National Security Agency/Central Security Service

Information Assurance Directorate

Spotting the Adversary with Windows Event Log Monitoring

February 28, 2013

A product of the Network Components and Applications Division

TSA-13-1004-SG

Contents

1 Introduction 1

2 Deployment 1

2.1 Ensuring Integrity of Event Logs 2
2.2 Environment Requirements 3
2.3 Log Aggregation on Windows Server 2008 R2 4
2.4 Configuring Source Computer Policies 11
2.5 Disabling Windows Remote Shell 16
2.6 Firewall Modification 16
2.7 Restricting WinRM Access 20
2.8 Disabling WinRM and Windows Collector Service 21
2.9 Operating System Based Subscriptions 22

3 Hardening Event Collection 22

3.1 WinRM Authentication Hardening Methods 23
3.2 Security Log in Windows XP 27
3.3 Secure Socket Layer and WinRM 28

4 Recommended Events to Collect 28

4.1 Application Whitelisting 28
4.2 Application Crashes 29
4.3 System or Service Failures 29
4.4 Windows Firewall 30
4.5 Clearing Event Logs 31
4.6 Software and Service Installation 32
4.7 Account Usage 32
4.8 Kernel Driver Signing 33
4.9 Pass the Hash Detection 34
4.10 Remote Desktop Logon Detection 35

5 Event Log Retention 36

6 Final Recommendations 37

7 Appendix 37

7.1 Subscriptions 37
7.2 Event ID Definitions 49
7.3 Windows Remote Management Versions 51
7.4 WinRM 2.0 Configuration Settings 52
7.5 WinRM Registry Keys and Values 56
7.6 Installation Batch Script 57
7.7 WinRM LogOnAs Correction Batch Script 58
7.8 Troubleshooting 60
7.9 WinRM and IIS 65
7.10 Windows Server 2003 R2 66

8 Works Cited 67

List of Figures

Figure 1: Creating a Subscription 7
Figure 2: Configuring Subscription Properties 7
Figure 3: Event Delivery Optimization Configuration 8
Figure 4: Completed Subscription 8
Figure 5: Event Source GPO 11
Figure 6: Enabling Windows Remote Management 12
Figure 7: Setting Service Startup Type 12
Figure 8: Enabling WinRM listeners 13
Figure 9: WinRM listener's IP Filter Options 13
Figure 10: Enable SubscriptionManager 14
Figure 11: Configuration of SubscriptionManager 15
Figure 12: GPO Inbound Firewall Rules 18
Figure 13: Open Ports for WinRM 18
Figure 14: Allow Any Connection to Port 18
Figure 15: Verify Firewalls are Enabled 18
Figure 16: Enabling Predefined Firewall Rules for WinRM 18
Figure 17: Predefined Rule for WinRM 2.0 19
Figure 18: Adding Selective IP addresses 20
Figure 19: Add IP of Event Collector 20
Figure 20: The Event Collector Firewall allowing Local subnet to Connect 21
Figure 21: WinRM Service Authentication Policies 23
Figure 22: WinRM Client Authentication Policies 23
Figure 23: WinRM IIS Extension in Server Manager 66
Figure 24: Subscription Manager Policy Supported OS Version 67
Figure 25: WinRM Listener Policy Supported OS Version 67

List of Tables

Table 1: Vista and Later Events 9
Table 2: Windows XP Events 9
Table 3: DISA STIG Auditing Policies Recommendations 27
Table 4: Windows 7 Whilelisting Events 29
Table 5: Windows XP Whitelisting Events 29
Table 6: Windows 7 Application Events 29
Table 7: Windows XP Application Events 29
Table 8: Windows 7 System Events 30
Table 9: Windows XP System Events 30
Table 10: Windows 7 Firewall Events 31
Table 11: Windows XP Firewall Events 31
Table 12: Windows 7 Log Activity Events 32
Table 13: Windows XP Log Activity Events 32
Table 14: Windows 7 Software and Service Events 32
Table 15: Windows XP Software and Service Events 32
Table 16: Windows 7 Account Activity Events 33
Table 17: Windows XP Account Activity Events 33
Table 18: Windows 7 Kernel Driver Signing Events 34
Table 19: Subscription XML Description 39
Table 20: WinRM Version Correlation 51

