NetWitness® Platform

Microsoft WinRM Configuration

NETWITNESS
Platform

Contact Information

NetWitness Community at https://community.netwitness.com contains a knowledge base that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

License Agreement

Third-Party Licenses

Note on Encryption Technologies

Distribution

Miscellaneous

November, 2024

Contents

Introduction

This document provides information for configuring the Windows WinRM (Windows Remote Management) service to allow NetWitness to collect event logs from Microsoft Windows machines. In this document, the word "Collector" refers to either the NetWitness Log Collector or the Virtual Log Collector. The word “Channel” refers to a Windows event log, for example, Security, System, Forwarded Event, or DNS.

This topic also documents the requirements and permissions to collect events and SIDs (Security Identifiers displayed in the events which can be translated to human user and group names by NetWitness from a system using a non-administrative account. NetWitness recommends using a non-administrator account for the collection user. You can perform the steps to create these permissions manually on each target system, or you can use a Group Policy Object (GPO). You can also use an NetWitness-supplied PowerShell script to accomplish these tasks either manually on each Windows system or as a logon script via a GPO to apply the same configuration across a broad number of systems. This is described in this document in Group Policy Method for Mass-enabling WinRM.

Note: NetWitness recommends that you test the script first by running it manually on a test machine in a lab to observe what it does, before pushing it out on a large scale across multiple systems via GPO.

You can use the PowerShell script (**winrmconfig**) to:

- Automate all of the steps to create a WinRM Listener that accepts requests from a collector
- Enable system access to the Security event log (in order to read Security Event logs via WinRM)
- Create user permissions, as well as other features

You can download the script from NetWitness Link here: https://community.netwitness.com/s/article/RSANetWitnessWinRMEventSourceTroubleshooting-PowerShellScript.

When to Use the winrmconfig Script

The winrmconfig script can be used in the following ways:

- Troubleshoot collection issues with **report** mode. For information, see Troubleshoot Collection Issues in Report Mode
- Configuration tool to enable HTTP or HTTPS listeners, enable security log access, create user permissions to access WMI remotely and access the WinRM WMI plugin (for SID enumeration). For information, see winrmconfig as a Configuration Tool
- Push the script via GPO to multiple systems. For information, see Group Policy Method for Mass-enabling WinRM

Limitations on Collection

The Collector uses Kerberos to retrieve a Ticket Granting Ticket (TGT) for WinRM Log Collection when the authentication type Negotiate selected in the Event Category. Kerberos stores only ***one*** Ticket Granting Ticket in its cache per domain. So, creating multiple Event Categories for the same domain with different collection accounts would mean that only one account's TGT would be used to collect; therefore, there would be no valid reason to do this.

You can use either 1) a single account to collect from all domains, or 2) one account per domain. In the case of a single account, the collector would use that configured Domain Controller (DC) where that account belongs to try to acquire Service Tickets (ST) to access all systems in all domains being collected from. The drawback of using this approach is that the DC would have to know about all those systems, which would require at least one-way trusts between the other domains and the DC.

Having one account for each domain is preferable, as the DC for each domain would likely know all systems within its own domain, and be readily able to provide an ST for each. In this case (where a Kerberos Realm is configured per domain and an account belonging to that domain is available) the collector will directly request ST's from the DC's in each domain. This configuration has the following advantages:

- If one of the collection user accounts gets locked out, all collection is not down.
- From a manageability point of view, a category per domain with a domain user from that domain would seem to fit most environments better than one account having access to all domains.

Note the following:

- If Windows DNS is to be used to resolve DC's and Systems being collected from, then ALL Domain controllers configured in Kerberos Realm are required to have an entry in the `/etc/resolv.conf` file. A common mistake is that only the first DC is entered: in this case, Kerberos is forced to operate as if there is only one category with one account, because it only has one path—the original DC. A common side effect is that ST's are not returned for certain domains.
- When you use Kerberos tools (such as **klist** or **kdestroy**) on the collector for troubleshooting, always use the `-A` option. The `-A` option denotes operation on ALL domains, rather than just the primary domain. Collection can be compromised if you use `klist -a` (note the lowercase "a") when you have multiple domains configured: this command will only return tokes for one domain (usually the first one that was configured).
- Before you use these tools you MUST execute the following command in the SSH session:

  ```
  export KRB5CCNAME=DIR:/var/netwitness/logcollector/runtime/krb5_ccache_dir
  ```

  Otherwise, there will be issues with how TGTs and STs are stored in the Kerberos cache. For example, if you ran **kinit** (used to test user credentials) without first running the above command, Kerberos would only store tokens for the primary domain configured. As a result, it would seem like collection is supported from only one domain at a time.

Troubleshoot Collection Issues in Report Mode

You can use the script in **report** mode to troubleshoot issues with WinRM collection. **report** mode is described in detail in later in this document in Report Mode. (More troubleshooting details can be found in Test and Troubleshoot Microsoft WinRM, which is located on NetWitness Link, in the Event Source Configuration Guides space).

winrmconfig as a Configuration Tool

When you use the script as a configuration tool, you have all the capabilities of report mode plus the ability to configure WinRM and to make updates to fix issues.

Enable HTTP or HTTPS Listeners

Because WinRM is a web-based protocol, you can create a listener for HTTP, HTTPS, or both protocols. (Although it makes little sense to use both, if HTTPS is required, all clients, such as NetWitness, should use that protocol). HTTP is simple, with only one real option, port selection, which is by default 5985 for HTTP, and 5986 for HTTPS. WinRM listens on these ports for all interfaces. An HTTPS listener requires a certificate, much like a web server serving HTTPS pages does. This certificate must have sufficient Enhanced Key Usage (EKU) bits enabled (it must have at least the Server Authentication EKU; for more information, see https://support.microsoft.com/en-us/kb/2019527). This is needed to allow WinRM to create the listener. Also, the certificate's **Subject** field must be the FQDN of the system. In WinRM terms, a viable certificate is said to be bound to the listener port that is selected. In **enable** mode, the **winrmconfig.ps1** script can create either HTTP or HTTPS listeners (provided a viable certificate is found).

Enable Access to the Security Event Log

In **enable** mode, the script will automatically enable the local Windows Network Service account to access the Security Event log. This is an important step because Windows Security Events are probably the most critical data that NetWitness collects. The Windows Remote Management service in the services panel uses the Network Service account by default. You can check this by right-clicking the WinRM service, selecting **Properties**, and clicking the **Logon** tab. The script modifies the Access Control List (ACL) for the security event log and adds the account to it.

Permissions to Access the Windows WMI Subsystem Remotely

In **enable** mode, the script configures permissions to access the Windows WMI subsystem remotely, which is the equivalent of using the **wmimgmt** Windows application to enable remote access to the WMI root/CIM namespace (see Test and Troubleshoot Microsoft WinRM on NetWitness Link for more details). The script uses the Windows Security API to create an Access Control Entry (ACE) for the collection user in WMI with remote access permissions, which allows remote collection of events.

Permissions to Access the WinRM WMI Plugin

In **enable** mode, the script configures permissions to access the WinRM WMI plugin, which is the equivalent of running the Windows **winrm configsddl WMI** command to enable read access to the WinRM WMI plugin. Without this permission, the user account cannot be used to enumerate SIDs, and there would be no translation of SID strings to user/group names within certain event log message types, which are useful in reports and investigations.

Add User as a Member of Event Log Readers Group

In order to read events from event logs, the Collection user must be part of the following groups:

- The Domain-level Event Log Readers group if collecting from a domain controller.
- The local Event Log Readers group if collecting from a non-domain controller member server or a standalone machine (not a part of a domain).

The script adds the user to the correct Event Log Readers group, regardless of the type of machine it is running on (if the user has not already been added).

Network Service Account to Access the Security Event Log

As stated above, collecting security event logs is a key source of event data for NetWitness. By default, the WinRM service which runs as the **Network Service** account does not have access to the security event logs. The script uses the **wevtutil** Windows command to grant Network Service access (see Test and Troubleshoot Microsoft WinRM on NetWitness Link for more details).

