Defendpoint Windows Event Forwarding

Software Version: 5.0

Document Version: 1.2

Document Date: March 2018

Accessibility Notice
In the event that you are unable to read any of the pages or documents on this website, please contact us and we will arrange to get an accessible version to you.

Table of Contents

Chapter 1 - Introduction

This document provides guidance on how to centralize Defendpoint events to a central server using Windows Event Forwarding. Avecto provides an Enterprise Reporting Pack which includes enterprise class trend analysis dashboards, allowing organizations to understand and be pro-active about the Defendpoint events raised within their environment.

With the Enterprise Reporting Pack, Defendpoint events from all managed endpoints can be centrally collected to a SQL Server database. The Enterprise Reporting Pack builds on a number of Microsoft technologies which include Windows Event Forwarding, SQL Server, and SQL Server Reporting Services (SSRS). This approach provides a scalable and secure architecture, which can cope with high volumes of events and handle the largest enterprise environments. For more information on Avecto's Enterprise Reporting Pack visit www.avecto.com

Event Forwarding is provided by Windows Remote Management (WinRM) which is Microsoft's implementation of a WS-Management Protocol, a SOAP based firewall-friendly protocol, which provides a common way for systems to access and exchange management information across an IT infrastructure.

One of the most powerful features of WinRM is the ability to forward events which enable large scale health and state status monitoring of Windows environments (also known as Windows Eventing 6.0). Not only is this feature built into the latest versions of Windows (originally shipped with Windows Vista and Windows Server 2008), but it's also freely/readily available for down-level operating systems.

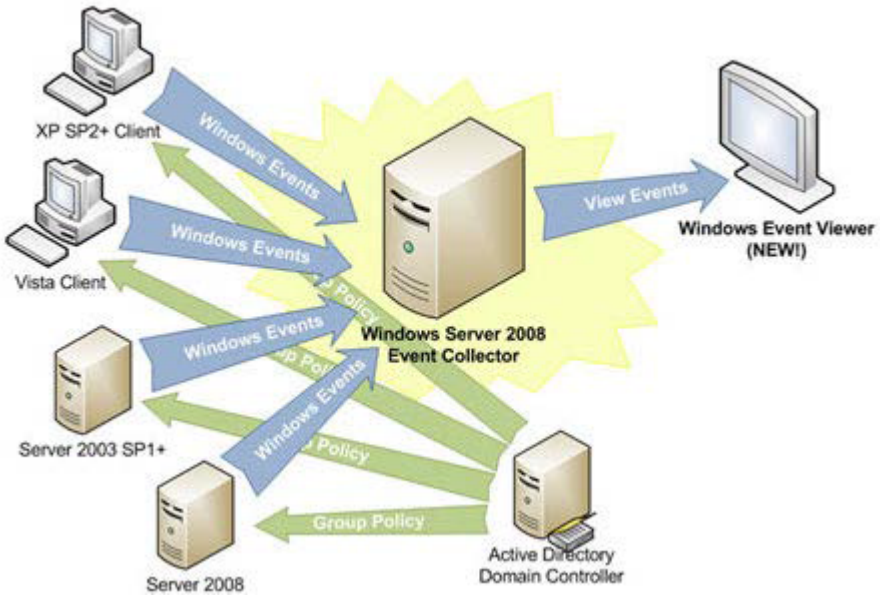
Chapter 2 - Windows Event Forwarding Collection

2.1 - Features

1. **Standards Based**: Leveraging the DMTF WS-Eventing standard which allows it to interoperate with other WS-Man implementations (see OpenWSMAN at SourceForge).
2. **Agentless**: Event Forwarding and Event Collection are included in the operating system by default.
3. **Down-Level Support**: Event Forwarding is freely/readily available.
4. **Multi-Tier**: Forwarding architecture is very scalable where a Source Computer may forward to a large number of collectors and collectors may forward to collectors.
5. **Scalable**: Event Collection is very scalable where the collector can maintain subscriptions with a large number of Source Computers and events per second.
6. **Group Policy Aware**: The entire model is configurable by Group Policy.
7. **Schematized Events**: Windows Events are now schematized and rendered in XML which enables many scripting and export scenarios.
8. **Pre-Rendering**: Forwarded Windows Events can now be pre-rendered on the Source Computer negating the need for local applications to render Windows Events.
9. **Resiliency**: Designed to enable mobile scenarios where laptops may be disconnected from the Event Collector for extended periods of time without event loss (except when logs wrap) as well as leveraging TCP for guaranteed delivery.
10. **Security**: Certificate based encryption via Kerberos or HTTPS.

2.2 - Architecture

The architectural approach used in this guide utilizes Group Policy to distribute WinRM and event forwarding configurations to a group of domain computers. Each client will be configured to forward events to a central Event Collector.


XP SP2+ Client
Windows Events
Vista Client
Windows Events
Windows Events
Server 2003 SP1+
Windows Events
Server 2008
Group Policy
Active Directory
Domain Controller
Windows Server 2008
Event Collector
View Events
Windows Event Viewer
(NEW!)

Chapter 3 - Pre-requisites

3.1 - Central Event Collector

A central Event Collector must be used as a repository for all the events collected from the Source Computer.

Windows Vista, Windows 7, Windows Server 2008, Windows Server 2008 R2 and Windows Server 2012 can be Event Collectors (this feature is not supported for down-level operating systems). There are no built-in limitations when client operating systems are used as an Event Collector. However, it is recommended that Server 2008/R2 or Server 2012 are used as the Event Collector as this will scale much better in high volume scenarios.

When using Windows Vista or Windows Server 2008 as the Event Collector, it is strongly recommended that you upgrade to Windows Remote Management 2.0. This will allow Windows 7 clients to be monitored without any additional configuration.

Depending on the volume of events, the Event Collector can either be a dedicated or an existing machine. True enterprise class Windows Eventing is included with enterprise monitoring solutions like System Center Operations Manager (SCOM) (Audit Collection Services ACS).

3.2 - Event Source Computers

The minimum operating system level required on the Source Computer is Windows 7.

Events can be centralized onto any of the supported Windows Event Collector operating systems from any supported Windows event source operating systems. Each Source Computer must have minimum of Windows Remote Management 1.1.

The following table shows the default installation for each OS:

Operating System	Windows Remote Management Version
Windows 7	2.0
Windows Server 2003/R2	Not installed
Windows Server 2008 R2	2.0
Windows Server 2012	2.0

Chapter 4 - Implementing Windowns Event Forwarding

Delete this text and replace it with your own content.

4.1 - Summary Checklist for the Setup of Event Forwarding

1. Install and disable the Avecto agent

   It is recommended that this step is performed before the creation of the subscription, as a reboot is required in order for the service to be made available to the subscription.

   The "Avecto Defendpoint Service" needs to be set to disabled to deactivate the agent

2. WinRM quickconfig

   See section **Configuring Event Collection Services and Windows Firewall** detailed on the next page.

3. Wecutil qc

   See section **Configuring Event Collection Services and Windows Firewall** detailed on the next page.

4. Create and name Subscription in Event Viewer

Name		Avecto Events
Destination		Forwarded Event Log
Type		Source Initiated subscription
Source Computers:		Domain Computers or other group containing computers in scope
Select Events:		
	Event Level:	
	By Source:	Avecto Privilege Guard Service / Avecto Defendpoint Service
Advanced:		Minimize Latency

   See section **Configuring Event Subscriptions** detailed on page 11.

5. wecutil ss <subscriptionname> /cf:Events

   This changes the subscription from the default behaviour of **RenderedText** to **Events**, which has the dual benefit of reducing Source Computer CPU overhead and the event size.

   See section **Pre-rendering Events** detailed on page 15.

6. wecutil ss <subscriptionname> /ree:True

This setting ensures that all desired events in the Application event log on a Source Computer are forwarded to the event collector; the default behaviour is to only forward future (arriving) events from the point the subscription begins and this can result in missing data.

See section **Increase the Event Batch Size** detailed on page 16.

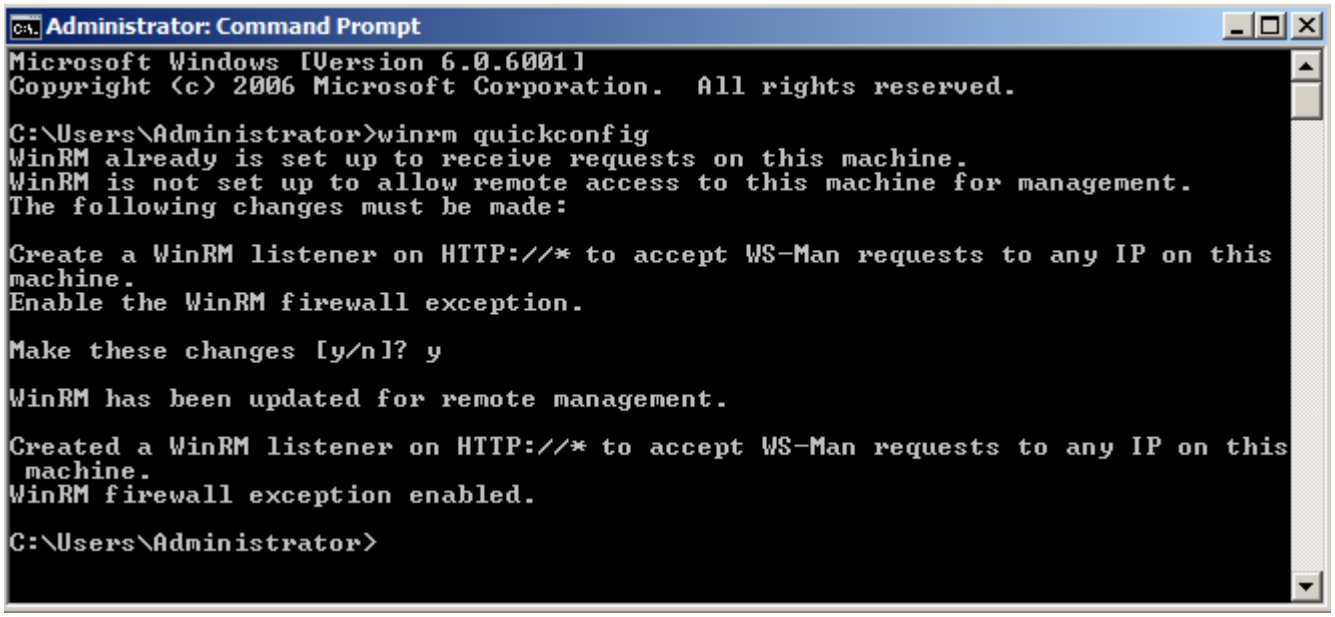
4.2 - Configuring the Event Collector(s)

4.2.1 - Configuring Event Collection Services and Windows Firewall

In order for Source Computers to communicate with the Event Collector machine, the correct inbound firewall ports need to be open and accepting connections. In addition, the WinRM and Event Collector services need to be running.

Configuration Steps:

1. On the **Event Collector** machine open a command prompt.
2. Type **winrm quickconfig**



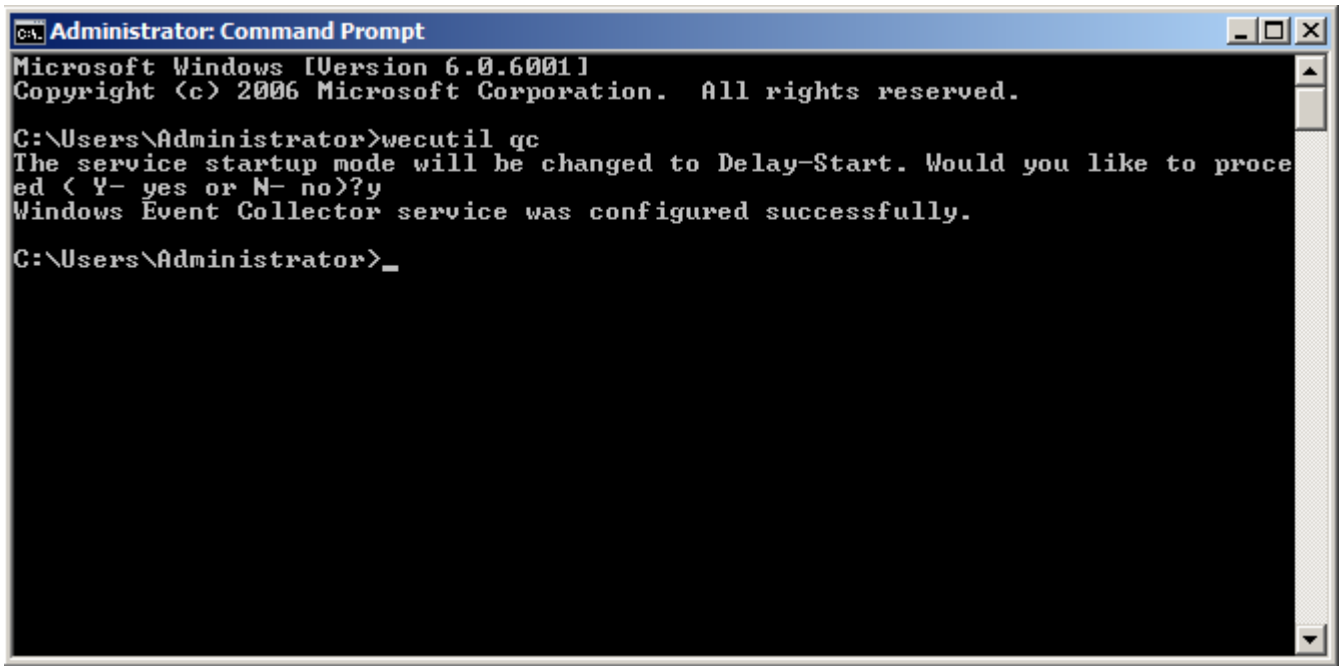
3. When prompted whether to continue with the configuration type **Y**.

- This command will check the current configuration and make the necessary changes. Upon completion the following will have been configured:
- Windows Remote Management service set to Automatic (Delayed Start) and Started. Windows Firewall port(s) Windows Remote Management (HTTP-In) Port 5985 configured for inbound communication OR Windows Firewall port(s) Windows Remote Management (HTTP-In) – Compatibility Mode - Port 80 configured for inbound communication.

