

MeshCentral²

Router User Guide



Version 0.0.1

May 7, 2019
Ylian Saint-Hilaire

Table of Contents

1.	Abstract	1
2.	Introduction	1
3.	Downloading	1
4.	Login	2
5.	Creating Port Maps	2
6.	Command Line Arguments	4
7.	Conclusion	5
8.	License	5

Document Changes

May 7, 2019 – 0.0.1
Initial version.

1. Abstract

This document takes a look at MeshCentral Router, a Windows application that performs TCP and UDP port mapping from a local machine to any remote computer thru a MeshCentral servers. This document should allow the user to Internet relay traffic thru NAT routers and firewalls.

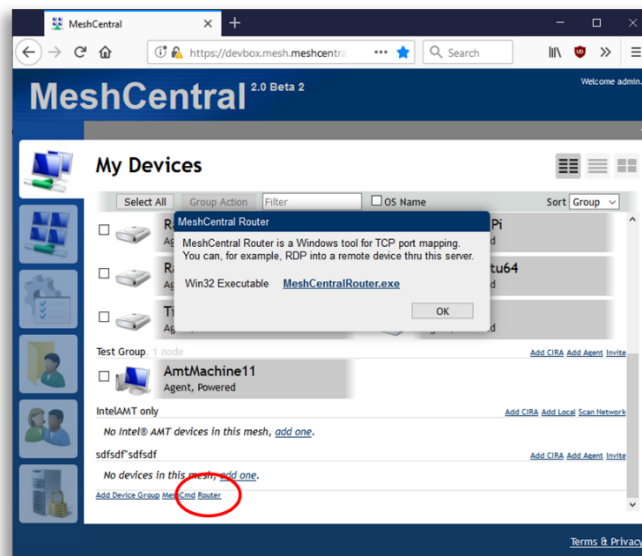
2. Introduction

MeshCentral is a remote management web site that connects users to remote computers over a local network or the internet. In addition to remote management features, MeshCentral is capable of relaying TCP and UDP traffic thru the server and thru mesh agents installed on remote computers. This feature can be useful to, for example, perform a remote desktop session using RDP or route custom traffic that would otherwise be difficult to route.

3. Downloading

MeshCentral router is a Windows application that comes built-into the MeshCentral server or can be downloaded at: <http://info.meshcentral.com/downloads/MeshCentral2/MeshCentralRouter.exe>

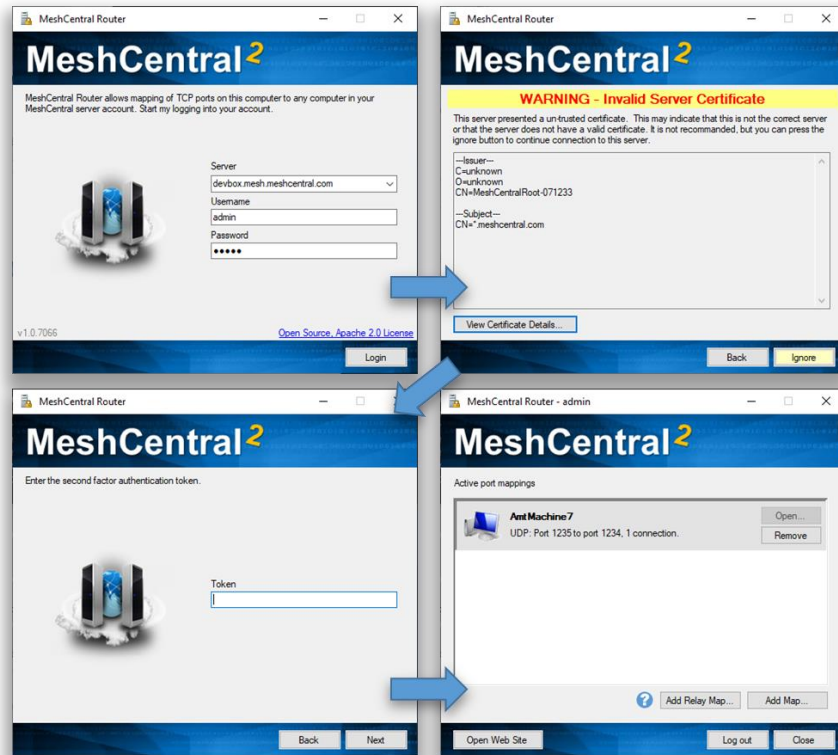
It's probably best to use the MeshCentral router that comes with your version of the MeshCentral server as the two will likely be most compatible. A given MeshCentral Router version may not work with any MeshCentral server versions. On MeshCentral, you can download MeshCentral Router with this link:



The router link will only show up on Windows browsers.