Running the winrmconfig Script on a Single System

The **winrmconfig.ps1** PowerShell script can be run with either of the following syntaxes

```
Powershell -command "c:\temp\winrmconfig.ps1 -Action report -User
mycollectionuser@mydomain.com"
```

Notice the double quotes around the command and arguments and the full path to the script location.

```
Powershell -File winrmconfig.ps1 -Action report -User
mycollectionuser@mydomain.com
```

Note: Notice that there are no double quotes around the command and arguments and no requirement for a full path to the script location.

winrmconfig Modes

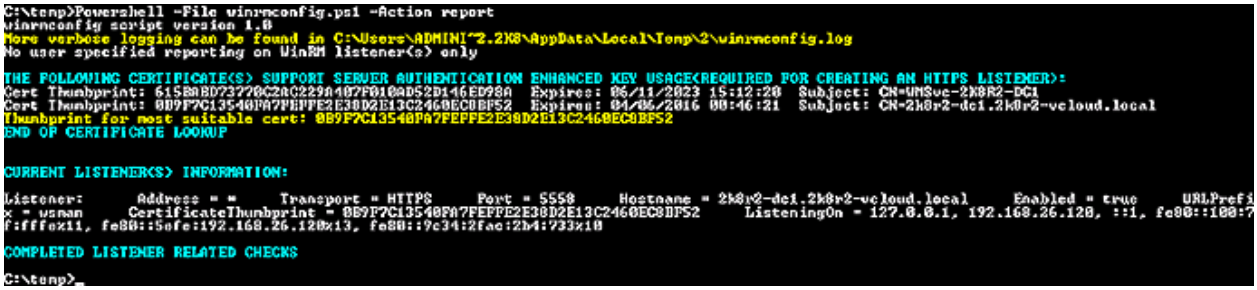
winrmconfig operates in two modes, **report** mode (used mainly for troubleshooting) and **enable** mode. The difference between the two modes is that **report** mode makes no changes. It runs a list of checks and reports back from each check with text in green (good), yellow (warnings), and red (errors).

In **enable** mode, the script attempts to fix any issue that could prevent collection and queries for SIDs, whether there is a listener-related issue if the **-ListenerType** command is specified on the command line, or an issue with user permissions, if **-User** is specified.

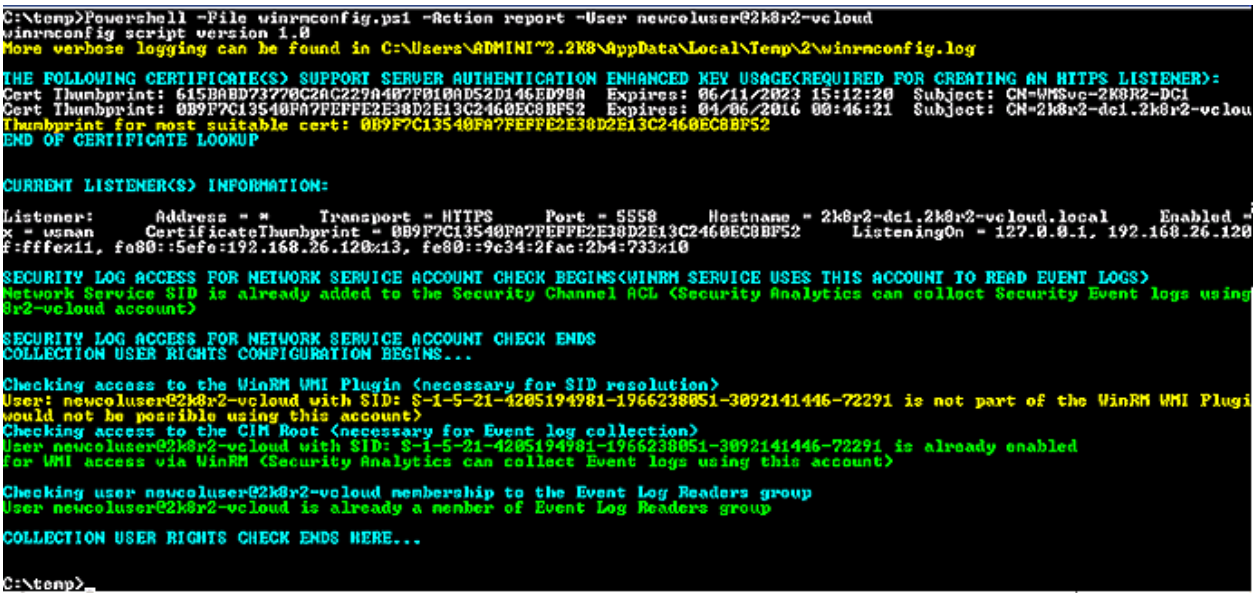
To select a mode, use the **-Action** command, for example:
`-Action report`
or
`-Action enable`.

Report Mode

This mode returns the listener states via the **-Action report** command, as shown below:



The report above returns information on the available certificates that might be used when creating an HTTPS listener, with a recommendation on which to choose, and also information on any currently configured listener(s). Adding the collection user account as shown below using the **-User** parameter yields information on the collection user account.



This output displays the following information:

1. Certificate information (including the recommended certificate for the HTTPS listener).
2. Current listener information (necessary for the collector to access the service remotely).
3. Current access to the security log for the Network Service account.
4. Access to the WinRM WMI plugin (required for SID enumeration). Note the warning that the user is not in the plugin's ACL list.
5. Current access to the WMI root (for event log collection). Note that it states in green (for good) that the user is already in that ACL.
6. Finally, a check is made to verify if the user is a member of the domain or local Event Log Readers group, whichever applies to the current system.

If all responses are green, everything is good. Event collection and SID enumeration are configured and are ready to use.

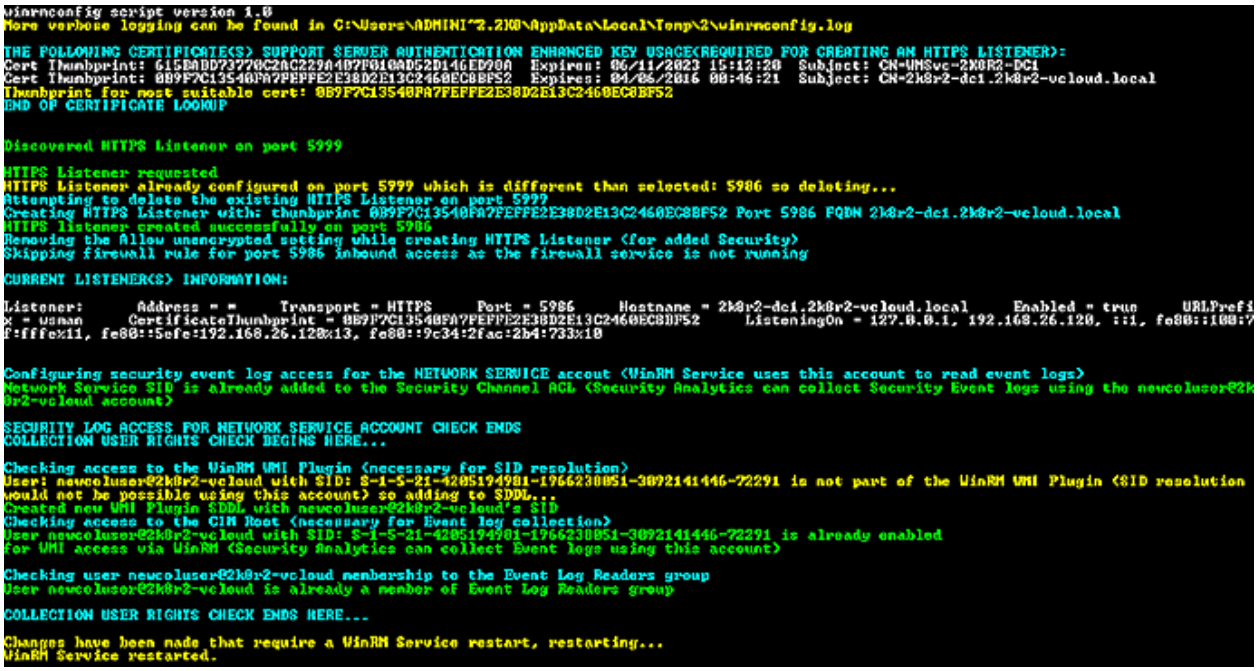
Enable Mode

In the **-Action enable** mode, the script can correct issues automatically or create a new configuration (for example, a new listener). In this mode, the steps are the same as for the **report** mode (it has the same features as **report** mode, and it makes updates to fix any issues).

As in **report** mode, a user can be omitted (see number 4 above), so the script only checks and enables a listener based on the arguments that are passed in.

The example below shows the results of running the following command:

```
Powershell -File winrmconfig.ps1 -Action enable -ListenerType https -
User newcoluser@2k8r2-vcloud
```



Now that the script is in **enable** mode, an HTTP, or in this case an HTTPS listener, can be created. A **-Port** option was not passed to the script, so the default port number of 5986 is assumed for HTTPS. The script first confirms that the certificate is correct, and then deletes the old listener and creates the new one on the requested port.

The script continues by checking access to the Security Event log, and detects that the Network Service account is already added to the ACL for the Security Event log.

The script then checks access to the WMI WinRM plugin. Note that the example shows text in yellow, which is warning-level and states that NetWitness would not be able to collect SIDs. In the following few lines, the script corrects this, and in green text (for good results) states that it created a new WMI Plugin SDDL. Note that this action requires a restart of the **winrm** service, which it does at the end.