In addition, the Event Collector service needs to be configured and started.

Configuration Steps:

1. On the **Event Collector** machine open a command prompt.
2. Type **wecutil qc**



3. When prompted whether to continue with the configuration type **Y**.

This command will check the current configuration and make the necessary changes. Upon completion the following will have been configured:

- **Windows Event Collector** service set to **Automatic (Delayed Start)** and **Started**.

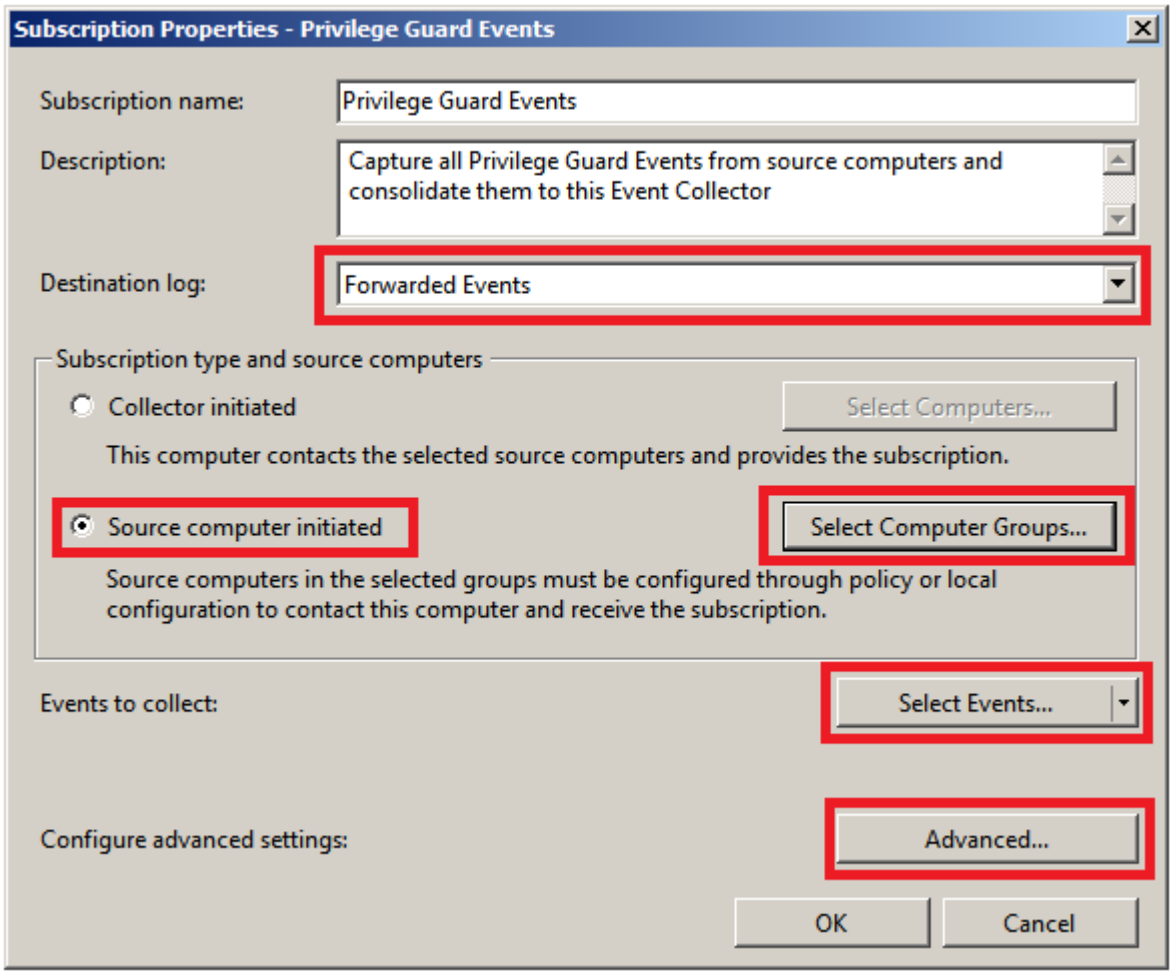
4.2.2 - Configuring Event Subscriptions

The Windows Event Forwarding architecture stores the subscription definition on the Event Collector, in order to reduce the number of touch-points in case a subscription needs to be created or modified. The following subscription will be configured so that event source computers retrieve subscriptions from the event collector host (Source-Initiated subscriptions).

Subscriptions are defined on the Event Collector through the new Event Viewer user interface by selecting the Create Subscription action, when the Subscriptions node is highlighted. The Subscription may also be created via the WECUTIL command-line utility.

Configuration Steps:

1. On the **Event Collector** open the **Event Viewer**.
2. Navigate to the **Subscriptions** node.
3. From the menu bar, choose **Action** > **Create Subscription...**
4. The **Subscriptions Properties** dialog will appear:

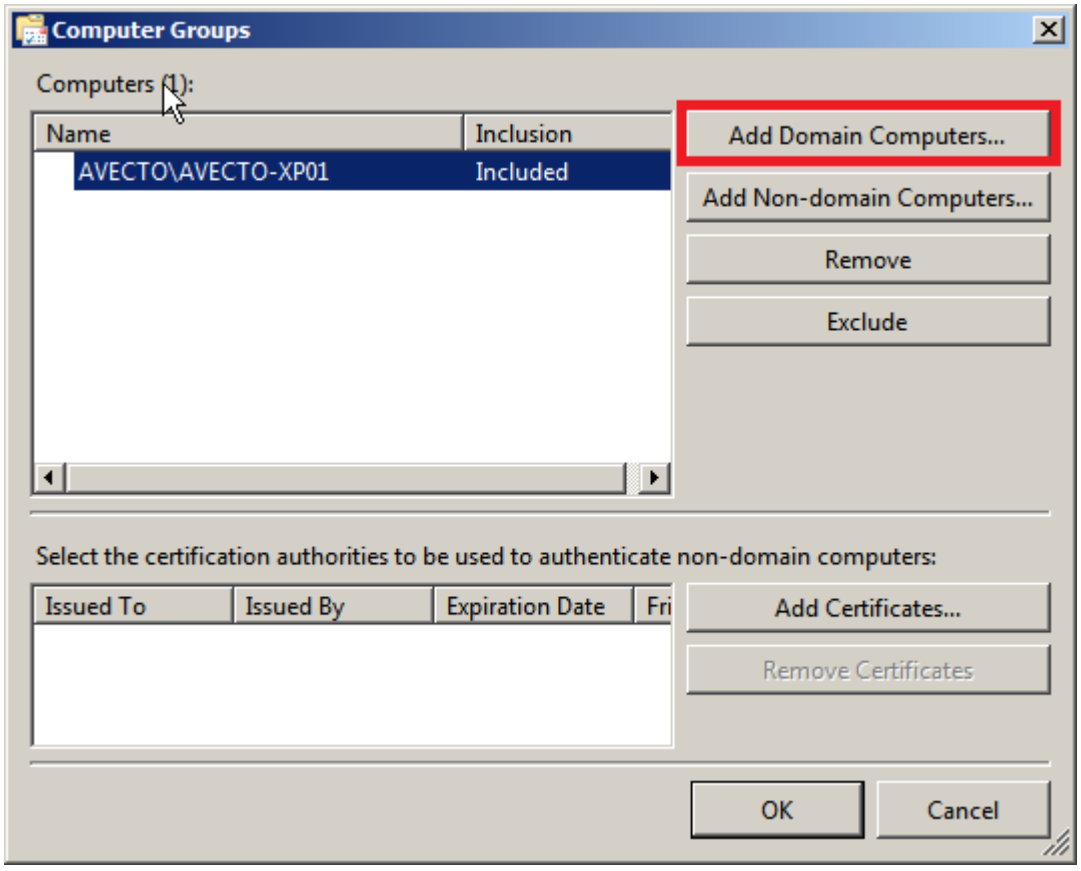


From here, you can specify a name, description, and the destination log (where the events will be collected).

5. Select **Forwarded Events** for the destination log.
6. Choose **Source Computer Initiated** (as Group Policy configures the Source Computer to contact the Event Collector for subscriptions settings).

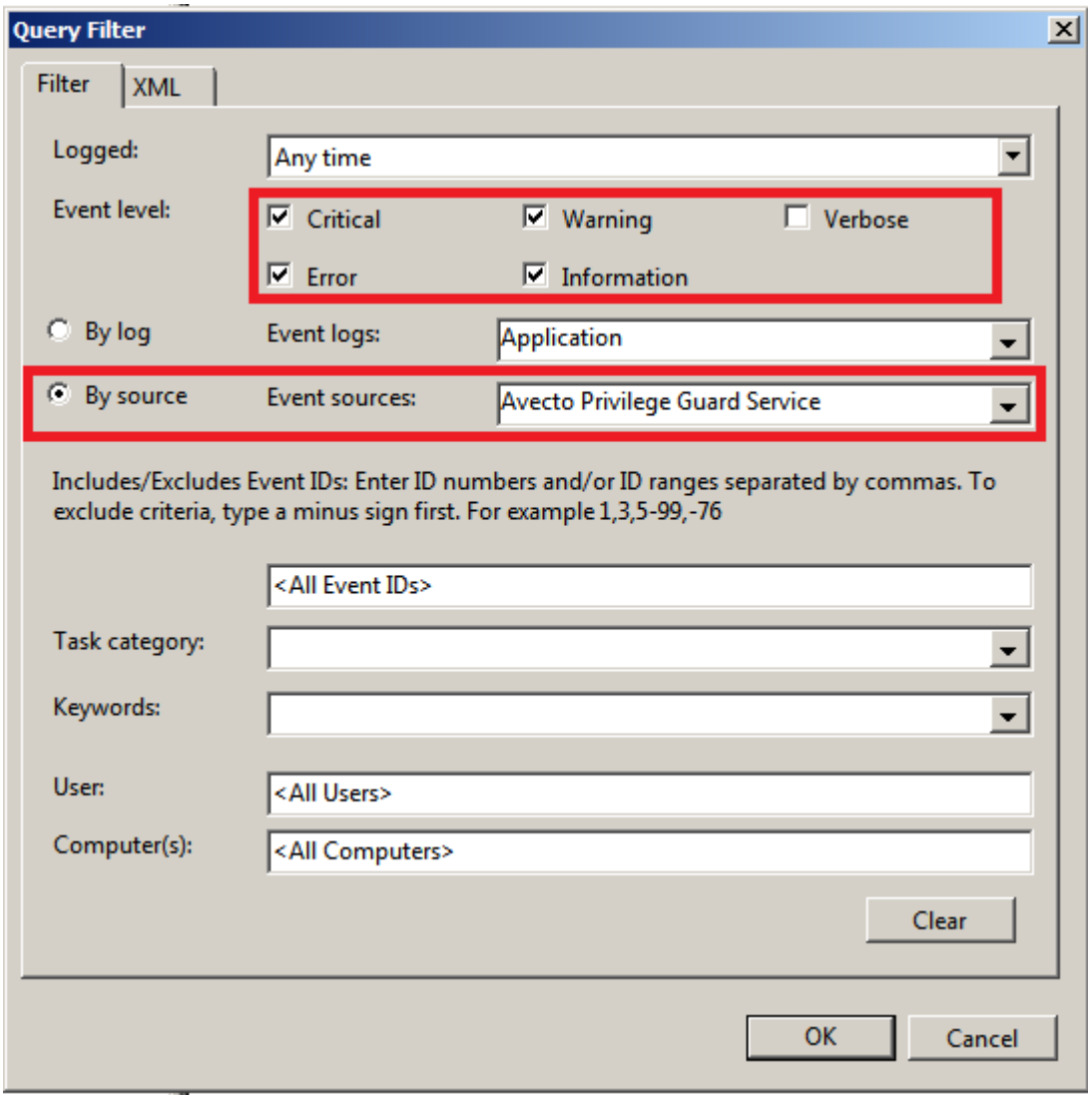
The Subscription Type can also be configured as Collector initiated. In this case Source Computers will need to be manually added to the Subscription either through the Subscription configuration or the WECUTIL command-line utility (which can also be scripted using PowerShell). It is recommended that Source computer initiated is used, as this configuration is the most scalable.

7. Click **Select Computer Groups**.
8. Click **Add Domain Computers** and select the required Source Computers.



It is recommended that a computer group which includes the required computer accounts, such as the Domain Computers group, is added to the subscription.

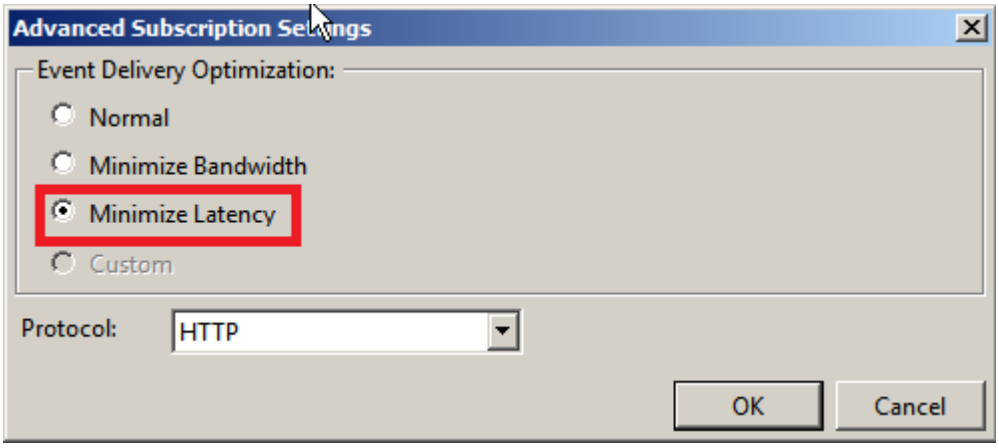
9. Click **OK** on the **Computer Groups** dialog.
10. Click Select Events.
11. Configure the following **Query Filter**:
    - Event Level = Critical, Warning, Error, Information
    - By Source = Avecto Defendpoint Service



In a production environment it may be advantageous to gather all events from the Application and System logs that have a level of Critical, Error, or Warning. This event scope can be expanded to gather all events from these logs or even add additional logs (like the Security log).

