# What to do if Remote Desktop Connection can't connect to a remote computer

Krishna G

This post aims to discuss all the common reasons why a Remote Desktop Protocol (RDP) connection can't connect to a remote computer. I will explain how to identify the cause and then show you how to fix your failing Remote Desktop Connection.

There are many reasons why an RDP connection to a remote machine might fail. The screen below shows a typical error for a failed RDP connection.

"Remote Desktop can't connect to the remote computer for one of these reasons:"

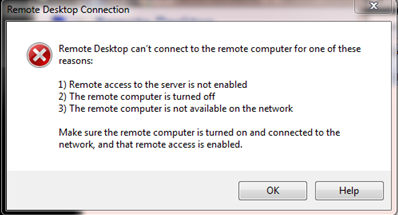


Figure : RDP connection failed

## Verify the network connectivity

Every admin should be familiar with this RDP error. The most common cause of a failing RDP connection concerns network connectivity issues, for instance, if a firewall is blocking access.

You can use ping, a Telnet client, and PsPing from your local machine to check the connectivity to the remote computer. Keep in mind ping won't work if ICMP is blocked on your network. The main advantage of Telnet and PsPing is that you can connect via TCP, and you can check whether the RDP port 3389 is open.

The Telnet client isn't enabled by default. Use this command to enable Telnet from a command prompt:

dism /online /Enable-Feature /FeatureName:TelnetClient

And use this one from a PowerShell console:

Install-WindowsFeature -name Telnet-Client

Use [PsPing](https://docs.microsoft.com/en-us/sysinternals/downloads/psping) if you have problems enabling the Telnet client. PsPing also lets you test connectivity to a specific TCP port. It is portable, so no installation is required.

First, try to ping the remote computer's hostname or IP address.

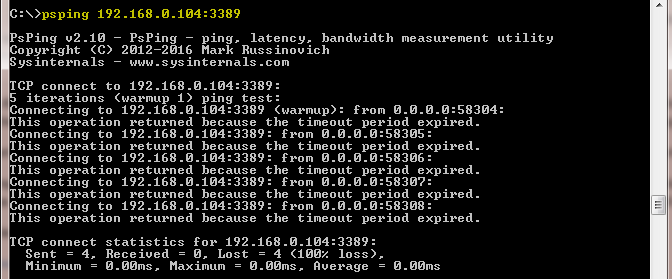


Figure 2: The remote machine connection timed out with PsPing

As see you can see in the screenshot above, I was unable to ping the remote machine, and the port was not reachable as well.

If this works, and you are unable to ping the machine using the FQDN name to check whether DNS resolution is working properly, sometimes the hostname is pointing to another machine on DNS that is either offline or not in use.

If you can't connect at all, a local firewall (Windows Firewall or third-party security software) or a network firewall might be blocking the port. The PowerShell command below lets you display the Windows Firewall state on the remote machine.

Invoke-Command -ComputerName [ComputerName] -ScriptBlock {netsh advfirewall show allprofiles}

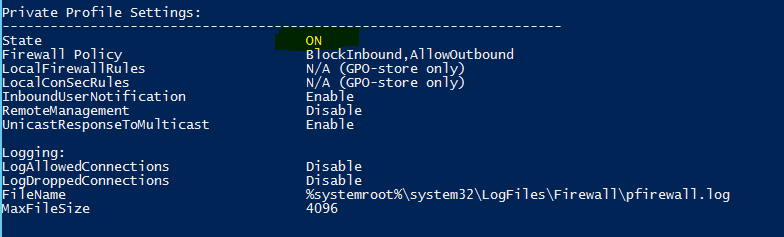


Figure 3: Remote computer firewall status

For testing purposes, you can disable Windows Firewall on the remote computer with this command:

Invoke-Command -ComputerName Win7 -ScriptBlock {netsh advfirewall set allprofiles state off}

Note that you should enable PSRemoting on the remote computer to execute the above command. If not, you can use [PsExec](https://docs.microsoft.com/en-us/sysinternals/downloads/psexec) to [enable PowerShell remoting](https://4sysops.com/wiki/enable-powershell-remoting/) with the command below:

psexec \\RemoteComputer -u administrator -p PASSWORD netsh advfirewall set allprofiles state off

## Verify user permissions

If your user account has no administrator privileges, you should be a member of the local Remote Desktop Users group to access the remote machine via RDP. By default, no members are in this group, and only members of the Administrators group can connect via RDP.

Read this 4sysops article to learn how to [add users remotely to a user group](https://4sysops.com/archives/add-a-user-to-the-local-administrators-group-on-a-remote-computer/).