Next, the script checks the WMI Remote privileges for the Collection user. This step is critical in collecting events remotely. Without it, the NetWitness Collector could not collect event logs. In this case, the check results in a green message, stating that this step was already completed.

Finally, the script enables user permissions to read the events by adding the user to the Event Log Readers group. The script does this whether the system is a standalone machine, a domain member, or a domain controller.

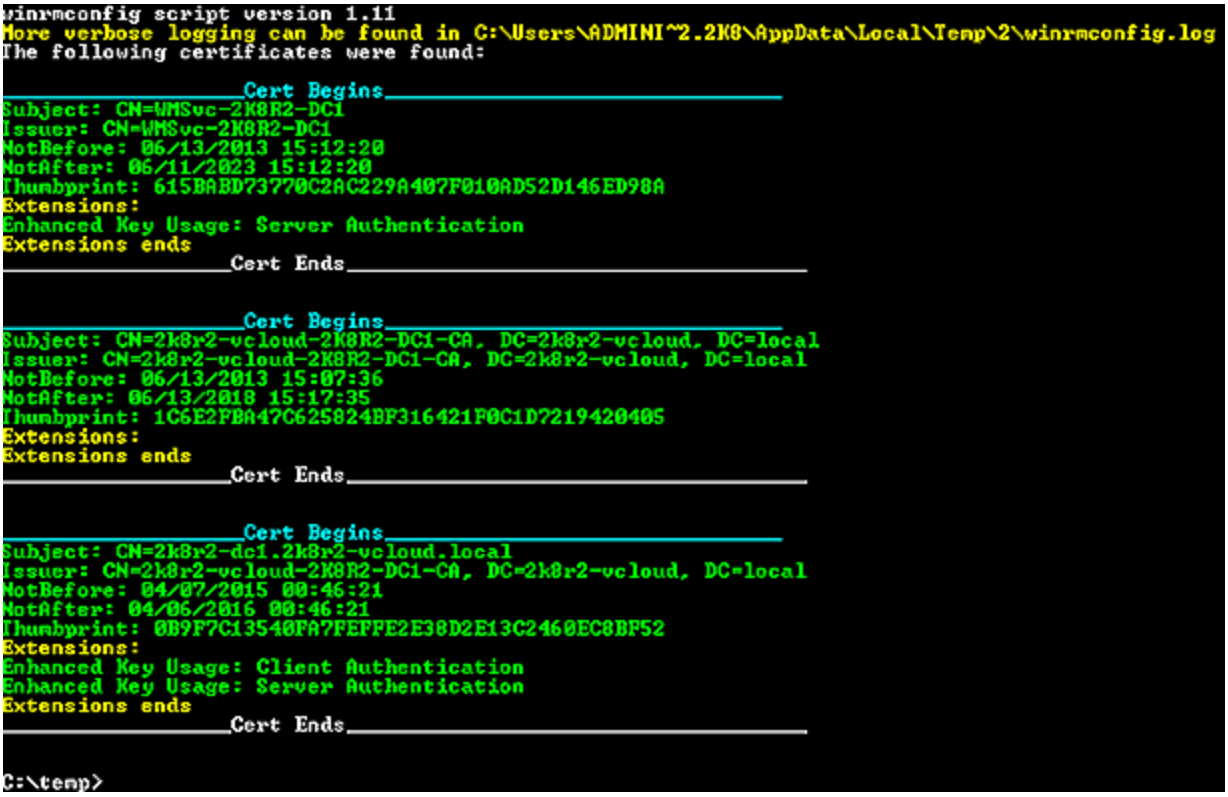
Other Script Options

The following sections describe certificate features that apply when you are using HTTPS.

WinRM Compatible Certificates used with HTTPS Listeners

The **ShowAllCerts** command dumps all machine certificates, along with their EKU bits. (A WinRM HTTPS listener requires a certificate that has at least "Server Authentication" EKU.)

```
Powershell -File winrmconfig.ps1 -Action ShowAllCerts
```



In this output, and the previous **enable** command, the last certificate in this list was selected by the script to use for creating the listener for the following reasons:

- The certificate is not expired
- The certificate has the Server Authentication EKU bit enabled
- The certificate's **Subject** field contains the FQDN of the system (which makes it a good choice in its favor since the subject CN must be the FQDN of the host for the listener creation to succeed)

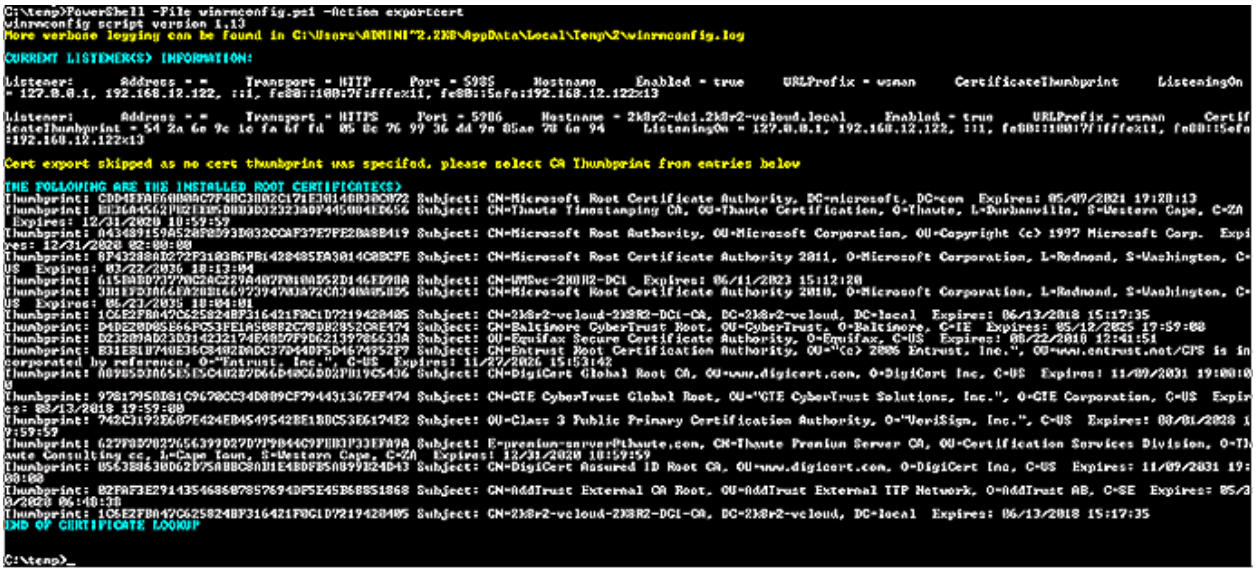
If another certificate has the FQDN in its **Subject** and is better suited, you can select it from the command line using the **-Thumbprint** option. The following example creates a new listener by selecting the certificate that has the specified thumbprint.

```
Powershell -File winrmconfig.ps1 -Action enable -ListenerType https -
Port 5555 -User newcoluser@2k8r2-vcloud -ThumbPrint 615BABD7377
229A407F010AD52D146ED98A
```

WinRM Over HTTPS with Mutual Authentication

You can enable HTTPS-based log collection, without importing certificates, via the NetWitness console. In the event source, HTTPS is selected, but the certificate field is left empty. This mode of operation is the same as using HTTPS for a web server without setting up user certificates. However, you can perform the extra step to import each Windows system's CA certificate into the Collector and configure the certificate name in each event source. This provides mutual authentication, that is, it allows the Collector to verify that the target system is configured for log collection. The **winrmconfig** script makes this process easier by using the **exportcacert** action and **-ExportCertPath** and **-ThumbPrint** options. For example, the following command lists all CA certificates on the system, as shown in the image below the string:

```
PowerShell -File winrmconfig.ps1 -Action exportcert
```



This command lists all the CA (Root) certificates. Typically, a CA certificate with the domain/hostname in the subject is the correct certificate.