If the Defendpoint Agent is not installed on the Event Collector you will not be able to select Avecto Defendpoint Service as the Event Source. It is recommended that the Defendpoint Agent is installed and the "Avecto Defendpoint Service" set to disabled to deactivate the agent, if desired. If it is not possible to install the agent the subscription can be configured to collect events from the Application event log and filtered on event IDs 100 to 501. Please refer to the Defendpoint Administration Guide to verify the minimum and maximum event ID's created by the Defendpoint Service as these are subject to change.

12. Click OK on the Query Filter dialog.
13. Click Advanced on the Subscriptions Properties dialog.
14. Select Minimize Latency.



Normal - This option ensures reliable delivery of events and does not attempt to conserve bandwidth. It is the appropriate choice unless you need tighter control over bandwidth usage or need forwarded events delivered as quickly as possible. It uses pull delivery mode, batches 5 items at a time and sets a batch timeout of 15 minutes.

Minimize Bandwidth - This option ensures that the use of network bandwidth for event delivery is strictly controlled. It is an appropriate choice if you want to limit the frequency of network connections made to deliver events. It uses push delivery mode and sets a batch timeout of 6 hours. In addition, it uses a heartbeat interval of 6 hours.

Minimize Latency - This option ensures that events are delivered with minimal delay. It is an appropriate choice if you are collecting alerts or critical events. It uses push delivery mode and sets a batch timeout of 30 seconds.

Protocol - HTTPS can be used to secure the communication channel. However, this requires additional configuration steps and requires the Event Collector to use a certificate, see the appendences for more information

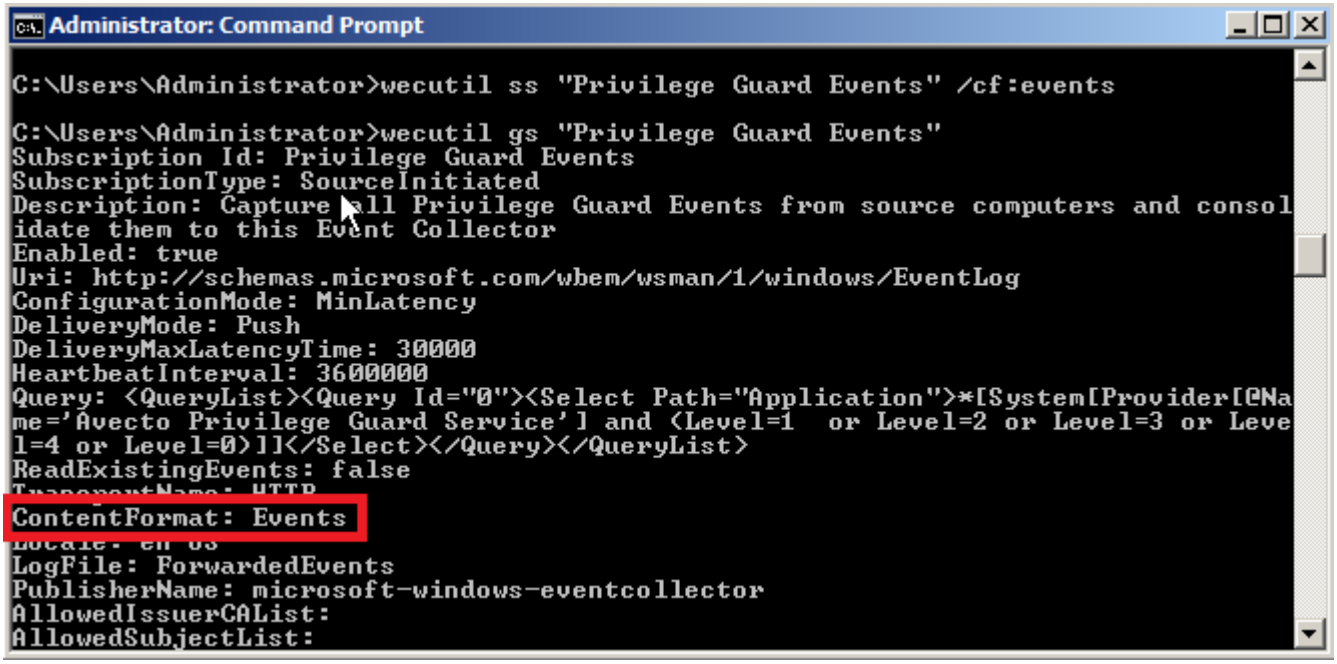
15. Click **OK** on the **Advanced Subscription** dialog.
16. Click **OK** on the **Subscription Properties** dialog.

4.2.3 - Pre-rendering Events

If the Source Computer is generating a large volume of forwarded events (e.g. Security events from a Domain Controller) then it is recommended that event rendering is disabled on the Event Collector. The task of pre-rendering an event on the source computer can be CPU intensive for a large number of events.

Configuration Steps:

1. On the **Event Collector** open a command prompt.
2. Type **wecutil ss** <subscriptionname> /cf:events



This will change the ContentFormat from RenderedText to Events which has the dual benefit of reducing Source Computer CPU overhead and the event size.

To view Event Subscriptions use the WECUTIL command utility and type: wecutil gs<subscriptionname>

4.2.4 - Increase the Event Batch Size

The batch size of events can be increased to reduce frequency at which Source Computers send their data. The follow command syntax can be used be to configure this; the example used here sets the batch size to 10,000.

wecutil ss sub_name /cf:Events /ree:false /dmi:10000 /cm:custom

/ree:[VALUE]

A value that specifies which events are to be delivered for the subscription. VALUE can be true or false. When VALUE is true, all existing events are read from the subscription event sources. When VALUE is false, only future (arriving) events are delivered. The default is true when /ree is specified without a value, and the default is false if /ree is not specified.

/dmi:NUMBER

A value that specifies the maximum number of items for batched delivery in the event subscription. This option is only valid if the /cm parameter is set to Custom.

/cm:CONFIGURATION_MODE

A value that specifies the configuration mode of the event subscription. CONFIGURATION_MODE can be one of the following strings: Normal, Custom, MinLatency or MinBandwidth. The EC_SUBSCRIPTION_ CONFIGURATION_MODE enumeration defines the configuration modes. The /dm, /dmi, /hi and /dmlt parameters can only be specified if the configuration mode is set to Custom.

4.3 - Configuring the Source Coumputer

Delete this text and replace it with your own content.

4.3.1 - Install the WinRM on Source Computers

When the down-level machines are Source Computers ensure that the WinRM client is installed on these machines (refer to Downloads in the Pre-Requisites section). It is recommended that a software distribution server, such as System Center Configuration Manager (SCCM) or Systems Management Server (SMS) is used to deploy the WinRM packages.

When upgrading an Event Collector from WinRM 1.1 to WinRM 2.0 ensure that there are no active Subscriptions running otherwise the upgrade may fail.

4.3.2 - Configuring the WinRM Service

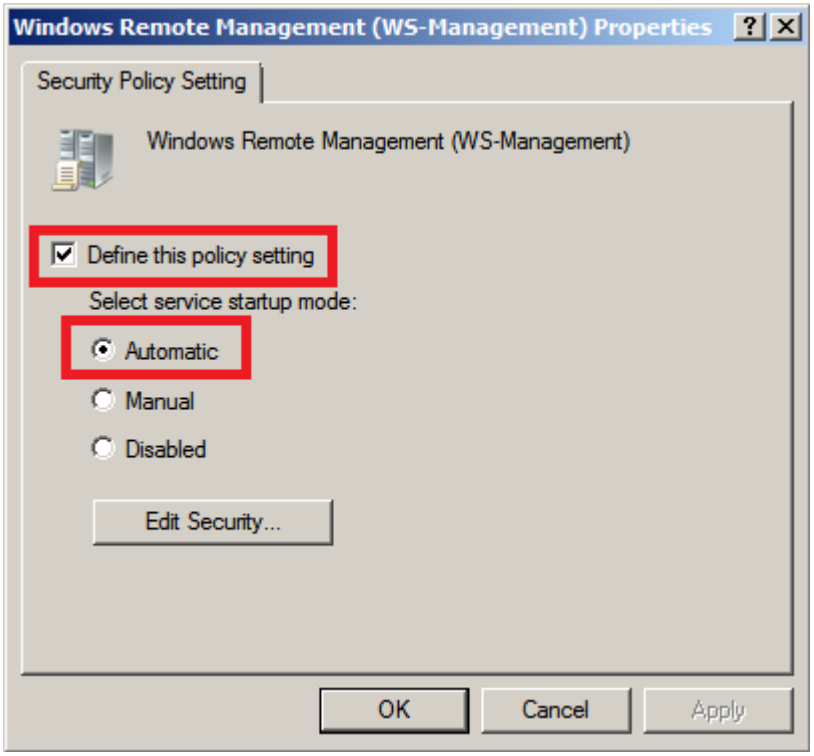
In order for Source Computers to communicate with the Event Collector machine the Windows Remote Management (WinRM) service needs to be running on the Source Computers. WinRM service auto start is necessary for the host to retrieve subscription information from event collectors and send/push event data to the event collector.

The following Group Policy Settings are used to configure WinRM to support Event Forwarding:

- Computer Configuration\Policies\Windows Settings\Security Settings\System Services

Configuration Steps:

1. Navigate to the **Windows Remote Management (WS-Management)** service.
2. Double-click the service.
3. Check **Define this policy setting**.
4. Select the **Automatic** radio button.



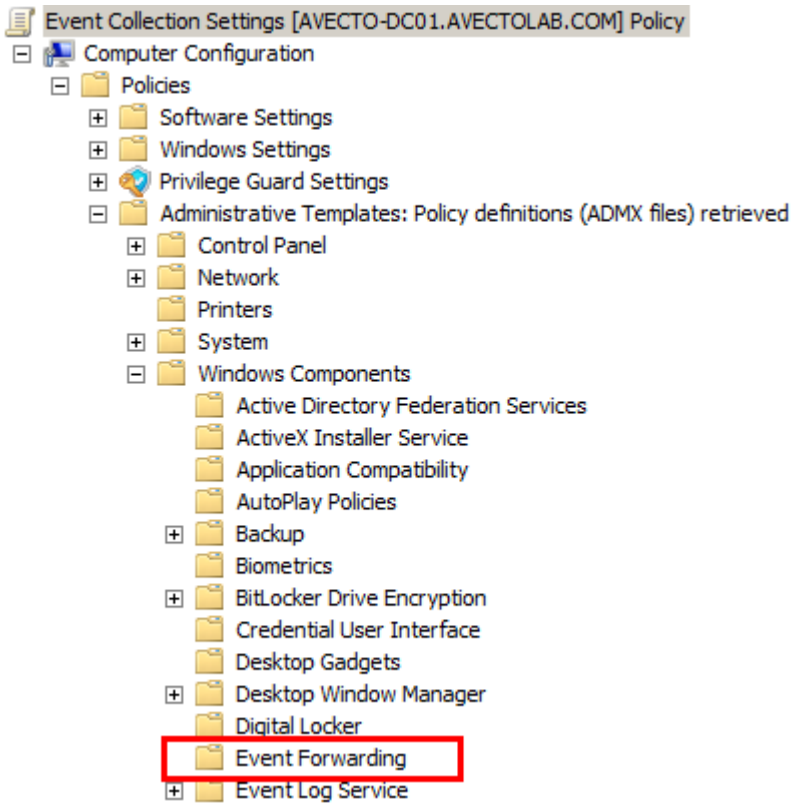
5. Click **OK**.

4.3.3 - Configuring Event Collector(s) Server Address

Group Policy may be used to configure Source Computers (Clients) to forward events to a collector (or set of collectors). The policy is very simple. It merely tells the Source Computer to contact a specific FQDN (Fully Qualified Domain Name) or IP Address and request subscription specifics. All of the other subscription details are held on the Event Collector.