Table 21: WinRM Version Update URLs 52
Table 22: Protocol Settings 54
Table 23: WinRM Client Configuration 54
Table 24: WinRM Service 56
Table 25: WinRS Configuration Settings 56
Table 26: WinRM, WinRS, WSMAN and Event Forwarding Registry Values 57
Table 27: XPath Errors based on OS Version 65

Disclaimer

This Guide is provided "as is." Any express or implied warranties, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the United States Government be liable for any direct, indirect, incidental, special, exemplary or consequential damages (including, but not limited to, procurement of substitute goods or services, loss of use, data or profits, or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this Guide, even if advised of the possibility of such damage.

The User of this Guide agrees to hold harmless and indemnify the United States Government, its agents and employees from every claim or liability (whether in tort or in contract), including attorneys' fees, court costs, and expenses, arising in direct consequence of Recipient's use of the item, including, but not limited to, claims or liabilities made for injury to or death of personnel of User or third parties, damage to or destruction of property of User or third parties, and infringement or other violations of intellectual property or technical data rights.

Nothing in this Guide is intended to constitute an endorsement, explicit or implied, by the U.S. Government of any particular manufacturer's product or service.

Trademark Information

This publication has not been authorized, sponsored, or otherwise approved by Microsoft Corporation.

Microsoft®, Windows®, Windows Server®, Windows Vista®, Active Directory®, Windows PowerShell™, AppLocker®, Excel® are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

1 Introduction

It is increasingly difficult to detect malicious activity, which makes it extremely important to monitor and collect log data from as many useful sources as possible. This paper provides an introduction to collecting important Windows workstation event logs and storing them in a central location for easier searching and monitoring of network health.

The focus of this guidance document is to assist the United States Government and Department of Defense administrators in configuring central event log collection and recommend a basic set of events to collect on an enterprise network using Group Policy.

Many commercially available tools exist for central event log collection, but this paper focuses on using the built-in tools already available in the Microsoft Windows operating system (OS). Central event log collection requires a Windows server operating system version 2003 R2 or above. Using a Windows Server 2008 R2 or above server is recommended. There are no additional licensing costs for using the event log collection feature. The cost of using this feature is based on the amount of additional storage hardware needed to support the amount of log data collected. This factor is dependent on the number of workstations within the log collection network.

The Windows operating system includes monitoring capabilities and logs data for most activities occurring within the operating system. The vast amount of logged events does not make it easy for an administrator to identify specific important events. This document defines a recommended set of events to collect and review on a frequent basis. The recommended set of events is common to both client and server versions of Windows. Product specific events, such as Microsoft Exchange or Internet Information Services (IIS), are not discussed in the document, but should be centrally collected and reviewed as well.

This guidance document is broken into three main parts. The first part, Deployment, focuses on configuring and deploying central log collection; the second part, Hardening Event Collection, concentrates on security hardening; the last section, Recommended Events to Collect, describes recommended events that should be collected. If a third party commercial product is already being used within the organization to centrally collect events, then skip ahead to the Recommended Events to Collect section. Review the recommended events and ensure they are being collected.

During the development of this guide, testing was done using Windows 7 and Windows XP Service Pack 3 (SP3) clients running Windows Remote Management (WinRM) 2.0. A Windows 8 client with WinRM 3.0 was also briefly tested and found to work the same as Windows 7. Windows Server 2008 R2 was used for the central event collection server. Configuration of Windows Server 2012 should work identically to Windows Server 2008 R2, but was not tested for this guide.