You can select the thumbprint of the certificate from the list and then run the following command to export the certificate, convert it to PEM format, and save it to the path specified.

```
PowerShell -File winrmconfig.ps1 -Action exportcert -ExportCertPath
c:\temp\ -ThumbPrint 1C6E2FBA47C625824BF316421F0C1D7219420405
```

The PEM filename is a concatenation of the certificate subject and the thumbprint with .pem extension as shown below:

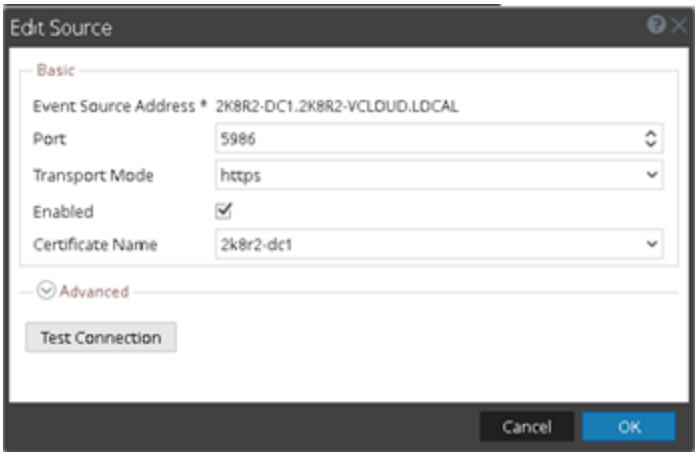
```
2k8r2-dc1.2k8r2-vcloud.local-CA-
1C6E2FBA47C625824BF316421F0C1D7219420405.pem
```

The **ExportCertPath** path can be a secure common share where the NetWitness UI can be used to import all the exported CA certificates.

To import the certificate into NetWitness:

1. Log on to the NetWitness Platform.
2. At the top of the page, click **Dashboard > ADMIN > Services**.
3. Select a Log Collector service, and in the **Actions** column, click the down arrow and select **View** > **Config**.
4. Click the **Settings** tab and from the left pane, select **Certificates**. The Certificates pane is displayed.
5. Click the plus sign to add a certificate. The Add Cert dialog is displayed.

6. Create a name for the certificate that is based on the system that the certificate came from, for example, the hostname.
7. Click **Browse** to select the newly exported PEM file. The script does not export the private key. (There is no requirement for the key to connect to the system.)
8. Click **Save**. (The password is not required for certificates exported by the script.)
9. Select the imported certificate in the Event Source (remember to change the port to HTTPS, the default is 5986).



Group Policy Method for Mass-enabling WinRM

Using Microsoft Group Policy Object (GPO) to enable WinRM can be very flexible. There are many ways to scope the number or types of systems the GPO is pushed to, for example, by using a WMI filter or by selecting specific domain names by using active directory domain trees. There are a number of decisions to be made before you decide to use GPO:

- Do I use an administrative or non-administrative account to collect events? (NetWitness strongly recommends using a non-administrative account.)
- Do I use HTTP or HTTPS transport ? (It's not uncommon to see both in use in the same domain, for example, HTTP for domain members and HTTPS for domain controllers.)
- Do all the systems, from which I collect events while using HTTPS transport, already have certificates?

With the answers to these questions in mind, taking the steps to deploy WinRM via GPO is actually quite simple, as the PowerShell script takes care of almost everything. The previous sections of this guide describe the **report** and **enable** modes of the script. In this section, only the **enable** mode is used, because we are setting up a GPO to configure systems from which NetWitness collects events.

Systems that require encrypted collection, for example, HTTPS, must have machine certificates installed that support at least server authentication key usage. Usually, this is not an issue for domain controllers. However, domain members and standalone systems often do not have certificates. This document has a section that describes pushing this type of certificate to those systems. There is no mandate that you use a single GPO to push the **winrmconfig** script. For example, HTTPS could be used to collect from domain controllers, and HTTP could be used for domain members. Separate Group Policy Objects could be created, one for each class of system to push the same script, with different parameters, to each of the different types of system.

WinRM over HTTPS SSL Certificate Deployment

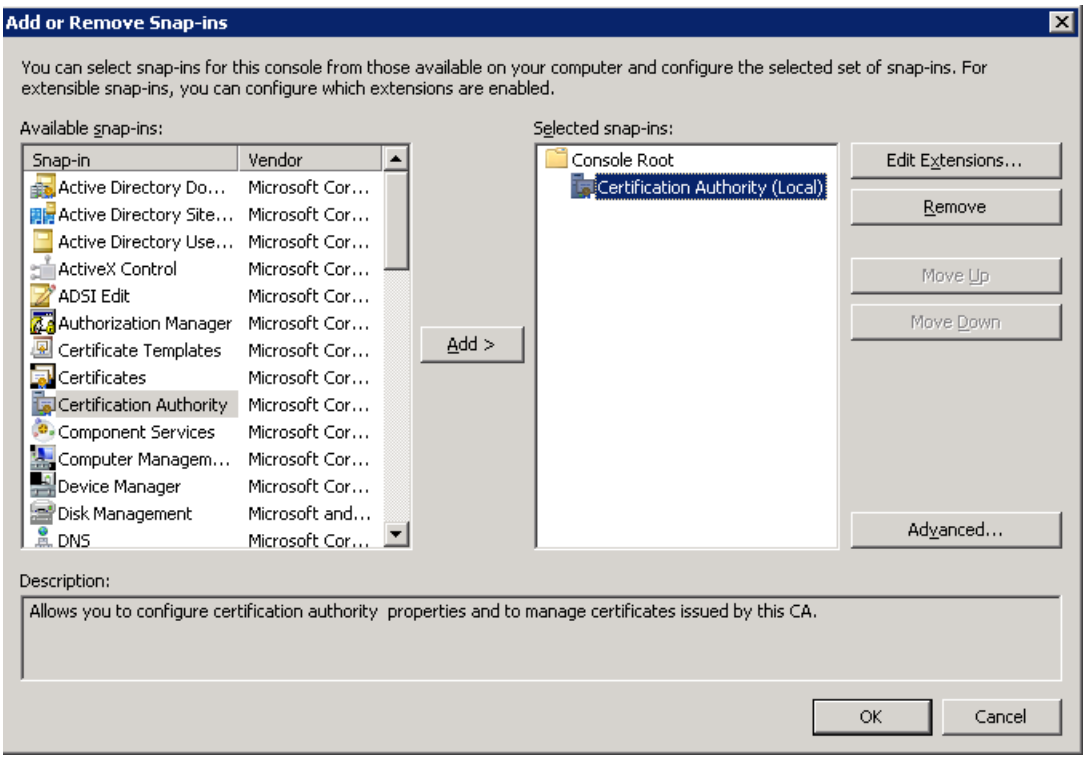
If you have decided to enable WinRM using HTTPS transport, the systems from which NetWitness collects events must have a viable certificate installed. The first step is to ensure that the systems from which NetWitness collects events have SSL certificates.

If those systems have certificates, skip this section. For example, most domain controllers already have a local machine certificate, so this step may not be necessary if you are collecting solely from domain controllers.

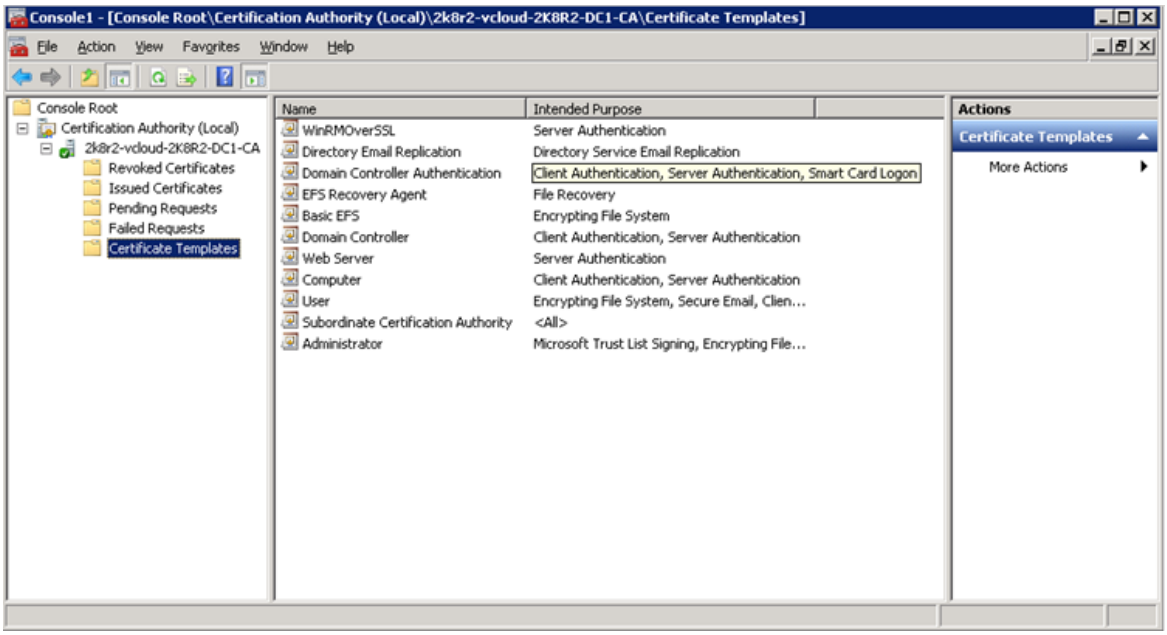
However, standalone machines might not have certificates. You can use a GPO to push certificates to machines that do not already have them by following the steps in this procedure.