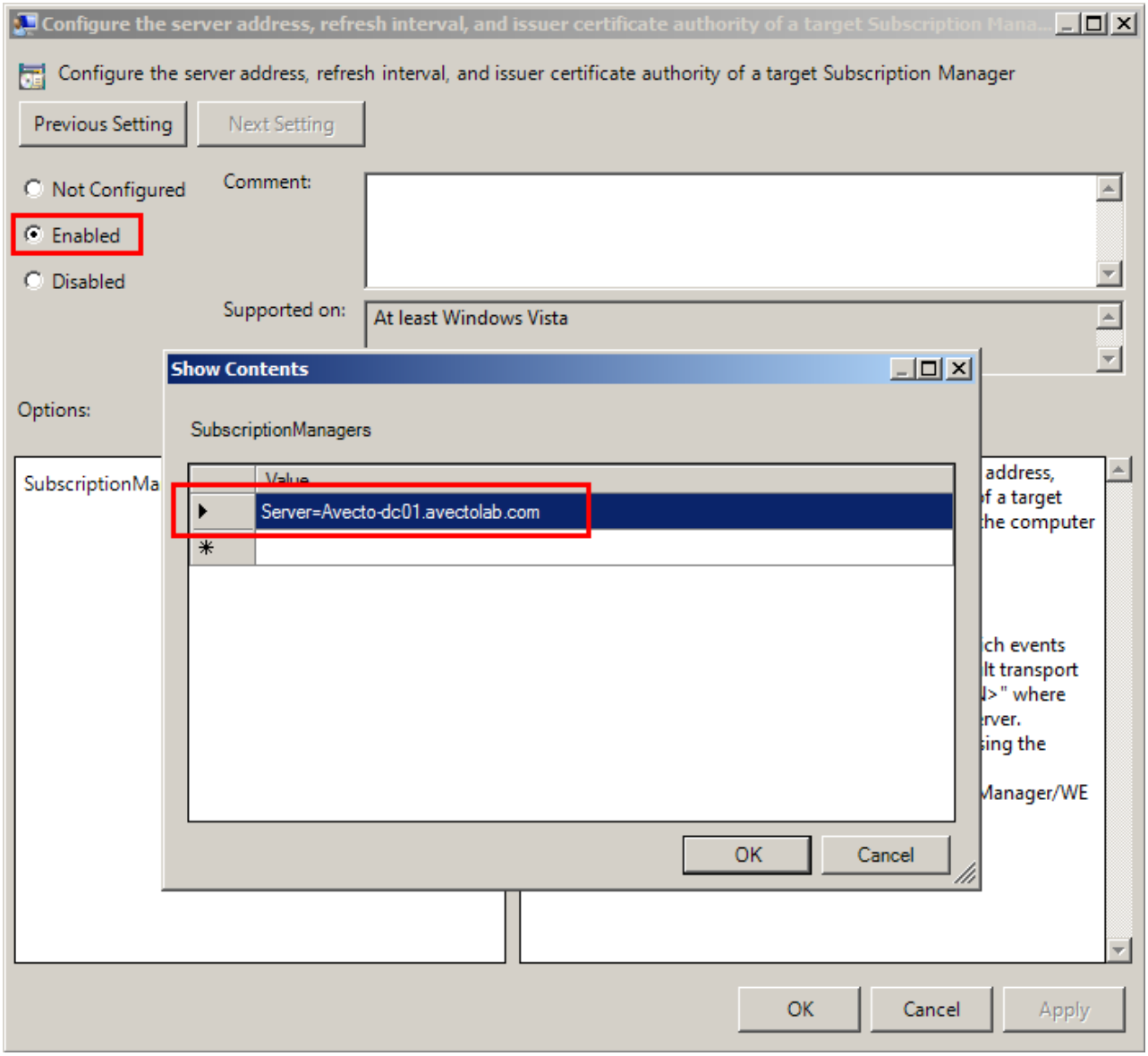
The following Group Policy Settings are used to configure event forwarding:

- Computer Configuration\Policies\Administrative Templates\Windows Components\Event Forwarding\



Configuration Steps:

1. Edit the Group Policy Object (GPO) being used.
2. Configure the **Configure the server address**… option.
3. Set this to **Enabled**.
4. Click **Show**. The **Subscription Managers** dialog will be displayed.



5. Click **Add** and enter the address of the **Event Collector**.

If the Event Collector's FQDN is Server1.avectolab.com the server address would be: Server=Server1.avectolab.com

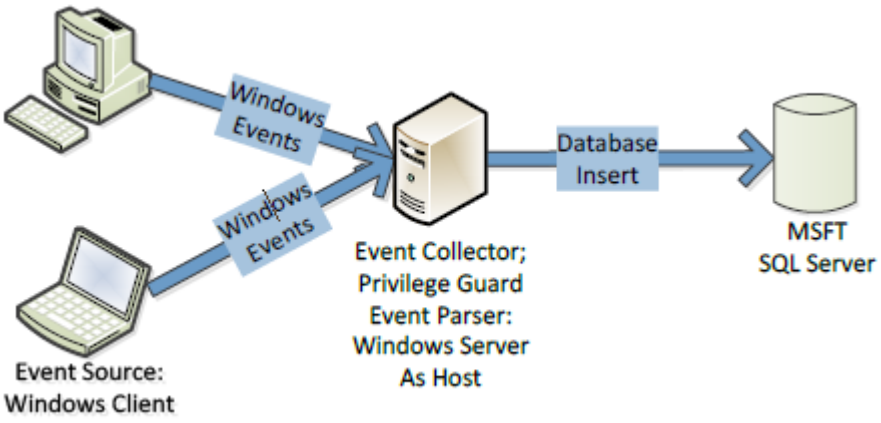
6. Click **OK**.

When editing Group Policy settings ensure that the Event Collector(s) and Source Computer(s) are under the management scope of the Group Policy Object being edited.

Chapter 5 - Event Fwd Imp Scenarios

The scenarios outlined below provide an overview of the most common Window Event Forwarding configurations, including both scaled out and fault tolerant designs.

5.1 - Basic Event Collection



The above design provides an example configuration for use within small to medium size organizations, where fault tolerance is not required.

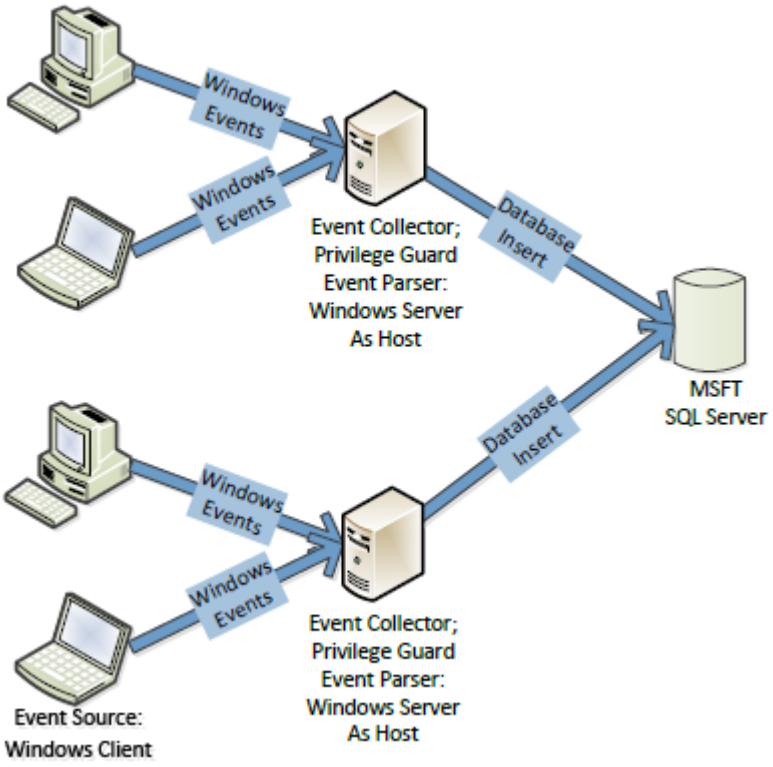
Positives

- Supports up to 100,000 source computers connecting to a single event collector.

Negatives

- Limited fault tolerance, if the Event Collector goes offline the events will be collected on the client and forwarding will resume once the event collector is back online.
- An extended fault could result in audit event loss on the client due to log rollover. This can be mitigated by large event log size.

5.2 - Scaled-Out Event Collectors



The above design provides scalability as the number of Event Collectors can be scaled-out (increased) to accommodate an unlimited number of Source Computers.

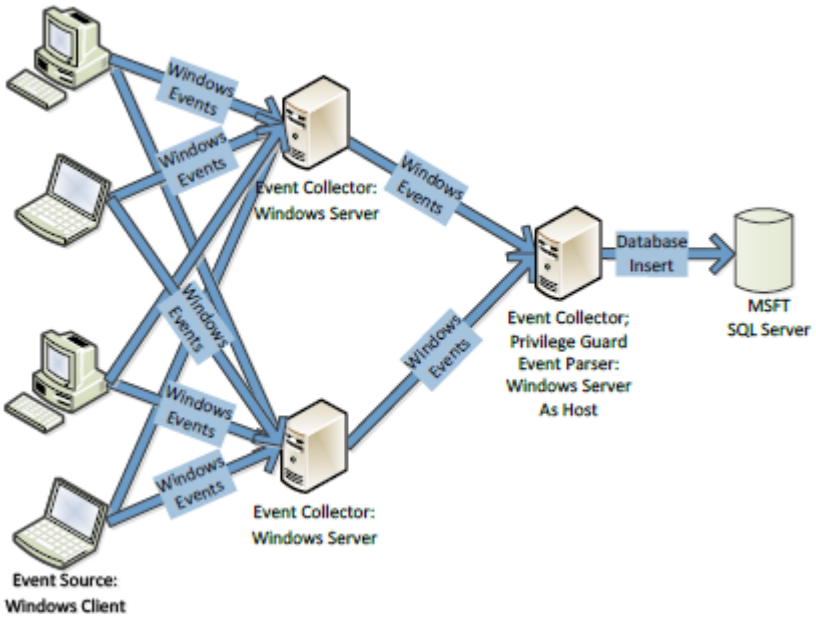
Positives

- Supports up to 100,000 source computers connecting to a single event collector.
- Supports an unlimited number of Source Computers.
- Accommodates broad geographic deployment, or network segmentation.

Negatives

- Limited fault tolerance, if the Event Collector goes off line the events will be collected on the client and forwarding will resume once the event collector is back online.
- An extended fault could result in audit event loss on the client due to log rollover. This can be mitigated by large event log size.
- Traffic to database across WAN links requires firewall configuration.
- Database insert performance may be affected by slow links.

5.3 - Scaled-Out Tiered Fault Tolerant Event Collection



The above design combines scalability and fault tolerance; Windows Event Forwarding supports fault tolerant event collection by transmitting events to duplicate event collectors. The solution consuming the events must identify duplicates and discard them, see Avecto's Enterprise Reporting pack for more information.

Positives

- Supports up to 100,000 source computers connecting to a single event collector.
- Supports an unlimited number of Source Computers.
- Accommodates broad geographic deployment, or network segmentation.
- Mitigates firewall and database performance concerns by placing 2nd tier collector proximate to database.
- Provides Fault toler

Negatives

- Limited fault tolerance, if the Event Collector goes off line the events will be collected on the client and forwarding will resume once the event collector is back online.
- Extended fault could result in audit event loss on the client due to log rollover. Mitigated by large -event log size.
- Traffic to database across WAN links requires firewall configuration.
- Database insert performance may be affected by slow links.

Specific hardware and software specifications will vary depending on the enterprise environment in which Event Forwarding is being configured. Avecto's Professional Services team can provide advice and assistance in this area if required. Please contact your account manager for more information.

Appendix A - Definitions

A.1 - Event Forwarders / Event Sources

The events you are interested in reside on these hosts.

A.2 - Event Collector

Events are collected onto these hosts based on events subscriptions defined on the collector host.

A.3 - Event Subscriptions

Determine which events are collected and defined on the event collector. Group Policy does not support definition of event subscriptions. Event subscriptions define:

- Event source hosts in scope
- Events in scope on those hosts
- Event data transmission characteristics – push from source/pull from collector, frequency, http/https

There are 2 ways for Event Source computers to become aware of event collection subscriptions.

- **Collector-initiated subscription (pull)**: subscription information is pushed to the event source hosts by the event collector via WinRM. This requires the event forwarder/source to listen for incoming WinRM connections from the collector.
- **Source-initiated subscription (push)**: the event source computer connects to the event collector via WinRM and requests subscription information. The event collector may be defined via Group Policy. Source-initiated subscription is preferred for its reliability and scalability in enterprise scenarios. A Source-Initiated subscription has an advantage of not requiring the collector to know all the computer names of the remote machines connecting to the service a priority, whereas a Collector-Initiated subscription requires the aforementioned information, which is harder to maintain.

Suited for large environments where Group Policy is available. Policy is dictated to the Source Computer via Group Policy. The Source Computer is told: "Contact Collector X and do what they say". Once the Source Computer contacts the Collector, the Collector looks up the subscription(s) for the Source Computer, and then sets up the subscription(s). Then this begins to act like a "Push" subscription.

- **Positive**: Very simple to configure via a single policy. Supports clustering of collectors. Only requires uni-directional TCP communication since the Collector never initiates communication to the Source Computer.
- **Negative**: Requires an AD infrastructure. Can be difficult to troubleshoot if the entire scope of Source Computers is successfully registered with their respective Collectors since the Collector doesn't know which Source Computers should be forwarding events to them.

Event subscriptions may not be defined via Group Policy.

A.4 - WinRM – Windows Remote Management

WinRM is the communication channel leveraged by the Windows Event Forwarders (event sources) and Windows Event Collectors.

There are 2 types of communication that take place between the hosts over WinRM:

- Event Subscriptions – which hosts are included, which events, pull or push, how much, how often
- Event Transmission – the events themselves

WinRM may act as a client or server component. It's necessary to configure WinRM as a "server" to listen for connections initiated from another host.

The host from which connections are initiated depends on the event collection/forwarding configuration chosen. In the typical configuration, connections are initiated from the forwarder/source to the collector as http or https on standard WinRM ports.

WinRM may be configured via AD Group Policy.

A.5 - Active Directory Group Policy (GPO)

Active Directory (AD) is a directory service created by Microsoft for Windows domain networks. It is included in most Windows Server operating systems.

Group Policy is a feature of the Microsoft Windows NT family of operating systems that control the working environment of user accounts and computer accounts. Group Policy provides the centralized management and configuration of operating systems, applications, and users' settings in an Active Directory environment

GPO provides a central configuration mechanism for WinRM and one aspect of Windows Event Forwarding – the event collector from which subscriptions are retrieved in source-initiated subscriptions.

Appendix B - Optional Configuration

B.1 - Optimizing Event Forwarding

B.1.1 - Forwarder Resource Usage

It is possible to control the volume of events sent to the Event Collector by the Source Computer, and this may be required in high volume environments

The following Group Policy Settings are used to configure **Forwarder Resource Usage**:

- Computer Configuration\Policies\Administrative Templates\Windows Components\Event Forwarding\ForwardResourceUsage

This GPO controls resource usage for the forwarder (Source Computer) by controlling the Events per second sent to the Event Collector. This setting applies across all subscriptions for the forwarder (Source Computer).

B.1.2 - Reducing the TCP/IP connection idle time

A Windows Server is capable of 16,000 concurrent TCP/IP connections; if your environment has more than 16,000 Source Computers connected to an Event Collector not all the machines will be able to communicate at the same time. In large enterprise environments where large numbers of Source Computers are required to connect to an event collector is it recommended the TCP/IP idle time is reduced to improve the speed at which Source Computers can connect.

It is possible to connect circa 100,000 Source Computers to a single Event Collector however in this scenario it is Microsoft recommend that the TCP/IP idle time is set to 2 minutes.

Configuration Steps:

1. Open an elevated command prompt on the Event Collector.
2. Enter net config server /autodisconnect:2
3. The “command completed successfully” should be displayed.

