

---

# Kousec Server Certificate Manager

## Configuring PowerShell Remoting for non ActiveDirectory Environments

---

Copyright 2010 Kousec Software, Inc. All rights reserved.

Kousec and Kousec Server Certificate Manager are trademarks of Kousec Software, Inc.  
All company names and product names are trademarks of their respective holders.

R6

---

---

## Table of Contents

1. Introduction.....	4
1.1. Key Concept in PowerShell Remoting Network Security .....	4
1.2. Applicable Product Versions .....	5
1.3. 32bit and 64bit Versions of PowerShell .....	6
2. Really Quick Configuration in Lab Environment .....	7
3. Enable PowerShell Remoting only on Server Side .....	7
4. Quick Configuration in Lab Environment .....	7
5. Secure Configuration in Production Environment.....	9
5.1. Prepare Server Certificate for WinRM .....	9
5.2. Creating WinRM HTTPS Listener.....	10
5.3. Connecting from Powershell Remoting Client.....	10
5.4. Renewing WinRM Certificate.....	12
6. Summary .....	14

---

## 1. Introduction

Kousec Server Certificate Manager uses the PowerShell V2 Remoting feature as one of the mechanisms for automating certificate installations on Windows Server computers. In the current version of Kousec Server Certificate Manager, PowerShell Remoting is used to install server certificates for IIS and other Microsoft products. In future, PowerShell Remoting will be used to automate certificate installations for other server products that are running on Windows Server computers. This will include many Java-based servers, Apache HTTP servers and other commercial application servers. In short, PowerShell Remoting will be used as Windows equivalent of SSH, which is used on Linux/Unix computers.

This document explains the procedure to enable PowerShell Remoting. This document focuses on an IT environment where managing servers (Kousec Server Certificate manager computers) and managed servers (computers that need server certificate, such as IIS) are not in the same Active Directory domain. We especially focus on a situation where managing server and managed server are in totally unrelated workgroups or AD domains.

In this document, Kousec Server Certificate Manager will be referred to as either “Kousec CertMgr” or just “CertMgr”.

References:

<http://blogs.msdn.com/wmi/archive/2009/07/24/powershell-remoting-between-two-workgroup-machines.aspx>

### 1.1. Key Concept in PowerShell Remoting Network Security

#### Need for Server Authentication

In setting up PowerShell Remoting among client and server computers, the “server authentication” is the key concept that determines how you will set up the WinRM listener, the network component that underlies PowerShell Remoting.

#### Server and Client in an ActiveDirectory Domain

If the server computer (that accepts PowerShell commands from a remote computer) and the client computer (that sends PowerShell commands to a remote computer) are members of an ActiveDirectory domain, the client authentication that will happen when starting a

---

---

PowerShell session will use the Kerberos authentication system. In this authentication scheme, mutual authentication (both client and server are authenticated) is carried out.

### **Server and Client in Unrelated AD Domains or Standalone**

If the server computer and the client computer are in different AD domains or are standalone computers, the client authentication for PowerShell session will use NTLM authentication scheme. In this authentication scheme, the server's identity is not authenticated.

### **Then, How Servers can be Authenticated?**

There are two approaches.

1. Assume that connect destinations (i.e., servers) are trusted.  
You should only use this method in a highly secured network for the production environment. For lab environments, this is a quick and easy method. We describe the procedure for this under *"Quick Configuration in Lab Environment"* section.
2. Use HTTPS and valid server certificate for server authentication.  
You should use this method for most production environments. The key point here is to use a digital certificate issued from a trusted CA. Depending on your deployment, you will either obtain a certificate from a commercial CA or another private CA who is trusted by both client and server computers.  
We describe the procedure for this under *"Secure Configuration in Production Environment"* section.

### **1.2. Applicable Product Versions**

Kousec Server Certificate Manager RC-3a or later

Windows Server 2008 R2

Windows Server 2008 + Windows Management Framework (Powershell 2.0, WinRM 2.0)

Windows Server 2003 + Windows Management Framework (Powershell 2.0, WinRM 2.0)

Windows XP SP3 + Windows Management Framework (Powershell 2.0, WinRM 2.0)

Windows Management Framework (Powershell 2.0, WinRM 2.0) can be obtained from the following location

<http://support.microsoft.com/?kbid=968930>

---

---

### 1.3. 32bit and 64bit Versions of PowerShell

On Windows x64 systems, there are two versions or modes of PowerShell, one that runs in 32-bit environment and another that runs in 64-bit environment. When you type “powershell” on a 64-bit Windows OS, the 64-bit version of PowerShell session starts.

32-bit and 64-bit PowerShells run scripts in separate PowerShell execution environments. For example, Script Execution Policy is set separately.