Create Certificate Templates

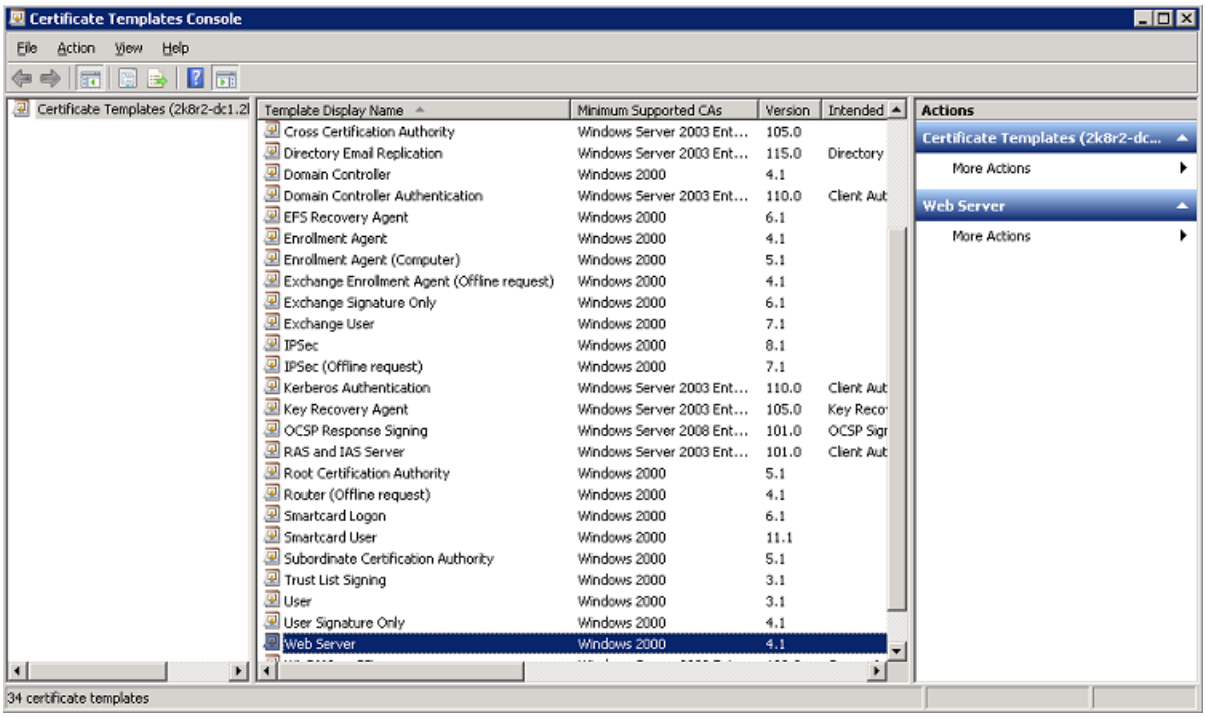
1. Run the local certificate authority application on the domain controller:
   a. At a command prompt, type `certsrv`.
   b. Run `mmc`, click **File** > **Add/Remove Snapin**, and then add the Certificate Authority as shown in the following figure.



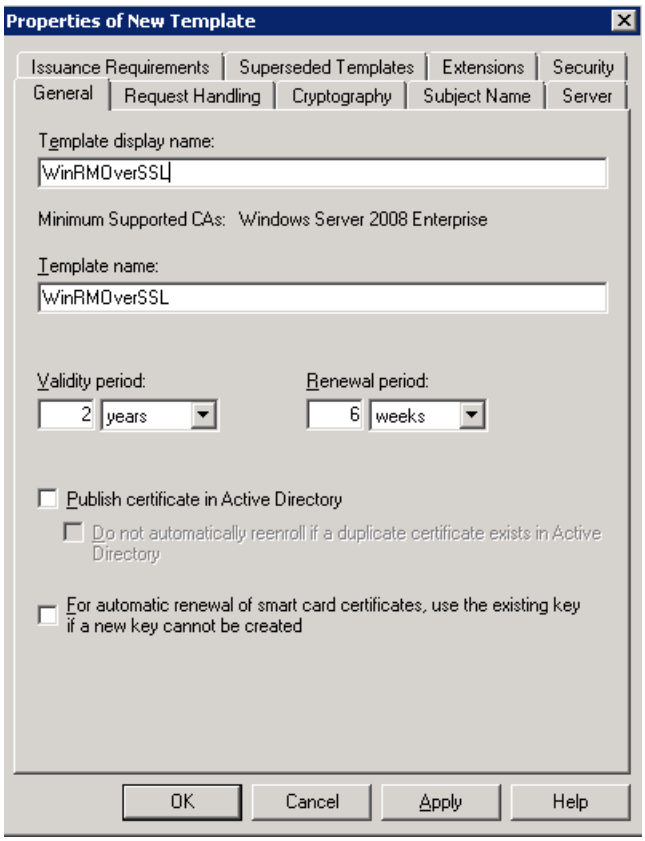
The Certificate Templates window opens.



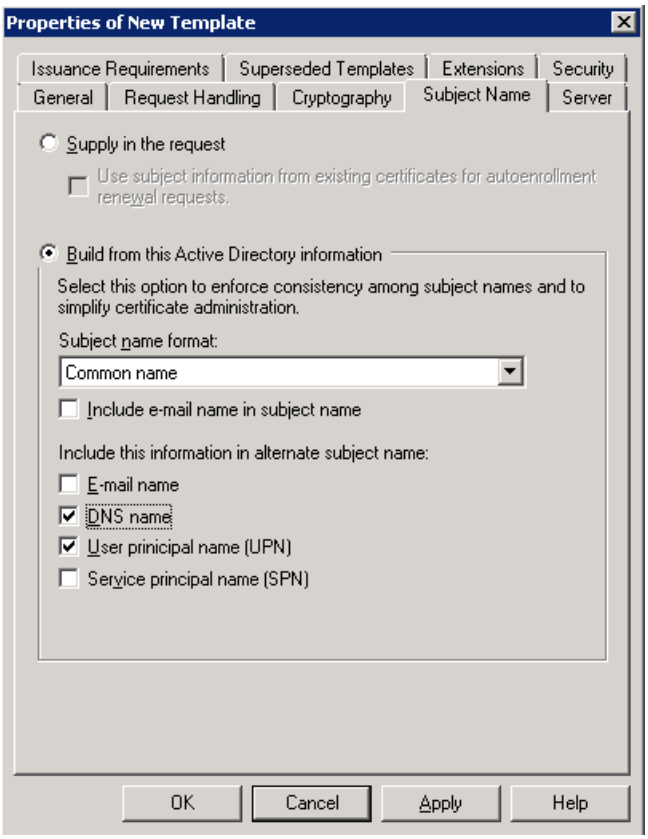
2. In the left pane, right-click **Certificate Templates** and select **Manage**. The Certificate Templates Console window is displayed.



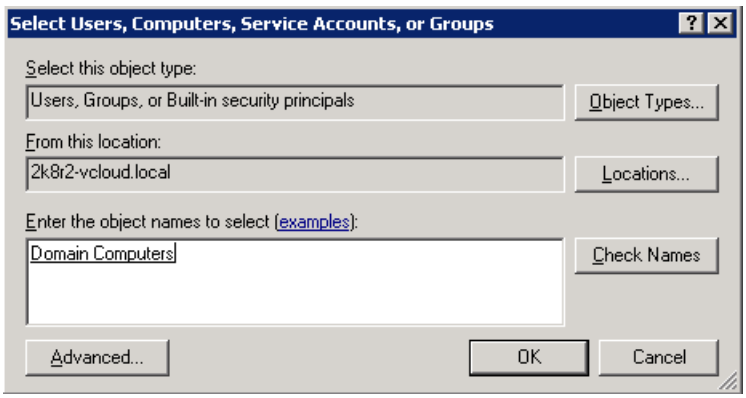
3. In the middle pane, right-click **Web Server** and select **Duplicate Template**. The Properties of New Template dialog opens.