The purpose is to disconnect idle sessions after a set number of minutes. The valid value range is -1 to 65535 minutes. To disable Autodisconnect set it to -1.

Setting Autodisconnect to 0 does not turn it off and results in very fast disconnects, within a few seconds of idle time. (However, the RAS Autodisconnect parameter is turned off if you set it to a value of 0.)

B.1.3 - Event Log Retention

It is important to ensure the forwarded events log is off sufficient size to ensure the log does not wrap prior to the data being parsed into Avecto’s Enterprise Reporting Pack or upstream Event Collector(s). The theoretical maximum log file size for the forwarded events log on Windows Server 2008 R2 is 2 terabytes, but as the log file becomes larger the Event Viewer UI takes longer to load and show results for custom views. Depending on the size of the network, a 1GB forwarded events log file can hold anywhere from a few hours to a few days’ worth of log data. Due to this size limitation, it is important to review the log regularly and setup the appropriate size for your environment.

The above information is intended for the server acting as the Event Collector. The size of the event log on the Source Computers is less critical, however if the Event Collector is unavailable events will be collected locally and forwarded on once the Event Collector is back on line. Therefore it is recommended organizations consider the impact of an offline Event Collector(s) and set the size of the Client machine's event log accordingly.

Avecto's Professional Services team can provide best practice advice in this area.

B.2 - Configuring the Event Collector Service via Group Policy

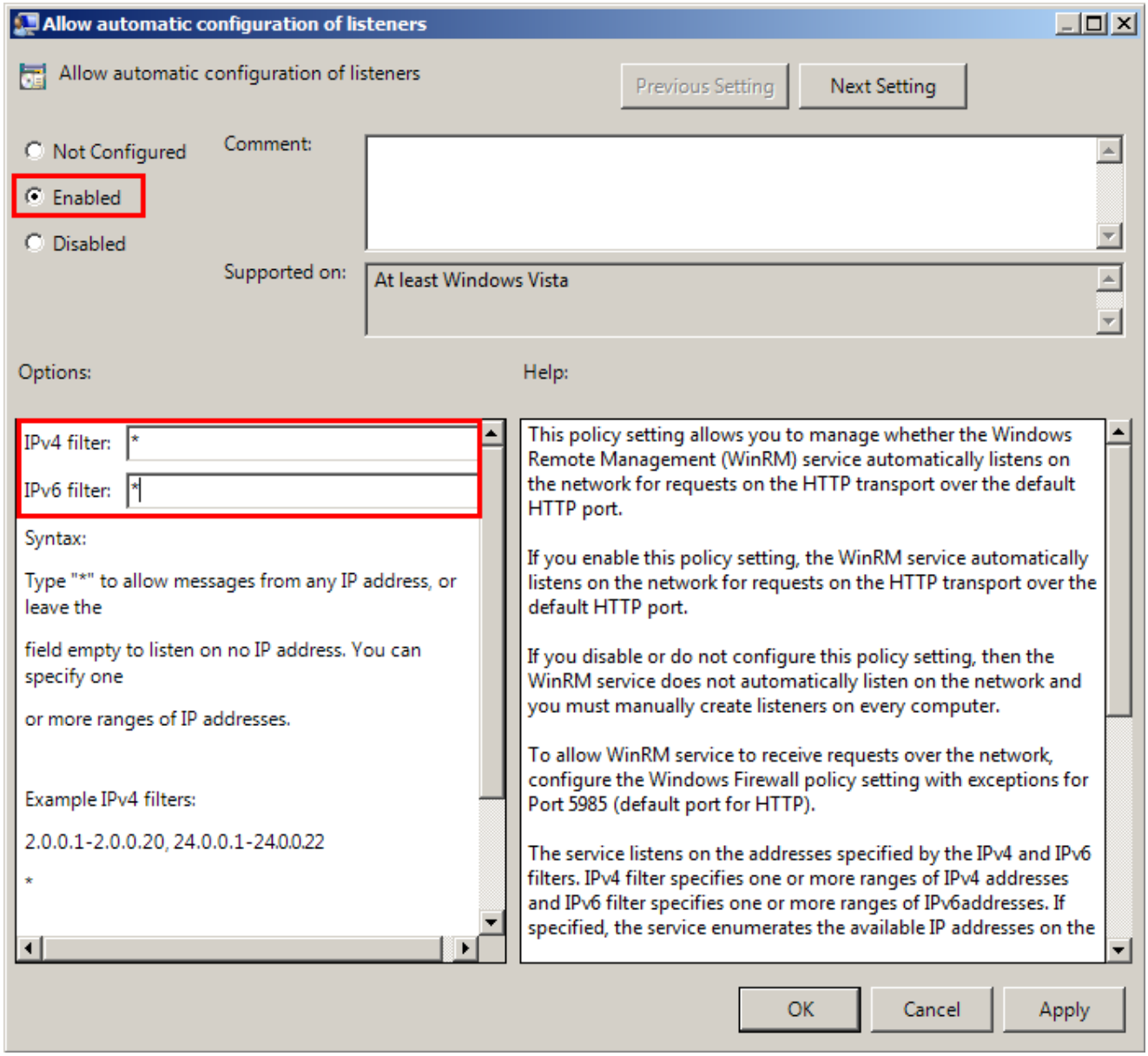
Group Policy may be used to enable and configure Windows Remote Management (WinRM). This section will focus on configuring the WinRM service to listen for incoming events. This can be configured via the following Group Policy setting:

- Computer Configuration\Policies\Administrative Templates\Windows Components\Windows Remote Management**WinRM Service**

When editing Group Policy settings ensure that the Event Collector(s) are under the management scope of the Group Policy Object being editing.

Configuration Steps:

1. Edit the Group Policy Object (GPO) being used.
2. Navigate to **./Allow automatic configuration of listeners** (see above for full path).
3. Set this to **Enabled**.
4. Specify * as the filter.



This Listener configuration should only be used in a trusted network environment. If the environment is not trusted (like the Internet), then configure only specific IP Addresses or ranges in the IPv4 and IPv6 filters.

If you are using Windows Server 2008 R2 as the Event Collector or have upgraded to Windows Remote Management 2.0 (which is recommended), then you will need to enable **Compatibility mode** to receive events from down-level clients. The following Group Policy settings are used:

- ./Turn on Compatibility HTTP Listener
- ./Turn on Compatibility HTTPS Listener

Configuration Steps:

1. Navigate to **./Turn on Compatibility HTTP Listener** (see above for full path).
2. Set this to **Enabled**.

3. Navigate to ./Turn on Compatibility HTTPS Listener (see above for full path).
4. Set this to **Enabled**.

The following command allows you to enable the compatibility listener from the command line: winrm set winrm/config/service @{EnableCompatibilityHttpListener="true"}

B.3 - Specifying the Event Collector(s) Server Address Port via Group Policy

The Event Collector's Server Address port can be configure via Group Policy, in order to do this the full URI must be specified within the address configuration of the follow GPO settings:

- Computer Configuration\Policies\Administrative Templates\Windows Components**Event Forwarding**

WinRM 2.0 Settings

- Server=http://<Event Collectors FQDN>:5985/wsman/SubscriptionManager/WEC
- Server=https:// <Event Collectors FQDN>:5986/wsman/SubscriptionManager/WEC

WinRM 1.1

- Server=http://<Event Collectors FQDN>:80/wsman/SubscriptionManager/WEC
- Server=https:// <Event Collectors FQDN>:443/wsman/SubscriptionManager/WEC

The syntax used here will depend on the WinRM version running on the Event Collector and whether HTTP or HTTPS is used. If HTTPS is being used a valid SSL certificate will be needed refer to http://msdn.microsoft.com/en-us/library/bb870973(VS.85).aspx for information configuring WinRM to utilize SSL certificates.

Additional information may be configured, the syntax of SubscriptionManagers value is:

Server=[http|https]://HOSTNAME[:PORT][/wsman/SubscriptionManager/WEC[,Refresh=SECONDS][,IssuerCA=THUMBPRINT]]

Each option for the SubscriptionManager is a comma delimited string containing the following parts:

- Server: FQDN or Hostname
- Refresh: The number of seconds to send events to the server
- IssuerCA: Thumbprint of the client authentication certificate

B.4 - Configuring WinRM Enhanced Security via Group

The security configuration is divided into two parts: service and client. The service configuration is used to manage the WinRM service that receives WS-Management requests from clients.

The following are supported authenticate methods:

- Basic Authentication
- Digest Authentication

- Credential Security
- Support Provider (CredSSP)
- Negotiate Authentication
- Kerberos Authentication
- Client Certificate-based Authentication
- Channel Binding Token

The security settings must be compatible between the client and the service The following Group Policy settings may be configured for the WinRM Client and Service:

Computer Configuration/Policies/Administrative Templates/Windows Components/Windows Remote Management/**WinRM Client**/

Computer Configuration/Policies/Administrative Templates/Windows Components/Windows Remote Management/**WinRM Service**/

It is important that these settings are compatible with your operating environment and that the WinRM Client and WinRM Service settings are compatible. Mis-configuration may stop WinRM from operating correctly.

B.4.1 - Allow Basic Authentication

This policy setting allows you to manage whether Windows Remote Management (WinRM) uses Basic authentication. If you enable this policy setting then WinRM will use Basic authentication. If WinRM is configured to use HTTP transport, then the user name and password are sent over the network as clear text.

B.4.2 - Disallow Digest Authentication

This mode of authentication is a challenge-response scheme. The client will initiate the request and in response, the server will send a server-specified token string to the client. After the token string has been received, the client will append the resource request with the username of the client, the hash of the username's password, and the token string to the response message. This method of authentication is abused by attackers using a technique called Pass the Hash. Pass the Hash is a way for an attacker to use the password hashes to authenticate as the user without ever discovering the user's actual password. The client has the option to set Digest Authentication, while the service does not. Additionally, the service can allow hardening of WinRM TLS connections using channel binding tokens.

B.4.3 - Allow CredSPP Authentication (Credential Security Support Provider)

This policy setting allows you to manage whether Windows Remote Management (WinRM) uses CredSSP authentication. Credential Security Support Provider (CredSSP) provides a secure way to delegate a user's credentials from a client to a target server. The SSP provides the capability of Single Sign-on (SSO) in Terminal Services sessions. This option is only available for WinRM 2.0.

B.4.4 - Disallow Keberos Authentication

Kerberos version 5 is used as a method of authentication and communication between the service and client. This policy setting allows you to manage whether Windows Remote Management (WinRM) will **not** use Kerberos authentication directly. If you enable this policy setting, then WinRM will not use Kerberos authentication directly. Kerberos may still be used if WinRM is using the Negotiate authentication and Kerberos is selected.

B.4.5 - Disallow Negotiate Authentication

Negotiate authentication is a Security Support Provider (SSP) that provides a client with two alternative methods for authentication: Kerberos and NTLM. This policy setting allows you to manage whether Windows Remote Management (WinRM) will not use Negotiate authentication. Negotiate will initially select Kerberos as the default; otherwise, NTLM is used.

Disabling Negotiate authentication may result in unforeseen problems when trying to configure WinRM locally. When the remote destination is the local host and the client is in the domain, WinRM uses Negotiate authentication. If an error arises stating Negotiate authentication is disabled, a Work around is to use Kerberos locally by specifying the local hostname in the remote switch. Setting the Disallow Negotiate Authentication policy to Enabled is recommended.

B.4.6 - Allow Unencrypted Traffic

This policy setting allows you to manage whether Windows Remote Management (WinRM) sends and receives unencrypted messages over the network. If you enable this policy setting, then WinRM sends and receives unencrypted messages over the network.

B.4.7 - Trusted Hosts (Client Only)

Trusted Host authentication is used for computers not using HTTPS or Kerberos for authentication. A list of computers (non-domain members) can be provided and marked trusted. If you enable this policy setting, the WinRM client uses a specified list to determine if the destination Event Collector is a trusted entity. These computers, when using WinRM, will not be authenticated.

B.4.8 - Specify channel binding token hardening level (Service Only)

A common threat amongst NTLML, NTLMv2, and Kerberos authentication methods is a Man-in-the-Middle (MitM) attack. Channel Binding Token (CBT) authentication option involves securing communication channels between a client and server using Transport Layer Security (TLS). A MitM attacker is positioned between a client and a server to impersonate as both the server and client. When the client initiates a request to the server, the attacker captures the client's first request and forwards it to the server on the client's behalf. The server responds with an authentication request. The attacker receives the server's request and forwards the request to the client. When this request is received by the client, the client sends their credentials as a response. As previously done, these credentials are sent to the attacker because the client assumes it is communicating with the server and now the attacker can access the resource. CBT ensures a secure communication channel with the client. If a MitM is being conducted, then the two connections will generate two different tokens (sessions in particular; server-to-attacker and client-to attacker). When the CBT-aware server notices this discrepancy, it will refuse the authentication request.

Channel Binding Tokens can be set to:

- None - Not using any CBTs
- Relaxed - Any invalid tokens are rejected, but any channel without a binding token will be accepted
- Strict - Any request with an invalid channel token is rejected

If Hardening Level is set to **Strict**, any request not containing a valid channel binding token will be rejected. This option is only available for WinRM 2.0.

B.4.9 - Disabling Windows Remote Shell

When WinRM completes execution of quickconfig, Windows Remote Shell (WinRS) will be enabled by default and will accept connections. This poses a potential security risk and you may wish to disable this. If the Windows

Remote Shell service is needed for a task, temporarily enable it and then disable it when the task is completed. WinRS can be disabled for domains via Group Policy. This policy enforcement applies for the collector and sources in the domain. WinRS policies can be found by navigating to:

Computer Configuration\Policies\Administrative Templates\Windows Components**Windows Remote Shell**\

To disable WinRS:

1. Set the Allow Remote Shell Access policy to Disabled
2. Click OK

WinRS can also be disabled by using the command line:

winrm set winrm/config/winrs @{AllowRemoteShellAccess=”false”}]

Parameters	Description
AllowRemoteShellAccess	Permit remote shell access
IdleTimeout	The time, in milliseconds, before a shell connection is terminated.
MaxConcurrentUsers	Maximum number of users that can request shell access at one time
MaxShellRunTime	Maximum duration, in milliseconds, that command can run for. This value is not configurable in WinRM 2.0.
MaxProcessesPerShell	Maximum number of processes that a single shell can create.
MaxMemoryPerShellMB	Maximum number of memory that a single shell can use.
MaxShellsPerUser	Maximum number of shells a user can create.

