with –list –verbose

1.2.10 0.9.1

- IP Link Protection
- Brute force protection
- Various bug fixes
- Cleaner output

1.2.11 0.8.0

- Port forwarding
- Http forwarding
- Restart on reboot

1.3 Build from Source

The openport client is fully open-source. You can find the source here

You need to have docker installed.

To compile the code, run

```
docker_compile_all.sh
```

Once that is done, you can run

```
./openport-<my-arch> <port>
```

1.4 Latency

Openport uses multiple servers, spread globally to minimize latency. Depending on the geographic location of the client device, it will be assigned a port on any of these servers:

- openport.io (Amsterdam, The Netherlands)
- us.openport.io (San Francisco, USA)
- spr.openport.io (Singapore, Singapore)

Contact us if you need an additional server in your region.

1.5 Recipes

Share your ssh port:

```
openport 22 --restart-on-reboot
```

Share your http port:

```
openport 8080 --http-forward -R
```

Share VNC:

```
openport 5900 -R
```

Share remote desktop:

```
openport 3389 -R
```

1.5.1 Automating the installation

In this guide, replace the xxxxxxxxx by the token you see at the bottom of this page: [https://openport.io/user/keys{]}(https://openport.io/user/keys)

Windows

You can start the installer with the "/S" flag to do a silent installation:

```
openport_installer.exe /S
```

Then you can register your key:

```
openport register-key --token=xxxxxxxxx --name="unique name"
```

Start your share:

```
openport --restart-on-reboot 8000
```

Done!

Linux

Install the .deb like you are used to:

```
sudo dpkg -i openport.deb
```

Register the key:

```
openport register-key --token=xxxxxxxxx --name=$HOSTNAME
```

Or you can use the MAC address:

```
mac_address=$(ifconfig eth0 | grep -o -E '([[:xdigit:]]{1,2}:){5}[[:xdigit:]]{1,2}')
openport register-key --token=xxxxxxxxx --name=$mac_address
```

Start your share:

```
openport 22 --restart-on-reboot
```

Done!

1.5.2 No outgoing traffic allowed on port 22

Consider next scenario:

I have a pc at home, that I want to contact from my work. Both locations are firewalled and do not allow outgoing connections to ports other than 80 (http) and 443 (https).

On my home pc: I use the commands

> openport register-key xxxxxxx # From your https://openport.io/user/keys page openport 22 –restart-on-reboot

Which gives me the output:

INFO - Now forwarding remote port openport.io:4104 to localhost:22.

On my work pc: I use the command

openport register-key xxxxxxxx # From your https://openport.io/user/keys page openport forward –remote-port 4104 -R

Which will give me the output:

INFO - You are now connected. You can access the remote pc's port 22 on localhost:58553

Now, if I want to connect to my home pc, I can use the command

ssh myhomeuser@localhost -p 58553

1.5.3 Forward to another server

If you have a device (let's call it machine A) that you cannot reach from your home (H), but you have another device (let's call it machine B) that does can reach machine A, you can forward your requests from H to B to A using these commands:

Let's say there is a webserver (port 80) running on A:

On B you run:

ssh -L 8000:<ip from A>:80 <user_on_B>@localhost openport 8000

That's it! Now you can access port 80 on A using the address the last command gave you.

1.5.4 Forwarding UDP

Openport uses reverse-ssh behind the scenes, so UDP forwarding is not (yet) supported out of the box. However, with some scripting you can achieve UDP forwarding.

This guide uses the command "socat" which is included in most Linux distributions. On windows you will need to use [Cygwin](https://cygwin.com/install.html).

Let's say you want to forward UDP port 1054 on machine A.

On Machine A: —

[Optional] If you want to start a UDP server to test:

Forward UDP traffic to a TCP port (6667 in this example):

Start openport to forward the TCP port to the outside world:

Example output:

Remember the port you got from openport (23607 in this example).

On Machine B: —

Here we listen on UDP port 2054 (but it can be the same port as on machine A if you want)

[Optional] If you want to test the forward:

1.5.5 Sharing a file

```
python -m SimpleHTTPServer 7999 &
openport 7999 --http-forward
```

1.5.6 Testing facebook buttons

When creating facebook like buttons for your page, facebook has to be able to reach you site. Deploying to a test site is slow, and you might have to do this multiple times, just for one button.

Using openport, you can have your development site online, so facebook can find your page, and read all the "og" tags it needs.

Simply use

```
openport 8080 --http-forward
```

And change the og:url tag to match the address you are given by openport.

```
<meta property="og:url" content="http://abcd.u.openport.io/" />
```

Then visit your site at the address that is given to you (abcd.u.openport.io in the example), and your facebook buttons will work.

1.5.7 Using openport with nodejs

Check out this project on GitHub to automatically launch openport when starting your gulp project.