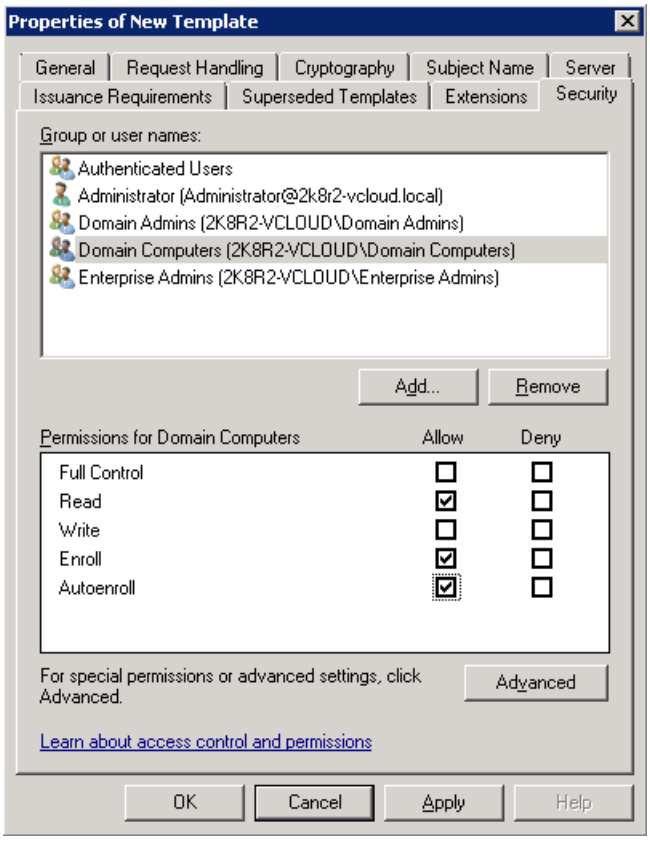
4. On the **General** tab, in both the **Template display name** and the **Template name** fields, type `WinRMOverSSL`.
5. Use the **Validity period** and **Renew period** fields to select a time period in which the certificate expires. Ensure that this time period complies with your company's policies.
6. Click the **Subject Name** tab and ensure that **Build from this Active Directory information** is selected.



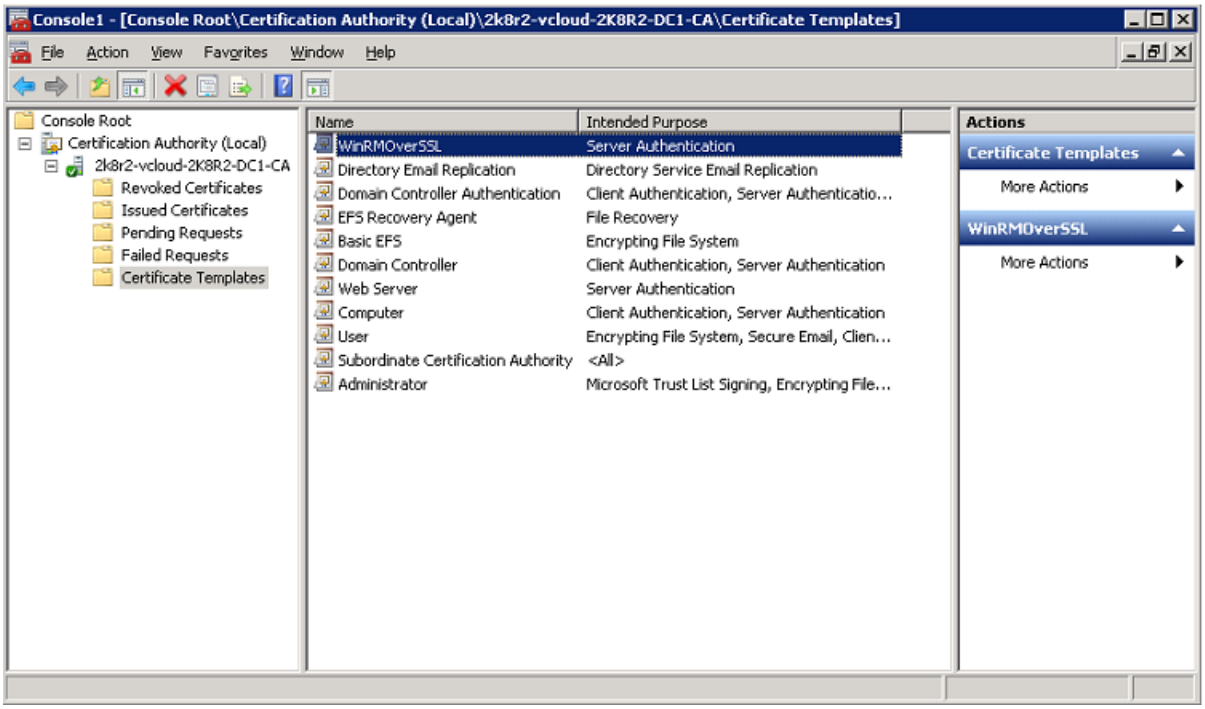
7. From the **Subject name format** drop-down menu, select **Common name**.
8. Under **Include this information in alternate subject name**, select **DNS Name**.
9. Click the **Security** tab, and then click **Add**. The Select Users, Computers, Service Accounts, or Groups dialog opens.



10. Click **Object Types**, select **Computers**, and click **OK**.
11. Click **Check Names**, enter `Domain Computers`, and click **OK**.
12. To configure permissions, click the **Security** tab.



13. Select **Allow** for **Read**, **Enroll**, and **AutoEnroll**. Click **Apply** and then click **OK** and close the Certificate Templates console.
14. To enable the new template, in the Certification Authority console, right-click **Certificate Templates**, click **New** > **Certificate Template to issue**, and select the template that you just created.

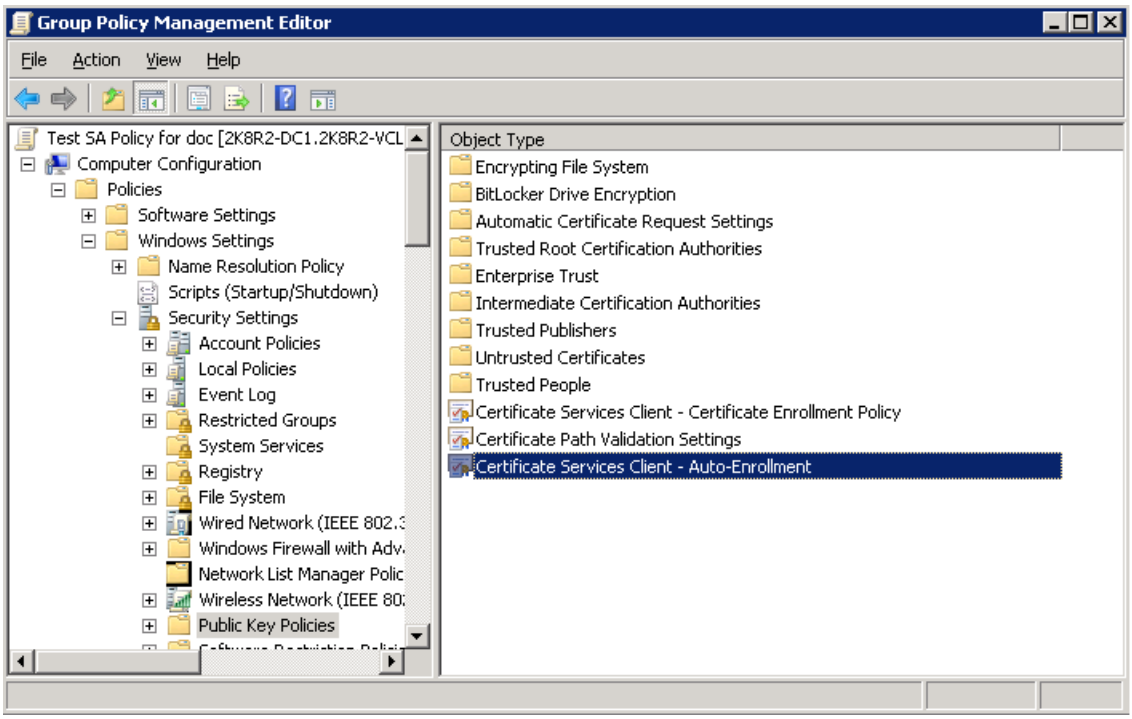


Link Certificate Templates to Group Policies

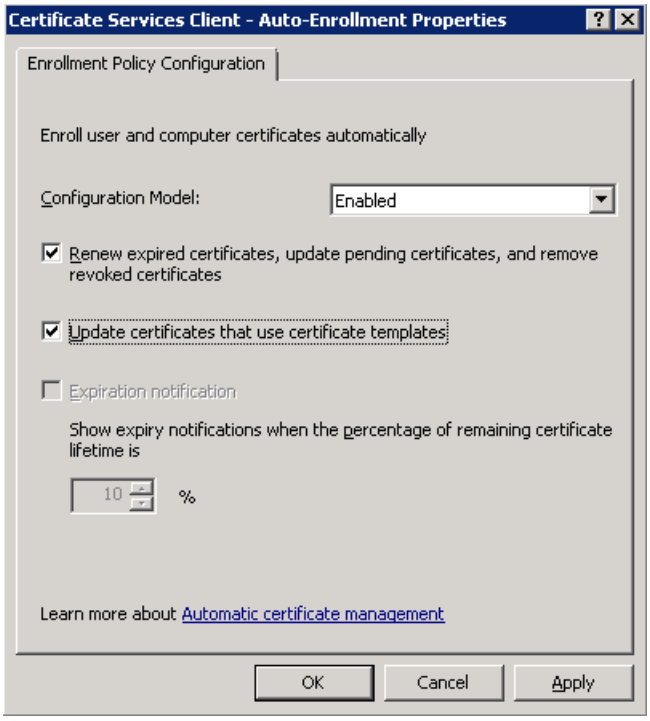
To create and publish certifications, you must link certificate templates to Group Policy Objects (GPOs).