B.4.10 - Client Certificate-Based Authentication

The WinRM traffic between the collector and source is encrypted using a Windows Integrated Authentication or HTTPS. The message payload of the WinRM traffic is encrypted using one of the three authentication methods provided by Integrated Windows Authentication: Negotiate, Kerberos, or CredSSP.

WinRM with SSL requires certificates to authenticate the collector and source. Services can verify the connecting client’s authenticity by examining its certificate. If the authentication process fails, then the client’s connection is revoked.

The general steps consist of configuring the listening port, creating certificates for collectors and sources, configuring the subscription manager, creating certificates, and configuring subscriptions.

There is no Group Policy setting to disable Certificate-Based Authentication for WinRM’s client configuration. The only alternative is via the command line:

winrm set winrm/config/client/auth @{Certificate=”false”}

Configuring Window Event Forwarding to use HTTPS is beyond the scope of this document. Avecto’s Professional Services team can provide advice and assistance in this area if required. Please contact your account manager for more information.

B.4.11 - Restricting WinRM Access

The default rules permit connections from any IP address to the specific WinRM port. An attacker who has presence on a network can possibly access machines and servers by accessing WinRM services. This attack can be mitigated by customizing firewall rules to only allow connections between collectors and sources. These configurations apply to the WinRM predefined firewall rules under.

- Computer Configuration\Policies\Windows Settings\Security Settings\Windows Firewall with Advanced Security\Inbound Rules.

B.4.12 - Event Source Firewall Modifications

To enable WinRM firewall rules on the sources:

1. Right-click the predefined WinRM firewall rule and select **Properties**
2. Navigate to the **Scope** tab
3. In the Remote IP Address area and select the **These IP addresses** option
4. Click the **Add…** button
5. Select the **This IP address or subnet** option and enter the IP address of the collector
6. Click **OK**

This assumes the Microsoft Windows Firewall is being used.

B.4.13 - Collector Firewall Modification

As done in the Source Firewall Modifications section, repeat the steps for the predefined WinRM rule. Setting the **Predefined set of computers** option to **Local subnet** is recommended. This rule can be changed to best suit your environment.

B.5 - Raising Actions & Tasks Based on Collected Events

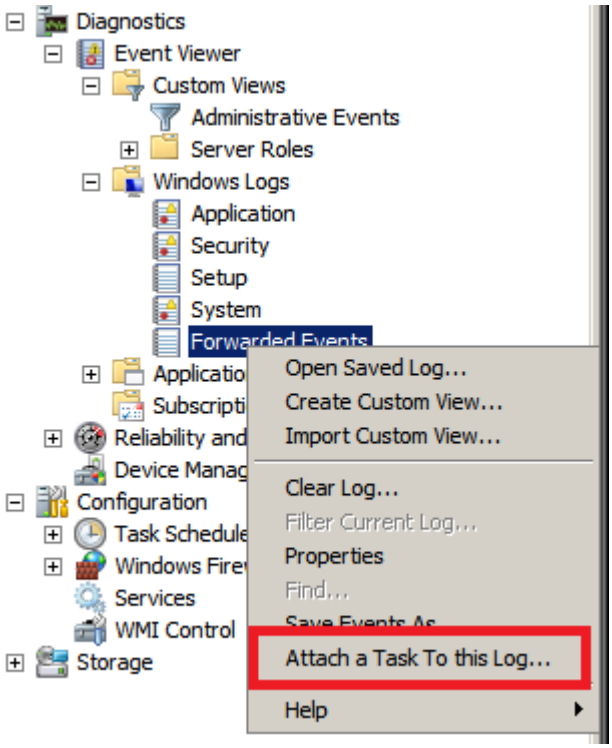
In many situations administrators or security professionals may want to be informed when a particular event is collected. It is possible to trigger the following actions by assigning a task to be in the Event Collector's forwarded events log:

- Start a program
- Email
- Display a message

For example, an administrator may want to be informed by e-mail when a user has elevated an application using the On-demand facility (Event ID 101).

Configurations steps

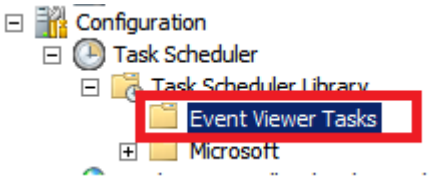
1. Open the **Event Viewer** utility on the **Event Collector**.
2. Right-click on the **Forwarded Events** log.
3. Click **Assign a Task** To this Log…

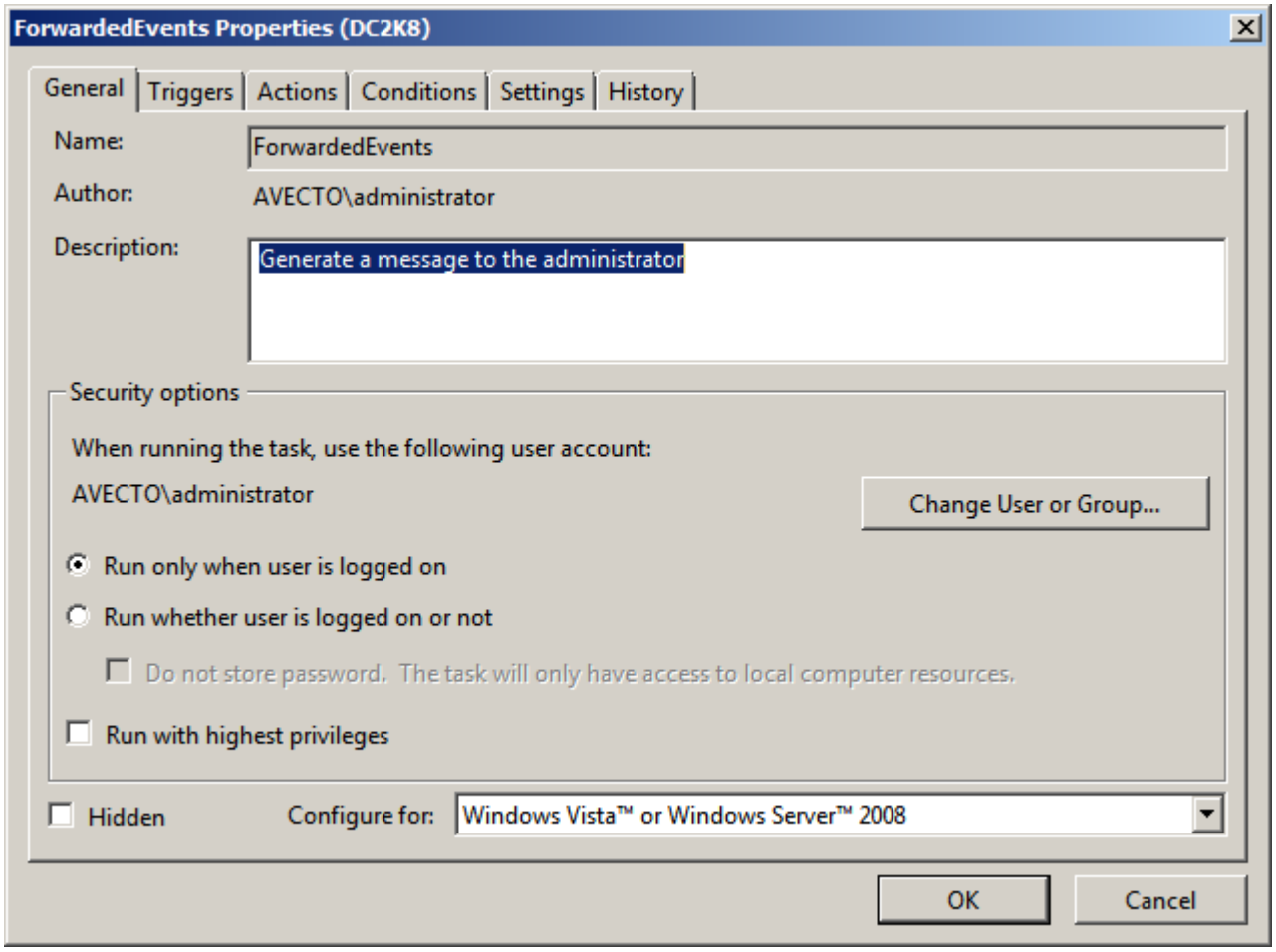


4. Give the Task a name and click **Next**.
5. Click **Next**.
6. Select the **Action** required.
7. Complete the action details and click **Next**.
8. Click **Finish** (the task is now setup).

B.5.1 - Advanced Options

It is possible to set advanced configuration options and filters by reviewing the action for the **Windows Task Scheduler** > **Event Viewer Tasks**:




ForwardedEvents Properties (DC2K8)
General
Triggers
Actions
Conditions
Settings
History
Name:
ForwardedEvents
Author:
AVECTO\administrator
Description:
Generate a message to the administrator
Security options
When running the task, use the following user account:
AVECTO\administrator
Change User or Group...
Run only when user is logged on
Run whether user is logged on or not
Do not store password. The task will only have access to local computer resources.
Run with highest privileges
Hidden
Configure for:
Windows Vista™ or Windows Server™ 2008
OK
Cancel

Appendix C - General Information

C.1 - Subscription XML Details

A subscription is simply a XML file that describes to the operating system what event logs to collect and forward. The following subscription example demonstrates the collection of Defendpoint events in the Application log from a source (client). The targeted sources are the Domain Computers group and the Domain Controllers group.