4. Login

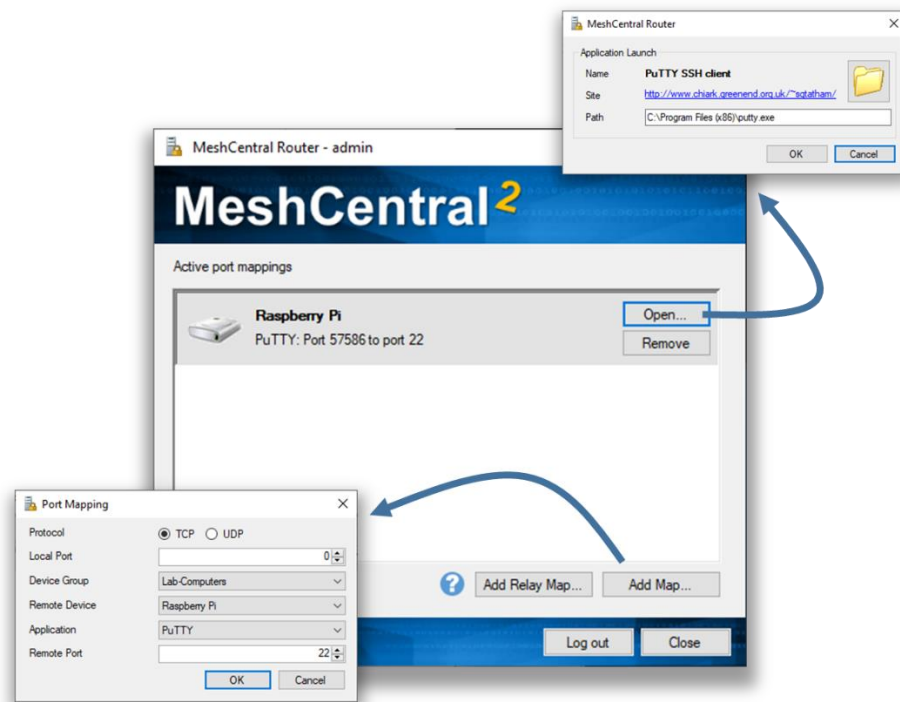
MeshCentral router will need to login to your MeshCentral server just like any browser. You can do this by entering the server name and account username and password. Depending on your server and account situation, you may see some or all of the following screens.



If the second factor authentication is required, MeshCentral Router does not support hardware keys (WebAuthn), but does support the YubiKey™ OTP.

5. Creating Port Maps

Once logged in, you can start adding port maps using the “Add Map...” and “Add Relay Map...” buttons on the bottom right. You can then create a map and open the associated application using the “Open...” button and remote to port map using the “Remove” button.



There are two different types of ports mappings. A normal port map and a relay port map. A normal port map will route packets to the selected destination computer that is running the mesh agent as shown here.



A relay port map will route traffic thru the server and thru the remote agent to a target IP address on the remote agent's network as shown here.



Note that all traffic is encrypted using TLS from MeshCentral Router to the MeshCentral server and from the server to the MeshAgent. The server and the agent do have access the traffic so it's recommended to use port mappings to tunnel data that is also encrypted for that end-to-end encryption is assured.

6. Command Line Arguments

MeshCentral router can be run with command line arguments to make it quicker and easier to use. The arguments range from debugging to being able to quickly login and setting up port maps.

`-debug`

Causes MeshCentral Router to generate a "debug.log" dump file that can be useful for debugging.

```
-host:<hostname>
-user:<username>
-pass:<password>
-ignorecert
```

This set of command line arguments make logging into the MeshCentral server easier. Note that specifying the password using a command line argument may not be secure as the command shell can record the password in the command history. The "ignorecert" argument is not recommended as it's going to cause MeshCentral Router to ignore untrusted server certificates. This should only be used for debugging.

```
-map:<protocol>:<localport>:<computername>:<app>:<remoteport>
```

The "map" argument will automatically create a network map once MeshCentral Router is logged In. The protocol must be "TCP" or "UDP, the local port can be 0 for any. The computer name is the server-side name of the computer, if many computers have the same one, one of them will be selected. The app can be empty for a custom application, or can be "HTTP", "HTTPS", "RDP", "PuTTY" or "WinSCP". For the UDP protocol, no apps are currently supported so it should be left blank. For example you can use:

```
-map:TCP:0:"MyComputer":HTTP:80
```

```
-map:UDP:1235:"MyComputer":1234
```

The first example will map a random local port to port 80 of "MyComputer" and is expected for use with HTTP traffic. The second example maps local UDP port 1235 to port 1234 on "MyComputer". It's best for the computer name to be in quotes.

In addition to port mapping, you can also setup relay maps where a remote computer is used as a traffic relay like this:

```
-relaymap:<protocol>:<localport>:<computername>  
:<app>:<remoteip>:<remoteport>
```

This will relays a local port to thru the server and thru a remote agent to a target IP address and port. For example:

```
-relaymap:TCP:555:"MyComputer":HTTP:192.168.1.1:80
```

This will relay local port 555 to a 192.168.1.1:80 for HTTP traffic. A typical use of this is to be able to remotely configure a home router from anywhere on the Internet.

```
-all
```

The "all" switch will bind local ports to all network interfaces so that other computers on the network can use the port maps provided by MeshCentral Router. By default, local ports will be bound to the loopback interface so that only local application can use the port mappings.

7. Conclusion

MeshCentral Router is a free, open source tool for routing TCP and UDP traffic over the internet. It's a powerful as it allows for any traffic to go thru NAT routers and firewalls while being encrypted using TLS.

8. License

MeshCentral, MeshCentral Router and this document are all opens source and licensed using Apache 2.0, the full license can be found at <https://www.apache.org/licenses/LICENSE-2.0>.