1.5.8 Using the API

When you have a paying account, you will get access to the openport API. Using this API, you can request a list of your clients, and active sessions. Find the full OpenAPI specs here.

Example response:

```
GET https://openport.io/api/v2/sessions/?token=SECRET
{
    "count": 2,
    "next": null,
    "previous": null,
    "results": [
        {
            "key": "https://openport.io/api/v2/keys/15371/",
            "port": 6260,
            "local_port": 22,
            "server": "openport.io",
            "http_forwarding_address": null,
            "open_port_for_ip_link": "https://openport.io/l/6260/",
            "redirect_url": "https://openport.io/r/pdDrqgF7/22"
        },
        {
            "key": "https://openport.io/api/v2/keys/4642/",
```

(continues on next page)

(continued from previous page)

```
            "port": 40497,
            "local_port": 8081,
            "server": "openport.io",
            "http_forwarding_address": null,
            "open_port_for_ip_link": "https://www.openport.io/l/40497/",
            "redirect_url": "https://www.openport.io/r/LDnIs6MF/8081"
        }
    }
```

1.6 Known limitations

When forwarding a port with the "–http-forward" option, not all incoming http requests are supported. Only GET and POST. Websockets and other methods will not work with this option, but can be used without the –http-forward option.

1.7 Security

Opening a port from your PC to the internet might be dangerous, that's why Openport offers the following safety measures to keep you and your PC safe:

- *Click a link to open a port to your ip*
- *Brute force blocking*
- Http forwards are not transmitted directly to you machine, but are interpreted by the openport servers, and then forwarded. This means there is an extra layer of security between you and the outside world.
- Port scanning detection

See also:

1.7.1 Brute Force

By default, a client can only make 10 connections per minute to your machine. You can change this number by going to 'My Openport' > 'Clients' > 'Edit'.

You can disable this feature by putting '0' in the text box, or by using –http-forward.

1.7.2 IP link protection

Opening a port from your pc to the internet sounds dangerous, and it is. That's why openport has build-in protection, that requires a user to know a unique token when he wants to access your port.

Let's say you want to share your ssh connection, you will get something like

```
$ openport 22
INFO - Now forwarding remote port openport.io:12345 to localhost:22 .
You can keep track of your shares at https://openport.io/user .
You are now connected. You still can transfer 10000 megabyte this month.
Tell the one you are sending this link to, to first go here: https://openport.io/l/12345/
↪abscsqdfklmagsdgq . This will allow his IP to contact you.
```

As long as the user on the other end of the line does not know this link, he cannot reach you, and so you are safe from intruders. Try it yourself!

You can also find this link in 'My Openport'>'Sessions'.

You can disable this feature using the 'Edit' button in the 'clients' tab in 'My Openport', but this is not recommended.

1.8 FAQ

1.8.1 Enable remote access to your Mac

On your Mac, open Sharing preferences, then check 'Remote Login'.

You can now run

this will allow you to access your Mac from anywhere in the world.

1.8.2 Stopping a session

You can kill shares via the manager using:

```
openport manager --list
openport manager --kill <local port>
```

On linux and mac

Ctrl-c or kill -2 will stop the share and the share will not restart on reboot.

Ctrl-z or kill -9 will stop the share, but the share will restart if the –restart-on-reboot flag was passed.

On windows

Ctrl-c will stop the share and the share will not restart on reboot.

Ctrl-z or stopping via task manager will stop the share, but the share will restart if the –restart-on-reboot flag was passed.

1.8.3 Is it safe

Yes. Openport is the safest reverse ssh solution on the internet. Why? Several security measures have been put in place:

- Port scanning is detected and blocked
- users must click a link first, before the port is opened for their IP (You can change this in your account settings)
- users cannot make more than 10 connections per minute to your port (You can change this in your account settings)
- Http-forward links are interpreted and reconstructed by the openport servers first, which means you are not vulnerable, even with old/unpatched webservers on your end

See also *here*

1.8.4 Shares not restarting on CentOS

If you notice that shares are not restarting for users other than root on Red Hat or one of its derivatives (CentOs, Scientific Linux, ...), you should remove following line from /etc/sudoers

```
defaults requiretty
```

The reason is that openport will sudo into the user to restart its shares, but cannot because of this rule.

The line is completely save to remove, as is stated here.

1.8.5 Why open source

Because we believe that open source projects can help the world become a nicer and safer place.

You can find the code here.

1.9 Contacting support

We are happy to hear from you!

Contact us using the Chat box on the website, or open a ticket on our GitHub.

1.10 About Us

Openport was founded in 2014 by Jan De Bleser (jan at openport dot io).

Initially it was created to service a server behind a firewall using SSH, but soon we realized it can be used for any TCP protocol.

We are based in Antwerp (Belgium).