Example Subscription XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Subscription
xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
<SubscriptionId>Application Log</SubscriptionId>
<SubscriptionType>SourceInitiated</SubscriptionType>
<Description></Description>
<Enabled>true</Enabled>
<Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
<ConfigurationMode>MinLatency</ConfigurationMode>
<Delivery Mode="Push">
<Batching>
<MaxLatencyTime>30000</MaxLatencyTime>
</Batching>
<PushSettings>
<Heartbeat Interval="3600000"/>
</PushSettings>
</Delivery>
<Query>
<![CDATA[
<QueryList>
<Query Id="0" Path="Application">
<Select Path="Application">*[System[Provider[@Name='Avecto Defendpoint Event
Service'] and (Level=1 or Level=2 or Level=3 or Level=4 or Level=0) and (
(EventID &gt;= 100 and EventID &lt;= 116) )]]</Select>
</Query>
</QueryList>
]]>
</Query>
<ReadExistingEvents>false</ReadExistingEvents>
<TransportName>HTTP</TransportName>
<ContentFormat>RenderedText</ContentFormat>
<Locale Language="en-US"/>
<LogFile>ForwardedEvents</LogFile>
<PublisherName>Microsoft-Windows-EventCollector</PublisherName>
<AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
<AllowedSourceDomainComputers> O:NSG:NSD:
(A;;GA;;;DC)A;;GA;;;DD)</AllowedSourceDomainComputers>
</Subscription>
```

This subscription example is for testing purposes as it will collect a large amount of events and is not recommended for production use.

C.1.1 - Subscription Details

Node	Description
Subscription	The subscription schema
SubscriptionId	The subscription's identification
Description	Describes the subscription
Enabled	Specifies if the current subscription is enabled or disabled
Uri	The type of event used by the subscription.
ConfigurationMode	Used for the Event Delivery Optimization of subscriptions. The four valid options are: Normal, MinLatency, MinBandwidth or Custom
Delivery Mode	Indicates how events should be sent to the subscription manager. The mode can either be: Push (Source-Initiated) or Pull (Collector-Initiated)
QueryList	Used for event filtering and <Select></Select> is a XPath query
Heatbeat	Used to validate the client's connectivity with subscription
ReadExistingEvents	Notifies the subscription to read all events matching the filter
TransportName	Indicates that either HTTP or HTTPS will be used
ContentFormat	Specifies how the event data will be given to the subscription manager
Locale	Language that the response is translated too
LogFile	The event log file where the received events will be stored at
PublisherName	The name of the publisher that owns or imports the log file
AllowedSourceNonDomainComputers	List the allowed non-domain computers that can receive the subscription
AllowedSourceDomainComputers	List the allowed domain computers that can receive the subscription

C.1.2 - WS-Management Protocol Settings

Parameters	Description
MaxEnvelopeSizekb	The Simple Object Access Protocol (SOAP) data size has maximum in kilobytes Default is 150 kilobytes
MaxTimeoutms	Each push request (not pull) has a maximum timeout. This value is in milliseconds. Default is 60000ms (60 seconds)
MaxBatchItems	The limit of elements used in a pull response. Default for WinRM 1.1 and earlier: 20 Default for WinRM 2.0: 32000
MaxProviderRequests	The limit on concurrent requests. Default for WinRM 1.1 and earlier: 25 Default for WinRM 2.0: Unsupported/Undefined

C.1.3 - WinRM Client Configuration

The following parameters configure how the WinRM client operates.

Parameters	Description
NetworkDelayms	A time buffer for the client computer to wait in milliseconds. Default WinRM 1.1 and earlier: 5000 Default WinRM 2.0: 5000
URLPrefix	The type of URLPrefix used on request for HTTP or HTTPS requests. Default WinRM 1.1 and earlier: wsman Default WinRM 2.0: wsman
AllowUnencrypted	Clients are allowed to request unencrypted traffic. Default WinRM 1.1 and earlier: false Default WinRM 2.0: false
Auth	Specifies which authentication method is allowed for the client computer
DefaultPorts	Default WinRM 1.1 and earlier: HTTP = 80, HTTPS = 443 Default WinRM 2.0: HTTP = 5985, HTTPS = 5986
TrustedHosts	These trusted hosts do not need to be authenticated.

C.1.4 - WinRM Service Configuration

Parameters	Description
RootSDDL	The security descriptor for remotely accessing the listener **Default WinRM 1.1 and earlier:** O:NSG:BAD:P(A;;GA;;;BA)S:P(AU;FA;GA;;;WD)(AU;SA;GWGX;;;WD) **Default WinRM 2.0:** O:NSG:BAD:P(A;;GA;;;BA)(A;;GR;;;ER)S:P(AU;FA;GA;;;WD)
MaxConcurrentOperations	The maximum number of concurrent operations. **Default WinRM 1.1 and earlier: 100** **Default WinRM 2.0: replaced with MaxConcurrentOperationPerUser**
MaxConcurrentOperationsPerUser	The limit of concurrent operation for each user on the same system. **Default WinRM 1.1 and earlier: Not available** **Default WinRM 2.0: 15**
EnumerationTimeoutms	The idle timeout between pull messages in milliseconds. **Default WinRM 1.1 and earlier: 60000** **Default WinRM 2.0: 60000**
MaxConnections	The maximum number of simultaneous active requests that can be processed. **Default WinRM 1.1 and earlier: 5** **Default WinRM 2.0: 25**
MaxPacketRetrievalTimeSeconds	The limit on the number of seconds to retrieve a packet. **Default WinRM 1.1 and earlier: Not available** **Default WinRM 2.0: 120**
AllowUnencrypted	Clients are allowed to request unencrypted traffic. **Default WinRM 1.1 and earlier: false** **Default WinRM 2.0: false**

Parameters	Description
Auth	Specifies which authentication method is allowed for the client computer.
DefaultPorts	Default WinRM 1.1 and earlier: HTTP = 80, HTTPS = 443 Default WinRM 2.0: HTTP = 5985, HTTPS = 5986
IPv(4/6) Filter	The IP for the WinRM service to listen on. **Default WinRM 1.1 and earlier: Any** **Default WinRM 2.0: Any**
EnableCompatibilityHttpListener	Service listens on port 80 and port 5985. **WinRM 1.1 and earlier: Not supported**
EnableCompatibilityHttpsListener	Service listens on port 443 and port 5986. **WinRM 1.1 and earlier: Not supported**
CertificateThumbprint	The certificate thumb print used for https. **WinRM 1.1 and earlier: Not supported**

C.1.5 - WinRM and IIS

Windows Server 2008 R2 introduced a feature called WinRM IIS Extension. The IIS Extension allows the redirection of WinRM traffic from port 80 to port 5985 using a WinRM module. This module permits sources running WinRM 1.1 and below to communicate with a collector that is also using port 80 for web traffic. When a WinRM connection arrives on port 80, IIS will investigate the incoming URL for the prefix /wsman. This URL prefix is reversed by IIS and no configuration of IIS is needed. All GET requests to the URL prefix /wsman will be forwarded to WinRM. Microsoft recommends not hosting any site with the aforementioned URL prefix. WinRM IIS Extension is not installed by default and must be added via Server Manager.

C.1.6 - WinRM Registry Keys and Values

Registry keys which relate to the settings described above can be found in the following locations, it is not recommended that these registry key are modified, they are only listed here for verification purposes. These keys are found by viewing the following GPO Administrative Template (ADM) files located at Event Forwarding:

- EventForwarding.adm
- Windows Remote Management: windowsremotemanagement.adm
- Windows Remote Shell: WindowsRemoteShell.adm

The policies registry keys appears once a Domain Controller configures WinRM via Group Policies.

Registry Values Description	Descrip-tion
HKLM\SOFTWARE\Policies\Microsoft\Windows\EventLog\EventForwarding\SubscriptionManager\1	Subscription Manager registry key

Registry Values Description	Description
HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service\AllowConfig HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service\IPv4Filter HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service\IPv6Filter HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service\AllowBasic HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service\AllowUnencryptedTraffic HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service\AllowCredSSP HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service\AllowKerberos HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service\CBTHardeningLevelStatus HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service\CbtHardeningLevel HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service\AllowNegotiate	WinRM Service registry keys
HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Client \AllowBasic HKLM \SOFTWARE\Policies\Microsoft\Windows\WinRM\Client \AllowUnencryptedTraffic HKLM \SOFTWARE\Policies\Microsoft\Windows\WinRM\Client \AllowCredSSP HKLM \SOFTWARE\Policies\Microsoft\Windows\WinRM\Client \AllowDigest HKLM \SOFTWARE\Policies\Microsoft\Windows\WinRM\Client \AllowKerberos HKLM \SOFTWARE\Policies\Microsoft\Windows\WinRM\Client \AllowNegotiate	WinRM Client registry keys
HKLM\SOFTWARE\Policies\Microsoft\Windows\WinRM\Service\WinRS\AllowRemoteShellAccess HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WSMAN\WINRS HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WSMAN\WINRS\CustomRemoteShell	Windows Remote Shell registry keys
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WSMAN\CertMap HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WSMAN\Client HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WSMAN\Listener HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WSMAN\Listener*+HTTP HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WSMAN\Plugin HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WSMAN\Plugin\EventForwarding Plugin HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WSMAN\Service	WSMAN Services registry keys

Appendix D - Troubleshooting

If the events are not appearing on the Event Collector perform the following troubleshooting steps:

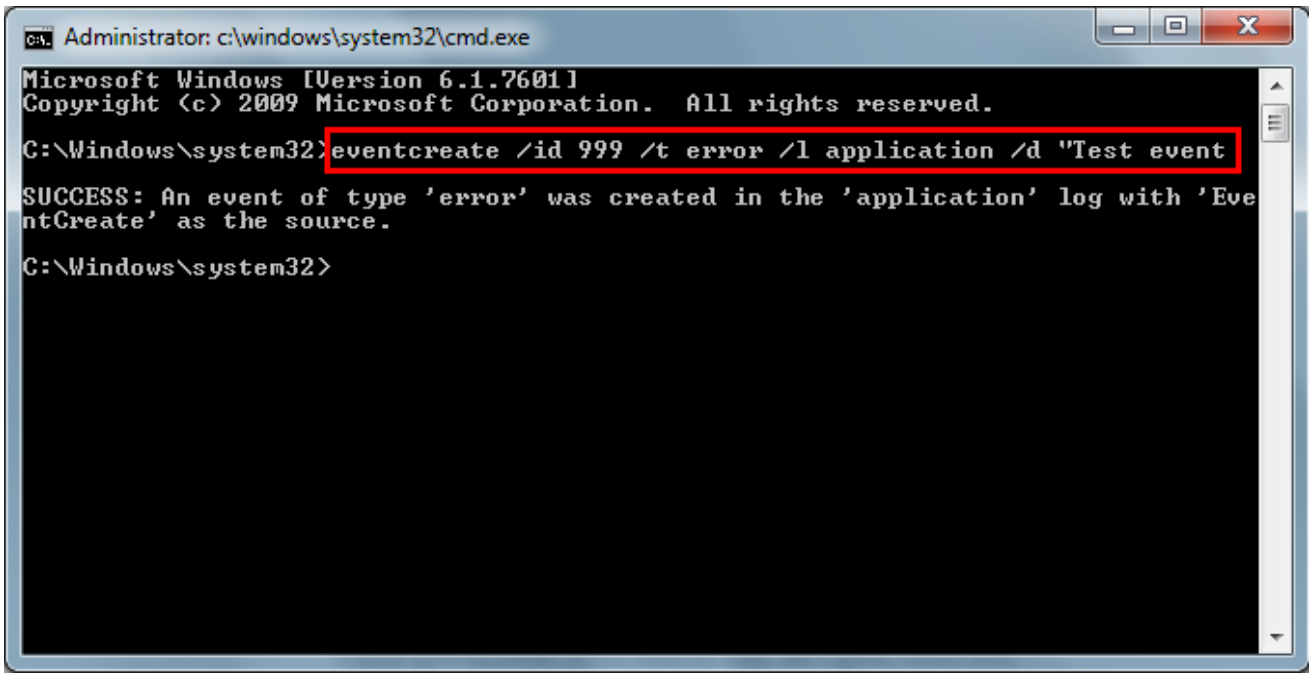
D.1 - Testing Event Forwarding

If all of the Event Forwarding components are functioning (and there's minimal network latency), a test event created on the Source Computer should arrive in the Event Collector's Forwarded Events log within 60 seconds.

On the Source Computer create a Privilege Guard event. Alternatively, if you have configured the subscription to capture all events from the application log you can use the following command line to create a test event.

1. On the **Source Computer** open a command prompt.
2. Type:

eventcreate /id 999 /t error /l application /d "Test event."



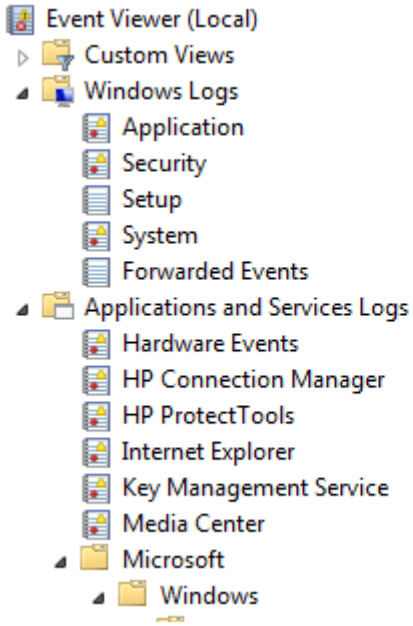
3. This event should appear on the Event Collector.

The syntax above will create an event which may not match the criteria within the previously created subscription. In order to use this test feature you will need to ensure your subscription will forward this event.

D.2 - Troubleshooting Log Locations

Event Forwarding and WinRM have operational logs that can be viewed in the Event Viewer or by using the command line tool wevtutil.exe. The following Windows logs will provide information on any errors which may have occurred:

- Down-level clients
  Windows Forwarding/Operational
- Vista Upwards
  Application and Services Logs > Microsoft > Windows
  - Eventlog-Forwarding Plugin (log)
  - Windows Remote Management (log)
  - Event Collector (log)



The **Eventlog-ForwardPlugin** and **Windows Remote Management** operational logs are the locations that the local WinRM service will log to. WinRM logs all activities to **Microsoft-Windows-Forwarding/Operational** in the Event Viewer on Windows XP.

Querying the Event Forwarding log can be done by using the Microsoft-Windows-Forwarding publisher with the command line tool wevtutil. An example of using wevtutil:

wevtutil qe "<PATH_TO_LOG>" /c:1 /rd:true /q:"<XPATH_QUERY>"

If PATH_TO_LOG is not within %SYSROOT%\system32\Winevt\Logs\, the /lf option must be used with the true argument. The /rd option cannot be used on evt files. The help documentation of the wevutil tool provides more insight of the other capabilities of the tool. This documentation can be found by executing the following command:

wevutil/?

D.2.1 - Check you can ping the Event Collector's FQDN

Ensure you can ping the FQDN of the Event Collector from the source computer:

Ping Server1.avectolab.com

D.2.2 - Check Policy has been applied to the Source Computer

This can be forced by running the following command on the Source Computer:

gpupdate /force

D.2.3 - Check Windows Remote Management Service on the Source Computer

On the source computer navigate to the services.msc and check the WinRM service is running and set to automatically.

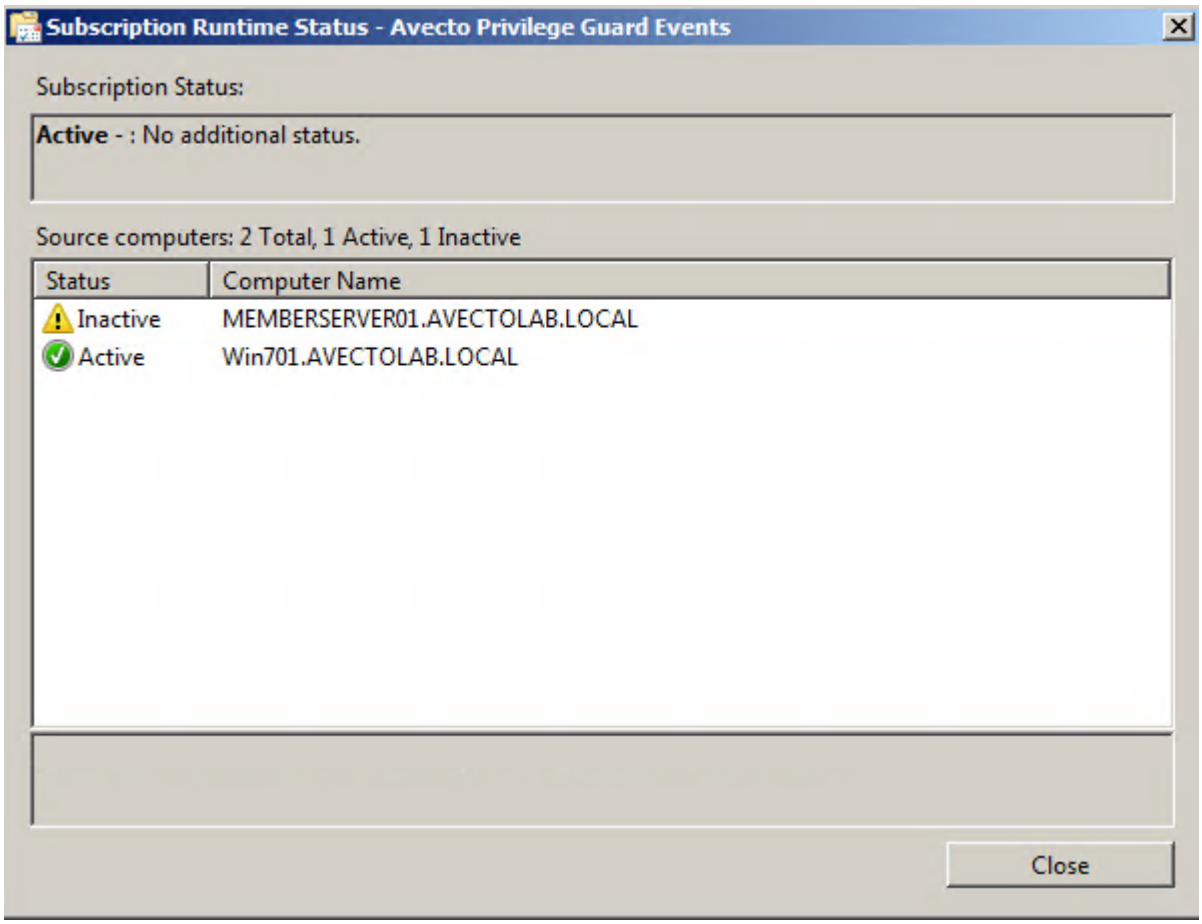
D.2.4 - Check the Collector can reach the Source Computer via WinRM

Run the following command on the Collector:

winrm id /r:<Source Computer> /a:none

D.2.5 - Check the Source Computer has successfully Subscribed