Since Kousec Server Certificate Manager is a 32-bit application, its PowerShell scripts run in 32-bit environment. Therefore when you set up PowerShell Remoting, you need to start 32-bit PowerShell.

Calling 32-bit version of PowerShell

```
> C:\WINDOWS\system32\windowspowershell\v1.0\powershell.exe
```

Calling 64-bit version of PowerShell

```
> powershell
```

---

## 2. Really Quick Configuration in Lab Environment

### Client Computer (on which CertMgr is installed)

Start a PowerShell window and enter the following.

```
PS> Set-ExecutionPolicy RemoteSigned
PS> set-item WSMAN:\localhost\Client\TrustedHosts -value *
```

If you are on Windows Vista or Windows Server 2008, also enter the following.

```
PS> Set-ItemProperty -Path
HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System -Name
LocalAccountTokenFilterPolicy -Value 1
```

### Server Computers (servers managed by CertMgr)

```
PS> Enable-PSRemoting
```

## 3. Enable PowerShell Remoting only on Server Side

You need to enable Powershell Remoting only on computers that accept commands from remote computers (i.e., remoting servers). For computers that will send commands to remote computers (i.e., remoting clients), there is no need to enable PowerShell Remoting. If you already enabled PS remoting on your client computers, you can disable it.

1. PS> Disable-PSRemoting
2. Follow instructions printed from Disable-PSRemoting command to completely restore the original state. This includes stopping WinRM and removing firewall exceptions for WinRM.

## 4. Quick Configuration in Lab Environment

Client computer: 192.168.1.20

Server computer 1: 192.168.1.101 (referenced by its IP address)

---

---

Server computer 2: server-2 (referenced by its server name)

## Client Computer

Start a PowerShell window and enter the following.

```
PS> Set-ExecutionPolicy RemoteSigned
```

Next, you need to specify the list of WinRM servers that you don't require server authentication.

(PowerShell Remoting does depend on WinRM for client-side configuration information. When running the following command, Windows may ask if it can start WinRM service. You can stop WinRM service afterwards.)

```
PS> set-item WSMan:\localhost\Client\TrustedHosts -value "192.168.1.101, server-2"
```

It is important that you add to this list each server that you want to have a server certificate installed by CertMgr. You can use the following command to add one server to the list.

```
PS>set-item WSMan:\localhost\Client\TrustedHosts -value server-3 -Concatenate
```

You can also use a wildcard (\*) in place of server IP addresses, meaning any remote computer.

```
PS> set-item WSMan:\localhost\Client\TrustedHosts -value *
```

or, limiting to computers within the company's domain.

```
PS> set-item WSMan:\localhost\Client\TrustedHosts -value *.example.com
```

## Server Computers

```
PS> Enable-PSRemoting
```

This will configure the computer to accept PowerShell Remoting commands from remote

---

---

computers.

## 5. Secure Configuration in Production Environment

Using an HTTP channel for WinRM is not very secure, especially when you add managed servers into the WinRM TrustedHosts as done earlier in this document. In this section we describe steps necessary to enable HTTPS WinRM channels.

### 5.1. Prepare Server Certificate for WinRM

You need to obtain a server certificate and install it on the server computer for WinRM.

Server certificate requirements:

- Common Name in the certificate must match the computer name of the WinRM server computer. If the computer has a domain name set, the common name must match the full computer name (FQDN), i.e., computer name + domain name.
- Extended Key Usage in the certificate must indicate Web Server Authentication.
- The certificate must be signed by a CA that the local computer trusts.
- The certificate must reside in MY certificate store of the local computer.
- There is no other non-expired certificate having the same common name in MY certificate store. If you have such certificates lying around, delete them using MMC certificate snap-in.

You can use Kousec CertMgr to easily create one and install it.

1. Create a certificate definition with the following spec:  
Certificate Common Name: **server-1.example.com**  
Subject Alt Names: **server-1.example.com, server-1, 192.168.1.101**  
Server Software: **IIS (generic)**
  2. Choose the **Built-in Private CA** as the certificate provider.  
You can also obtain one from a commercial CA or directly from other private CAs that you own
  3. Create a certificate install package and copy it to the WinRM server.  
Also write down the package password that's generated.  
Execute the package on the WinRM server. It will prompt for the package password
-

---

and then proceed to install the server certificate and key, along with necessary CA certificates.

## 5.2. Creating WinRM HTTPS Listener

By running the following command, you will create a WinRM HTTPS listener on port 5896.

```
CMD>winrm quickconfig -transport:HTTPS
```

If this command fails with a vague error, you can try to explicitly create an HTTPS listener with the following command. If there is a problem in creating an HTTPS listener, it may show more helpful errors than quickconfig.

```
CMD>winrm create winrm/config/listener?Address=*+Transport=HTTPS
```

Additionally, you can manually associate the HTTPS listener with a specific server certificate.

```
CMD>winrm create winrm/config/listener?Address=*+Transport=HTTPS
@{Hostname="<hostname>";CertificateThumbprint="<cert-thumb-print>"}
```

Where,

<hostname> is the common name set in the certificate, and

<cert-thumb-print> is a hexadecimal representation of the certificate hash value.