2 Deployment

The Windows Collector service has the ability to collect specific events from all domain computers for viewing on a single computer. The Event Forwarding feature of the Windows Collector Service has the ability to retrieve or receive events from remote computers. Event Forwarding can operate as Collector-

Initiated (pull) or Source-Initiated (push), respectively. The server archiving the events is a collector and the remote computer, where events are collected from is the source. 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. The Windows Collector service uses Web Services-Management (WS-Management, WS-Man) Protocol to communicate between sources and collectors. [1]

2.1 Ensuring Integrity of Event Logs

Prior installing and using the WinRM feature, several tamper-proof techniques should to be practiced. There is no software available to prevent an attacker from modifying event logs or preventing the recording of event data. An Access Control List (ACL) can be used to ensure the integrity of Windows events logs.

The Windows operating system uses permissions to ensure that certain log files are not modified by a normal user, members of an unprivileged group or members of a privileged group. The Defense Information Systems Agency (DISA) Security Technical Implementation Guides (STIG) recommends that an Information Assurance Officer (IAO) create an auditor's group and grant members of the group full permissions. If there is no IAO, it is still advised for a system administrator to create an auditor's group. The **Administrators** group's privileges must be reduced from **Full** to **Read** and **Execute** permissions for the Application, System and Security log files. [2][3][4] This single defense can be circumvented in multiple ways so, a defense in depth approach should be taken.

This guide does not discuss site specific auditor's group for WinRM purposes beyond this section. However, this should not deter the use of WinRM. The auditor's group is used to regulate who is permitted to operate on an event log file. Windows Vista and later created an **Event Log Readers** group whose purpose is to regulate access to the local event logs. [17] The auditor's group needs to be a member of the **Event Log Readers** group to access the event logs. In the case of Windows XP, there is no **Event Log Readers** group. The use of the auditor's group does not affect the configuration or the use of WinRM.

Several domain policies can be enabled to enforce restrictions of users and groups accessing event logs. DISA STIGs recommend enabling the **Manage auditing and security log** policy and configuring the policy for the auditor's group. [2][3][4] The policy is located under **Computer Configuration > Policies > Windows Settings > Local Policies > User Rights Assignment**. This policy creates a whitelist of users or groups who can access the audit log (security log). Enabling this policy does not affect WinRM operations.

The **Prevent local guests group from accessing application log**, **Prevent local guests group from accessing security log**, and **Prevent local guests group from accessing system log** policies are recommended to be enabled for Windows XP machines. [3] The policies are located under **Computer Configuration > Policies > Windows Settings > Security Settings > Event Log**. These policies are not applicable to any version of Windows later than Windows Server 2003.

---

[1] http://technet.microsoft.com/en-us/library/cc774957(v=ws.10).aspx
[2] DISA STIG: Windows Server 2008 R2 Member Server Security Technical Implementation Guide Version 1. Group ID (Vulid): V-1077, V-1137, V-26496, V-26489
[3] DISA STIG: Windows XP Security Technical Implementation Guide Version 6. Group ID (Vulid): V-1077, V-1095, V-1103, V-1137
[4] DISA STIG: Windows 7 Security Technical Implementation Guide Version 1. Group ID (Vulid): V-1077, V-1137, V-26496, V-26489

A policy, named **Generate security audits**, can be used to create a whitelist of users or groups permitted to write to the audit log. The policy is located under **Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > User Rights Assignment**. Only allow **Local Service** and **Network Service** as these are the default values of the policy. [2][3][4]

Administrators can use the Enhanced Mitigation Experience Toolkit (EMET) to heighten the security defense of machines and applications used in a network. [5] EMET provides the ability to enable and enforce specific enhanced security features for the operating system and applications. The WinRM service is hosted by svchost.exe (service host). The service host executable should have all security features enabled for an application. Enabling EMET for svchost.exe on Windows XP or Windows 7 does not prevent WinRM from working correctly. Using EMET on a default installation of Windows will not prevent the operating system from performing specific operations. However, site-specific software needs to be first tested with EMET to ensure compatibility.

Using a dedicated server whose primary role is an event collector is recommended. There should be no additional roles tasked to the event collector. Deploying WinRM on a dedicated machine helps protect it from having been previously compromised or infected with malware.

2.2 Environment Requirements

Windows Remote Management is available in multiple versions. The recommended minimal version of WinRM is 2.0. WinRM 2.0 is installed by default with Windows 7 and Windows Server 2008 R2. There are no additional updates needed for Windows 7 and Windows Server 2008 R2 for WinRM.