## Allow Remote Desktop Connection

Ensure Remote Desktop is enabled on the remote computer. The RDP listener could be inactive. You can enable Remote Desktop Connection either from System Properties or from the registry.

Option 1: Select **Start > Run**, type **sysdm.cpl**, and select the **Remote** tab.

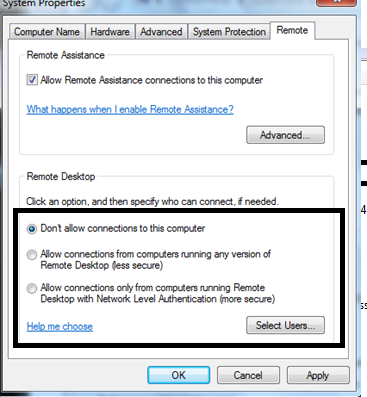


Figure 4: Remote computer RDP settings

Option 2: Select **Start > Run**, type **regedit**, navigate to HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server, and set the value for the key **fDenyTSConnections** to 0 (0 = Enable; 1 = Disable).

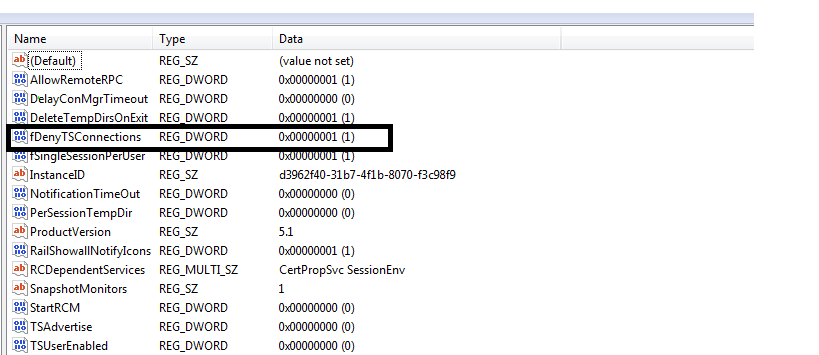


Figure 5: Remote computer RDP settings in the registry

You can use this PowerShell command to enable RDP remotely:

(Get-WmiObject Win32\_TerminalServiceSetting -Computername [ComputerName] ‑Namespace root\cimv2\TerminalServices).SetAllowTsConnections(1,1)

And from the command prompt, you can use the next command if the Remote Registry service is running on the remote computer:

REG ADD "\\[RemoteComputer] \HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /d 0 /f /t REG\_DWORD

## Verify the status of the RDP services

On both the local (client) computer and the remote (target) computer, the following services should be running:

* Remote Desktop Services (TermService)
* Remote Desktop Services UserMode Port Redirector (UmRdpService)

The UmRdpService is an RDP port redirector service, which helps redirect drives, printers, and ports from the local to the remote machine. For example, if you want to map all of your local drivers to the remote computer, this service will do the job.

If the UmRdpService service was set to disabled through a central Group Policy, RDP connections to this machine will fail. Note that sometimes restarting the service won't fix the issue, and you have to reboot the machine after reconfiguring the Startup Type to Automatic.

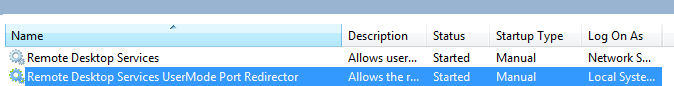


Figure 6: Remote computer RDP services status

The PowerShell command below starts both of these services remotely if they are in a stopped state. Note that this only works if the service Startup Type is set to either Automatic or Manual.

"TermService","UmRdpService" | ForEach-Object{ (Get-WmiObject Win32\_service -ComputerName [RemoteComputer] -Filter "Name = '$\_' ").StartService() }

The output of the command should be either 0 (started) or 10 (already running). Check out this [article](https://docs.microsoft.com/en-us/windows/win32/cimwin32prov/startservice-method-in-class-win32-service) to learn more about return codes and their descriptions.

## Identify whether Group Policy is blocking RDP

You can enable or disable Remote Desktop centrally through Group Policy settings. To check those settings, go to **Start > Run**, type **gpedit.msc**, navigate to **Computer Configuration\Administrative Templates\Windows Components\Remote Desktop Services\Remote Desktop Session Host\Connections**, and find the **Allow users to connect remotely by using Remote Desktop Services** setting. If the setting is **Disabled**, you should change it to **Enabled** or **Not Configured**.