To check the WinRM listener settings:

```
CMD>winrm enumerate winrm/config/listener
```

## 5.3. Connecting from Powershell Remoting Client

Specify **-UseSSL** option in PS Remoting commands.

```
PS> enter-psession -computername server-1 -credential administrator -usessl
```

### Requirements on client computers

1. The CA that issued the WinRM server certificate must be trusted by client computers.
2. The client computer must access the server using the common name or one of the names in the Subject Alt Names attribute in the server certificate.
3. The client computer must be able to access the issuing CA's certificate revocation data.

If the server certificate is issued from a private CA, ensure that the private CA is trusted by client computers.

If the server computer can be accessed with both the computer name without domain name and the full computer name with domain name attached, you may want to use a multi-domain certificate having both names in the Subject Alt Names attribute, with the common name set to the full computer name (i.e., FQDN).

Example

Server-1 can be accessed with either "server-1", "server-1.example.com", or "192.168.1.111".

Certificate Common Name: **server-1.example.com**

Subject Alt Names: **server-1.example.com, server-1, 192.168.1.101**

If the server certificate contains the Certificate Distribution Point (CDP) attribute, WinRM client tries to access the URL specified in CDP to check whether the certificate is revoked or not. Ensure that the URL in CDP is accessible from client computers.

### Relaxing the Requirements

The third requirement (access to CA's CRL data) can be dropped in many situations.

The second requirement (client accesses the server with correct name) should only be dropped if you are unable to obtain a multi-domain certificate for some reason.

Dropping the first requirement (trusted CA) defeats the purpose of SSL server authentication so it should be kept.

PowerShell Sessions provides the following session options with regard to SSL server

---

---

authentication, corresponding to each requirement in the above.

- **-SkipCACheck** : Do not validate signing CA's certificate
- **-SkipCNCheck**: Do not check if the certificate common name matches the name used by client when connecting the server.
- **-SkipRevocation**: Do not validate revocation status of the certificate.

The following is an example with SkipCNCheck and SkipRevocation options.

```
PS> enter-psession -computername server-1 -credential administrator -uessl -SessionOption  
(New-PSSessionOption -SkipRevocation -SkipCNCheck)
```

#### 5.4. Renewing WinRM Certificate

When the current certificate nears its expiration date, obtain the next certificate.

Installing the certificate consists of two steps, importing the certificate to the Windows certificate store and binding WinRM listener with the imported certificate.

##### Importing Certificate into Certificate Store

Use one-click certificate installer (i.e., certificate install package) or automated certificate installer to import the new certificate into the MY certificate store of the local computer.

##### Binding WinRM Listener with Imported Certificate

This step needs to be done while logging in locally or through Remote Desktop. You cannot use PowerShell Remoting to logging in to this machine while you reconfigure WinRM.

First delete the HTTPS listener.

```
CMD>winrm delete winrm/config/listener?Address=*+Transport=HTTPS
```

Then create a new HTTPS listener, specifying the thumbprint (hash value) of the new certificate.

```
CMD>winrm create winrm/config/listener?Address=*+Transport=HTTPS  
@{Hostname="<hostname>";CertificateThumbprint="<cert-thumb-print>"}
```

---

You can find the certificate thumb print on Kousec Certmgr as SHA1 hash. You can also use MMC certificate snap-in to look it up.

### **Deleting Old Certificate from Certificate Store**

This is an optional step.

If you delete the old certificate from the certificate store, re-creating the WinRM HTTPS listener will be easier. You can just enter “winrm quickconfig -transport:HTTPS” and winrm command will pick up the correct certificate.

Please note that the old certificate may still be used by other services like IIS or Remote Desktop Services. Those services also need to switch to the new certificate, but until then you should not delete the old one.

---

## 6. Summary

In this document, we have explained basic procedures for enabling PowerShell Remoting for use with Kousec Server Certificate Manager. PowerShell Remoting allows Kousec Server Certificate Manager to automate deployment of server certificates to remote servers in a secure way.

PowerShell Remoting also allows many management software products to remotely manage servers even over the public network in a secure way. Critical to the network security of PowerShell Remoting in public networks is SSL/TLS server authentication infrastructure, which can be efficiently and cost effectively administered by Kousec Server Certificate Manager.

For the details of Kousec Server Certificate Manager, please visit Kousec Software website at <http://www.kousec.com/>.