To link a template to a Group Policy Object:

1. To open the Group Policy Management Console on the domain controller, click **Start** > **Administrative Tools** > **Group Policy Management**, and in the left pane, browse to the **Group Policy Objects** folder for your domain.
2. Right-click the Group Policy Objects folder and click **New**
3. Type `WinRMCertEnrollment`, and leave **Source starter GPO** as none.
4. Right-click on the `WinRMCertEnrollment` policy and click **Edit**. The Group Policy Management Editor dialog opens.



a. Select **Computer Configuration** > **Policies** > **Windows Settings** > **Security Settings** > **Public Key Polices**. In the right pane, double-click on **Certificate Services Client - Auto Enrollment**. The Certificate Services Client - Auto-Enrollment Properties dialog opens.



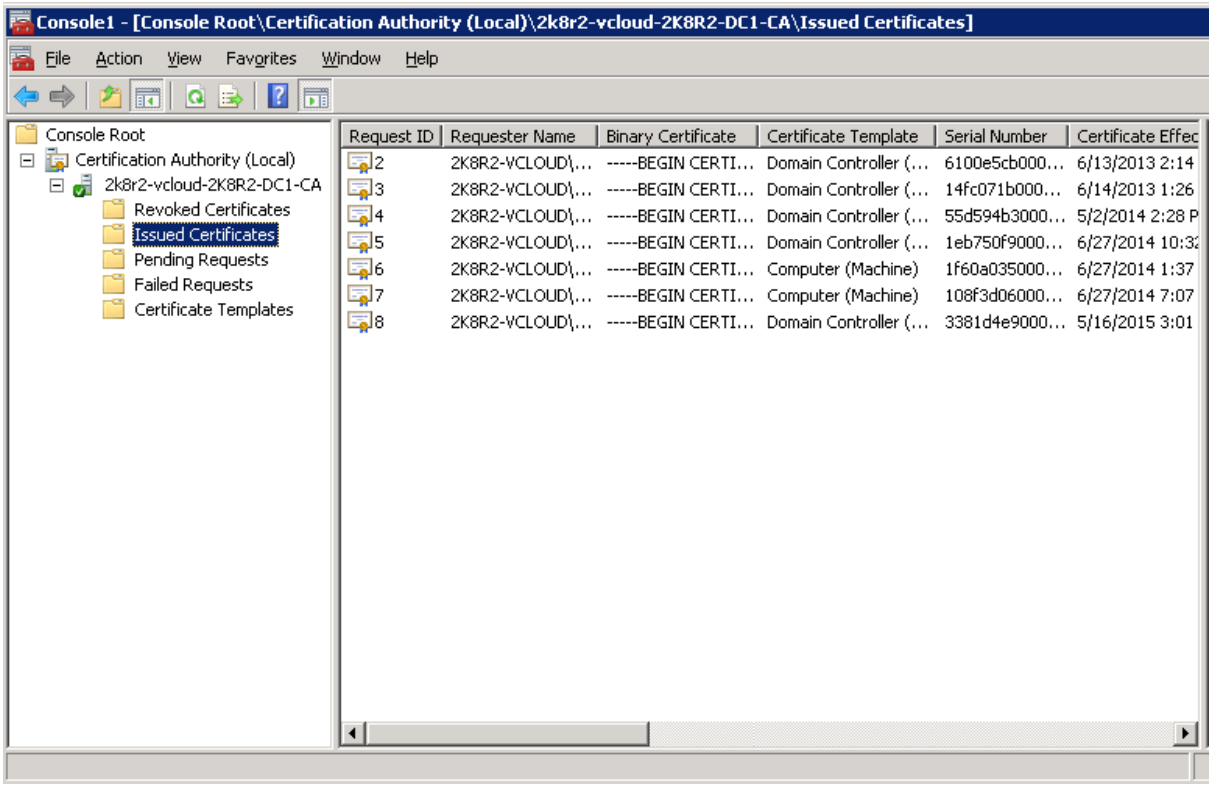
b. To set up automatic renewal of user and computer certificates, select **Renew expired certificates update pending certifications, and remove revoked certificates** and **Update certificates that use certificate templates**.

Note: It may take several hours before the policy is pushed and actioned. To deploy the certificate faster, run `gpupdate /force` from an elevated command line on any system that is covered by the GPO.

Use the Issued Certificates page in the Certificate Authority to see if the GPO is causing certificates to be pushed. It is important to find out when your GPO pushed certificates to all the machines within the scope of the GPO, because the HTTPS listeners that you create only work after the certificates have been pushed to those machines.

To find out if your GPO is causing certificates to be pushed:

1. In the Certificate Templates Console, in the left pane, click **Issued Certificates**
2. In the right pane, information about requests for certificates is displayed.

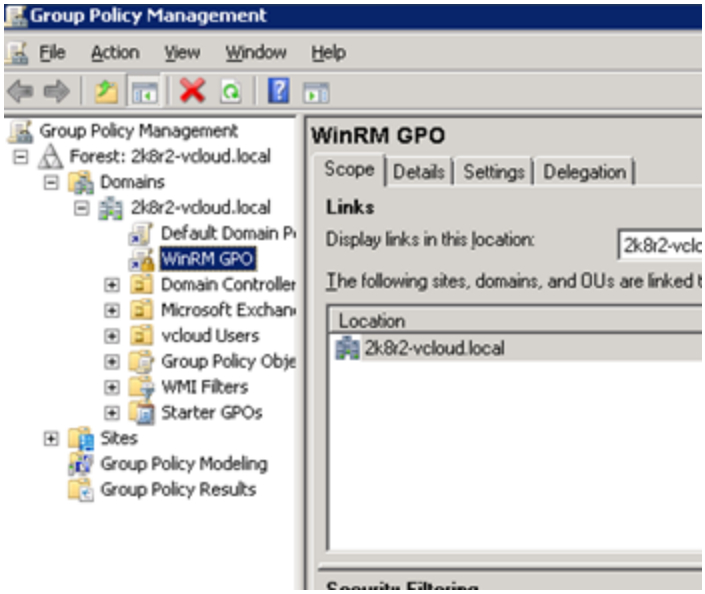


Create GPO to Push the winrmconfig Script

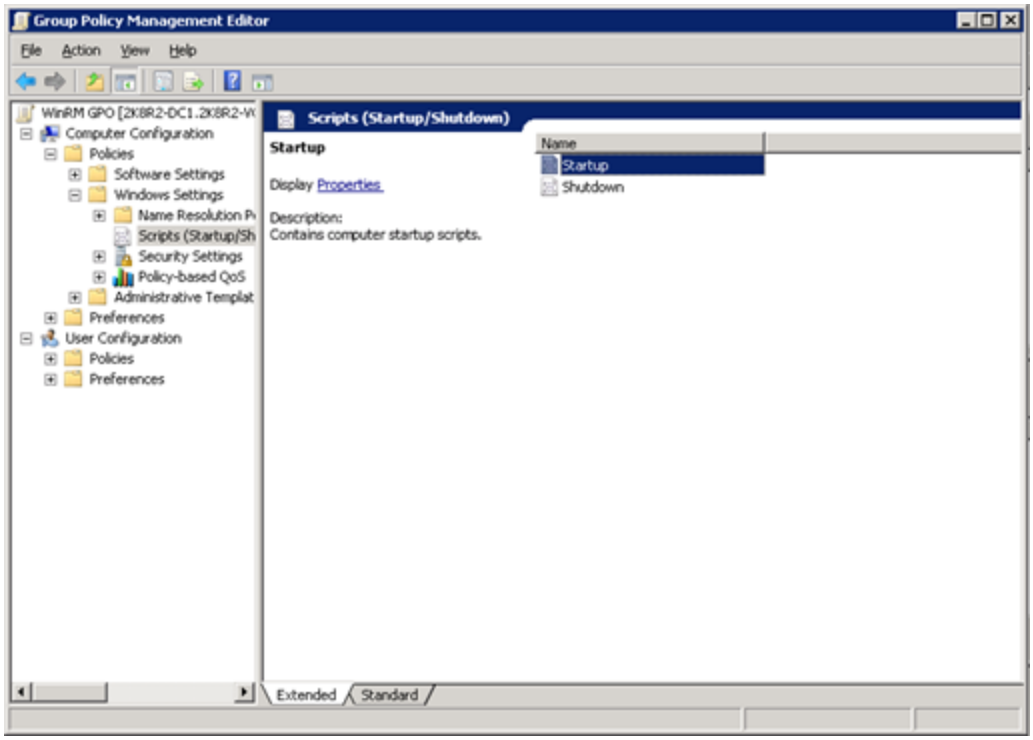
To vastly simplify WinRM configuration via GPO, you can push the **winrmconfig** PowerShell script to all systems from which NetWitness collects events. The script runs as a system startup. It creates any listeners that are needed and enables the relevant permissions for a non-administrative account to be used to collect events and enumerate SIDs (which are the two operations that a NetWitness Collector performs). If the collection user was an administrative account and the listener type was HTTP, a very simple GPO could be used. However, as previously stated, NetWitness does not recommend using an administrative account.