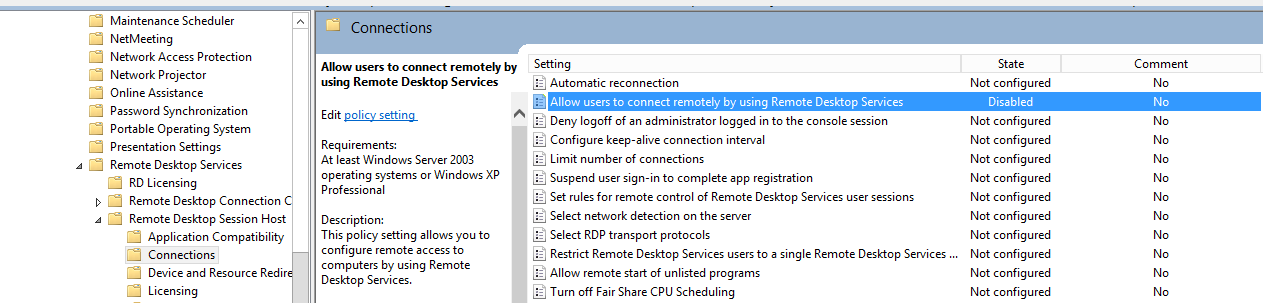


Figure 7: RDP settings from Group Policy

Use GPResult (gpresult /h C:\output.htm) from a console on the remote machine to verify whether Group Policy has been applied properly. Also you can use rsop.msc to get the applied Group Policy settings on a particular machine.

## Check the RDP listener port on the remote computer

By default, the RDP client verifies that the Remote Desktop service on the remote computer is listening on port 3389. If not, another application could be occupying the same port.

To check whether any remote session (RDP-TCP) already exists on that computer, use qwinsta, which gives you a list of local as well as remote sessions.



Figure 8: Using qwinsta to list sessions

The screenshot above shows that the rdp-tcp session with session ID 65536 already exists.

To verify that the Remote Desktop service is using the correct port, use the Registry Editor. Go to **Start > Run**, type **regedit**, navigate to HKEY\_LOCAL\_MACHINE > SYSTEM > CurrentControlSet > Control > Terminal Server > WinStations > RDP-Tcp, and review the PortNumber setting.

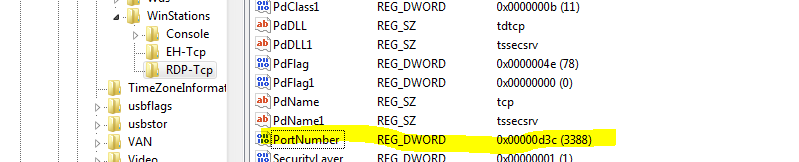


Figure 9: RDP port setting from the registry

Alternatively, you can use the command below:

REG QUERY "\\[Remote Computer]\HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /F "PortNumber"

If the output of the RDP port value is 0x00000d3d (hex), your RDP port is configured with a default port, which is 3389. In the screenshot above, the default RDP port was changed to 3388. In this case, either you have to change the RDP port to the default one, or you access the remote machine via the new port 3388.

In the Remote Desktop client, you have to specify the custom RDP port in the computer address space as shown in below:

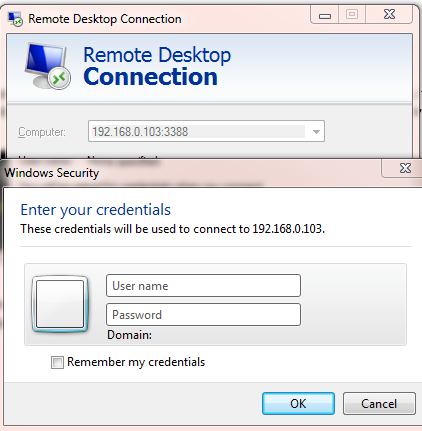


Figure 10: RDP access with a different port

If another application is using the RDP port, you have to find that application on the remote machine and then reconfigure it to use a port other than 3389. Use the netstat command to find the application PID listening on port 3389. And with the tasklist command, you can identify the name of the application running with this PID as shown below:

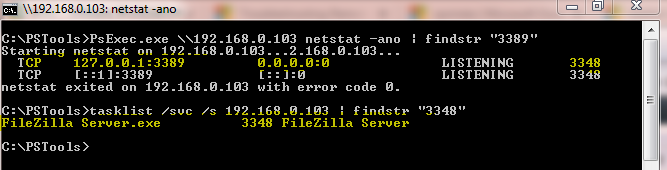


Figure 11: Check whether another process is using the RDP port

## Conclusion

Many articles discuss Remote Desktop connection problems. I wrote this one mainly to compile all possible causes of failed RDP connections. If you know of another possible cause, please post a comment below.