From the Event Collector you can check whether the source computer has subscribed by viewing the subscription status:



D.2.6 - Check the Collector is using the Right Credentials (Collector Initiated Only)

Run the following command on the Collector:

winrm id /r:<Source Computer> /u:<username> /p:<password>

These are the credentials defined in the Subscription on the Event Collector. The credentials don't need to be in the local administrators group on the Source Computer, as long as they are in the Event Log Readers group on the Source Computer (local administrators group will also work).

D.2.7 - Check the Source Computer has registered with the Collector

Run the following command on the Collector:

wecutil gr <subscription name>

This will list all the registered Source Computers (if the Subscription is "Collector Initiated" then this will list all configured Source Computers), their state (from the Collector's perspective), and their last heartbeat time.

D.2.8 - Check the Windows Forwarding/Operational event log on the Source Computer for errors

Event ID 105 “The forwarder is having a problem communicating with the subscription manager address” is often a result of the Windows Firewall on the Event collector blocking communication.

Ensure the following rules are accepting incoming connections:

- Windows Firewall port(s) **Windows Remote Management (HTTP-In) Port 5985** configured for inbound communication.
- Windows Firewall port(s) **Windows Remote Management (HTTP-In) – Compatibility Mode - Port 80** configured for inbound communication.
- Windows Firewall port(s) **Windows Remote Management (HTTPs-In)** configured for inbound communication.

D.2.9 - Enumerate the active WinRM Listener

The command below provides the syntax required to enumerate the active WinRM listeners on the Event Collector:

enumerate winrm/config/listener

When compatibility mode is enabled, WinRM creates a second port (80) to access its services. The approach to test if WinRM is listening on port 80 is to enumerate the listeners.

D.2.10 - View the WinRM config

The command line below provides syntax to view the WinRM configuration on the event collector:

winrm get winrm/config

These two commands display the configuration for both WinRM client and service. Viewing configuration settings can help identify any possible incorrect configuration settings.

winrm get winrm/config/client/auth

winrm get winrm/config/service/auth

D.2.11 - View remote machine details

winrm id –remote:TARGET

The above command identifies (id) the remote machine (TARGET) by asking the remote machine its operating system version and WinRM version. The TARGET can be a NetBIOS name, Domain name, or FQDN. Alternatively, using the–auth:none option will forces WinRM to not use authentication when requesting information from the remote machine. Using this option only provides a minimal set of details (version of WinRM only).

D.2.12 - View WinRM communication information

The identify option provides insight if communication between two WinRM parties are correct and not interrupted. This interruption can be the result of a firewall blocking WinRM or WinRM not running.

winrm get wmi/root/cimv2/Win32_Service?Name=WinRM

This command provides useful information (e.g., ProcessID and Context WinRM runs in) regarding the WinRM service running on the local machine.

D.2.13 - Restore WinRM Defaults

WinRM allows the restoration of default settings using the above command.

winrm invoke restore winrm/config @{}

D.2.14 - View Error Code Help

WinRM error messages display the description of the error and an error code. The definition behind the error code can be shown by executing the above command. The ERRORCODE needs to be supplied verbatim as it was displayed in the original error message (e.g., 0x80070005 means Access Denied). These errors are Win32 error codes.

winrm helpmsg ERRORCODE

D.2.15 - View Authentication Help

Generally, WinRM produces an error message when authentication fails. The service provides a second option to help the authentication process. A detailed explanation of different authentication methods used by WinRM can be viewed using the above command.

winrm help auth

Authentication Error Example: The WinRM client cannot process the request. Negotiate authentication is currently disabled in the client configuration. Change the client configuration and try the request again. If this is a request for the local configuration, use one of the enabled authentication mechanisms still enabled. To use Kerberos, specify the local computer name as the remote destination. To use Basic, specify the local computer name as the remote destination, specify Basic authentication and provide user name and password.

The recommended method to satisfy WinRM is to supply the –remote option with the target hostname (local or remote). If the source is part of a domain, then executing this command requires an uninterrupted connection to the Domain Controller. Assume the command is being executed on a computer whose hostname is ABCD.

winrm get winrm/config –remote:ABCD

D.2.16 - Access Denied Errors

Certain operations of the WinRM command may result in access denied errors. There are multiple reasons for the following error:

WSManFault

Message = Access is denied.

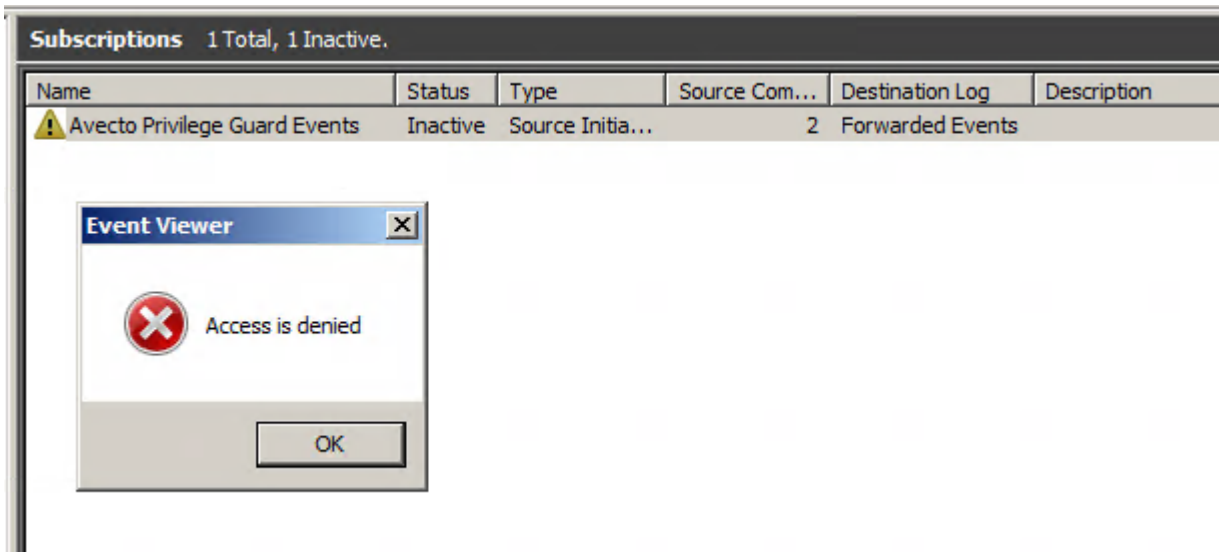
Error number: -2147024891 0x80070005

Access is denied.

- User needs to be part of local administration group, WinRMRemoteWMIUsers__, or domain administrator
  - The administrator password cannot be blank
- Incorrect username or password
- WMI operations need permissions to allow secure connections
- Windows Firewall service needs to be running (this will result in the subscription set to inactive see **Event Collector Subscription is Inactive** detailed below.

D.2.17 - Event Collector Subscription is Inactive

The Event Collectors Subscription's status is "Inactive" when a retry is initiated you may receive an access denied error:



The root cause of this problem is related to an unspecified dependency on the **Windows Firewall Service**. Please ensure the service is installed and started, you will then be able to stat the subscription.

D.2.18 - Ensure the WinRM firewall ports are open

When using third-party firewalls you will need to ensure the following ports are open on the Event Collector's Firewall:

- Windows Remote Management v2.0 over HTTP = Port 5985 #
- Windows Remote Management v2.0 over HTTPS = Port 5986
- Windows Remote Management v1.1 over HTTP = Port 80
- Windows Remote Management v1.1 over HTTPS = Port 443

If using the Windows Firewall the above ports can be configured via the GUI, alternatively the following command line options can be used to configure the firewall:

netsh advfirewall firewall set rule name="Windows Remote Management – Compatibility Mod (HTTP-In)" new enable=yes

netsh advfirewall firewall set rule name="Windows Remote Management (HTTP-In)" new

enable=yes

D.2.19 - Large Kerberos token sizes may cause Event Forwarding to fail

If your organization has large Kerberos token sizes you may experience issues with Event Forwarding, Avecto recommends following the steps listed in Microsoft's KB970875.

http://support.microsoft.com/kb/970875

D.2.20 - How to check the WinRM version you are running

http://technet.microsoft.com/en-us/library/ff520073(WS.10).aspx

D.2.21 - Creation of Subscription Errors

Common subscription creation errors:

wecutil cs Subscriptions\Logons.xml

Subscription Error Example 1 - The subscription is saved successfully, but it can't be activated at this time. Use retry-subscription command to retry the subscription. If subscription is running, you can also use getsubscriptionruntimestatus command to get extended error status.

Error = 0x3ae8. The subscription fails to activate.

This error may be caused by the WinRM Firewall exception rule being disabled. The error code that is displayed is a Win32 error code. The error code's message is shown beneath it.

Subscription Error Example 2
Failed to open subscription.
Error = 0x6b5.The interface is unknown.

This error may be caused by the Windows Event Collector not running.

Sources will create subscriptions locally after receiving a list of subscriptions applicable to them. Certain subscriptions may not be created on the sources due to permissions issues or non-existing logs. WinRM will raise an Event ID 102 with a Win32 error code of 5004 in the **Eventlog-ForwardingPlugin/Operational** log. The error

code states that a cluster resource is not available. This error code may be a result of the subscription attempting to access a log file that it does not permissions to access.

Verify the channel's (log file) permissions by navigating to: **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\Channels** and locating the channel of interest. Within the registry key of the desired channel, view the contents of the registry value named **ChannelAccess** to identify the permissions of the channel. This previous error is applicable to Windows Vista and later.

D.2.22 - XPath Query Diagnostic

XPath queries used in subscriptions do not display errors to the user who created them when deployed to sources. Query errors are shown in **the Applications and Services Logs > Microsoft > Windows > Eventlog-ForwardingPlugin > Operational** log on Windows Vista and later sources. Event ID 101 raised by the Event Forwarding plug-in is to notify the user an XPath query was incorrect::

ID	Level	Event Log	Event Source	OS Version
101	Warning (3)	Eventlog- ForwardingPlugin/Operational	Eventlog- ForwardingPlugin	Windows 7+

The human-readable details of the event do not clearly indicate the reason why the event was raised. The specific reason can be identified by viewing the XML details of the event. An error code of the XPath query is hidden as part of the event data. The error code can be viewed by:

1. Locating event ID 101 under the **Eventlog-ForwardingPlugin** > **Operational** log
2. Selecting the **Details** tab followed by selecting the **XML** view
3. Under the EventData node exists a Data node named **Status** that shows the decimal value of a Win32 error code.

A Win32 error code of 15001 indicates an invalid query of ERROR_EVT_INVALID_QUERY.

Appendix E - Additional Resources

E.1 - Configuring HTTPS

http://support.microsoft.com/kb/2019527

http://msdn.microsoft.com/en-us/library/bb870973(VS.85).aspx

E.2 - Event Subscriptions

http://go.microsoft.com/fwlink/?linkid=71431

E.3 - Source vs Collector Initiated Subscriptions

http://msdn.microsoft.com/en-us/library/windows/desktop/bb870973(v=vs.85).aspx

E.4 - Advanced Subscription Settings

http://technet.microsoft.com/en-us/library/cc749167.aspx

http://msdn.microsoft.com/en-gb/library/windows/desktop/bb736545%28v=vs.85%29.aspx (Wecutil.exe)

http://support.microsoft.com/kb/138365 (How Autodisconnect)

E.5 - Event ID Definitions

Windows Server 2000 Event log listing

http://technet.microsoft.com/en-us/library/cc952180.aspx

Windows Server 2000 Security Event Descriptions

Part 1: http://support.microsoft.com/kb/299475/en-us

Part 2: http://support.microsoft.com/kb/301677/en-us

Windows Server 2003 auditing event ID listings can be found in two locations:

Auditing Policy from Windows Server 2003: Security and Protection:

http://technet.microsoft.com/en-us/library/cc779526(v=ws.10).aspx

Chapter 4 of the Windows Server 2003 Security Guide:

http://technet.micosoft.com/library/cc163121.aspx

Windows Server 2008 and Windows Server 2008 R2 events and errors details for general OS components can be found on Microsoft's TechNet website

http://technet.microsoft.com/en-us/library/cc754424(v=ws.10).aspx

Windows Server 2008 Component-Based Servicing events

http://technet.microsoft.com/en-us/library/cc756291(v=ws.10).aspx

Windows 7 AppLocker Event IDs and definitions:

http://technet.microsoft.com/en-us/library/ee844150(v=ws.10).aspx

E.6 - Useful Links

Distributed Management Task Force, Inc: Web Services for Management (WSManagement) Specification.

http://www.dmtf.org/standards/published_documents/DSP0226_1.0.0.pdf

Microsoft Corporation: Credential Security Support Provider (CredSSP) Procotol.

http://msdn.microsoft.com/enus/library/cc226764(v=prot.20).aspx

Microsoft Corporation: Windows Error Codes. http://msdn.microsoft.com/en-us/library/cc231196.aspx

Microsoft Corporation: Web Services Management Protocol Extensions for Windows Vista.

http://msdn.microsoft.com/enus/library/cc251526(prot.20).aspx

Microsoft Corporation: Setting up a Source Initiated Subscription.

http://msdn.microsoft.com/en-us/library/bb870973(VS.85).aspx

E.7 - Software Sleuthing

http://joshpoley.blogspot.co.uk/2011/09/hresults-facilitywinrm.html

This page contains raw error codes and is meant as a software developer reference.