1. Open the Group Policy Management Console on the domain controller: Click **Start** > **Administrative Tools** > **Group Policy Management**, and in the left pane, browse to the **Group Policy Objects** folder for your domain.

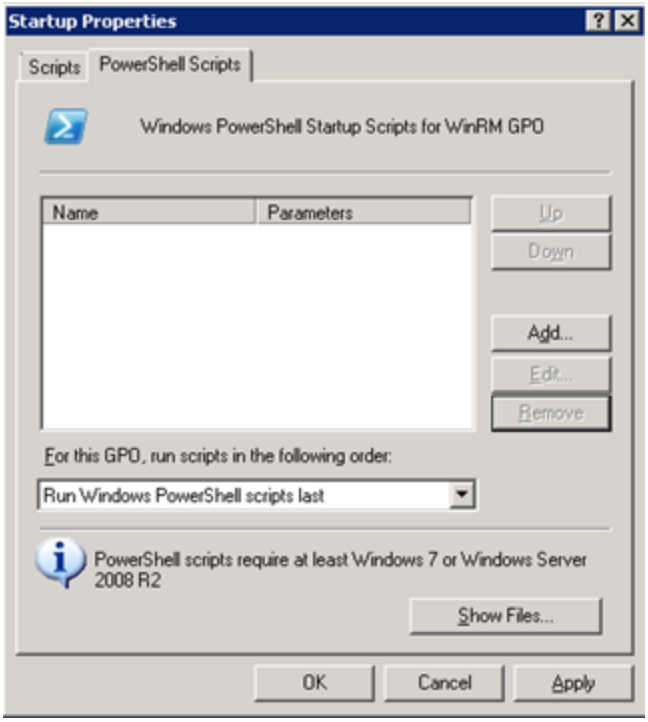
2. Create a new Group Policy object, for example, `WinRM GPO`. Refer to your network administrator when deciding on the scope of the GPO. Defining the scope of the GPO directly chooses the machines the script will be run on. See step 8 in this procedure for information about WMI filters.



3. Right-click the newly-created Group Policy object and click **Edit**.



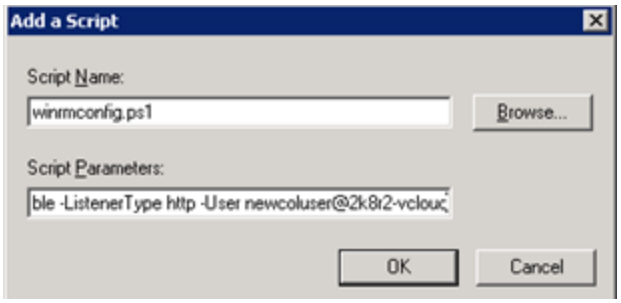
4. Expand **Computer Configuration** > **Windows Settings** > **Scripts** and double-click **Startup**.



5. Click the **PowerShell Scripts** tab, click **Add**, and then click **Browse**. The default directory in the Browse dialog should be the current **sysvol** path, for example:

   ```
   \\2k8r2.local\SysVol\2k8r2.local\Policies\{68EB4039-9768-48B2-9E0B-58153D188233}\Machine\Scripts\Startup
   ```

6. Copy and paste the path from the Browse dialog, and using that path, copy the **winrmconfig** script to that location using Windows Explorer. The script is displayed in the Browse dialog. Select the script. The Add a Script dialog is displayed.



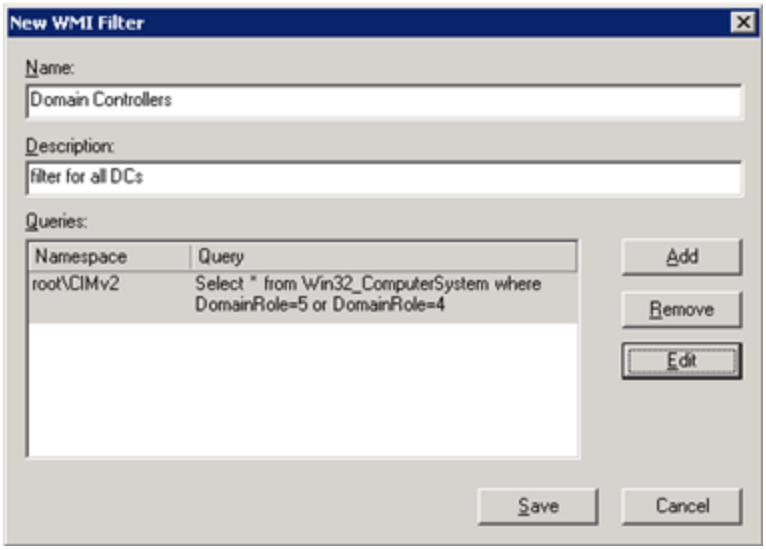
   The **Script Parameters** field is the key to configuring listener and collection user permissions, as this is where options such as the listener type and collection user account is specified. In this document, the sections winrmconfig Modes and Other Script Options describe options for the script. Here are some examples of script options:

   - **Action enable -ListenerType http -User myuser@domain.com** causes the script to create an HTTP listener on each system that the GPO is applied to, and creates all the permissions for the **myuser@domain.com** account to enable event collection from those systems.
   - **Action enable -ListenerType https -User myuser@domain.com** causes the script to create an HTTPS listener using any available non-expired certificate that has the Server Authentication key usage enabled, with a Subject DN string that matches the name of the system. If the certificate is

not available, it should have been pushed to the listener using the previous steps to push certificates to systems via GPO.

- **Port xxxx** causes the script to create a listener on a customer port other than the default 5985 for HTTP or 5986 for HTTPS (-Port 5999 for example).

7. Paste the string that you selected in step 6 into the **Script Parameters** field and click **OK**, **Apply**, and **OK** to return to the Group Policy Editor dialog.
8. Link the GPO to a container in the Active directory that fits the scope of the systems to which to apply the GPO. In this case, the scope is the whole domain or all systems in the domain. You can select a WMI filter to better isolate the machines to which to push the GPO.



In this example, the GPO will be pushed to the domain primary and backup domain controllers only.

With the GPO linked and enabled, if a system in the scope of the GPO is rebooted, the script will execute on restart and configure WinRM in the manner determined by the script parameters of the startup script.

Firewall Information

The following information is relevant for systems that are using firewalls.

Systems running Windows built-in Firewall

If the winrmconfig script is run with the `--enable` option on a system that has the Windows local firewall service running, the script automatically adds the necessary local firewall port rules, based on the selected port or the default WinRM listener port for the listener type.

Other Firewall Considerations

Note that the Collector must be able to poll each configured WinRM system. So, if there is a firewall between the Collector(s) and system being collected from via WinRM, the configured listener port must be opened. This is true regardless of the authentication type selected in the Event Category section of the WinRM configuration.

If you selected "Negotiate" as the authentication method, the collector needs to obtain a TGT (Ticket Granting Ticket) in order to get Service Tickets for the configured systems being collected from. This applies to all collectors, both local and remote, that are performing collection via WinRM. That is, each Windows system being collected from would need a rule to open port 88 udp/tcp for all configured Domain controllers configured under the Kerberos realms configuration tab.

Note: NetWitness highly recommends that if you are using the Negotiate method of communication, you create a firewall rule to open port 88 for UDP and TCP, in the direction of the Log Collector to the Domain Controllers being configured. Requests go from the LC to the DC only, so the rule can be one-way.

Both TCP and UDP access (at least one way from collector to Domain controller) is required as the Kerberos implementation on the collector initially attempts to request the TGT or ST using UDP on port 88. However, the response to Kerberos may be too large for a single UDP response. The Domain Controller will then respond with a "response too large" message which will cause the collector will switch automatically to TCP on port 88 and re-request the tickets.