WinRM is not installed by default on Windows XP and is not included in any of its service packs. The installation of WinRM 2.0 requires an update provided by Microsoft for Windows XP SP3. Since the update requires Microsoft Windows Installer 3.1, Windows XP Service Pack 2 (SP2) cannot install WinRM 2.0. Microsoft Windows Installer 3.1 is installed automatically, when upgrading to Windows XP SP3. Applying the WinRM 2.0 update additionally requires .NET Framework 2.0 Service Pack 1 (SP1).

WinRM 2.0 is part of the Windows Management Framework core package. The KB968930 [6] update installs PowerShell 2.0 along with WinRM 2.0. This update requires the machine to have .NET Framework 2.0 SP1 or later to install PowerShell. The complete list of applicable Windows operating systems versions and the download location for the updates can be found in the Windows Remote Management Versions section.

WinRM 3.0 is the latest version and is only supported on Windows 7 and Windows Server 2008 R2. [7] The specific versions of WinRM installed by default are detailed further in the Windows Remote Management Versions section.

For Windows XP SP3 clients, an administrator must download the Windows Remote Management update executable and .NET Framework 2.0 update from Microsoft and use a batch script to install the

---

[5] https://www.microsoft.com/en-us/download/details.aspx?id=29851
[6] http://support.microsoft.com/kb/KB968930
[7] http://www.microsoft.com/en-us/download/details.aspx?id=34595

updates. A batch script is provided in the Installation Batch Script section to complete this task. The administrator should place the executable files at a network path that is accessible to workstations and readable by domain users.

The test environment focuses on three roles in the domain: the domain controller, the event collector, and the event sources. All policies configured through Active Directory are restricted to computer groups, rather than the default Authenticated Users group, for Group Policy Object (GPO) security filtering. The domain controller, collector, and each source in the domain should have the latest updates from Microsoft. This guide focuses on Windows 7 and Windows XP SP3 clients; however, the recommendation is to use Windows 7.

2.2.1 Administrator's Quick Environment Setup

This section summarizes the steps for installing WinRM on machines that require it.

Preparation Steps:

1. Identify the operating system versions in the targeted environment.
    a. Windows 7 workstations do not require any updates or actions.
    b. Windows XP workstations require the WinRM 2.0 update and the .NET Framework 2.0 SP1 update.
        i. Ensure Windows XP machines have upgraded to Service Pack 3.
        ii. Download the WinRM 2.0 and the .NET Framework 2.0 SP1 updates from Microsoft. [72][74]
        iii. If an alternative option is not available to deploy executables to clients, then use the batch script in Installation Batch Script section.
            1. Create a network share that all machines can access and place the WinRM 2.0 and .NET Framework 2.0 SP1 updates there.
            2. Set the first parameter of the startup script to the full directory path of where the updates are stored (e.g., Z:\updates).
        iv. Restart the client machines.

2.3 Log Aggregation on Windows Server 2008 R2

A single dedicated server should have the role of event collector in a local subnet. Isolation of the event collector avoids confusion, frustration of troubleshooting, and security related concerns. Source-Initiated subscriptions can be configured for clients to be in the same or different domain of the collector. The focus of this guidance document is using Source-Initiated subscriptions, where the collector and sources are in the same domain, and configuring event collection locally. Event collection capabilities can be configured via the GPO as well. The only issue with GPO method is that the Windows Event Collector service will not be configured for using subscriptions. The proceeding sections cover local configuration of WinRM and the Windows Event Collection service.

On the domain controller, create a GPO for the event collector. To create and link a GPO:

1. Open **Group Policy Management** in Server Manager
2. Navigate to **Group Policy Management > Forest > Domains > Domain**
3. Right-click the domain and select **Create a GPO in this domain, and Link it here…**

In Group Policy Management, the newly created GPO for the event collector server must have the **Enforce** and **Link Enable** options enabled.

Create two new groups: EventSources and EventCollectors. These groups associate each computer in the domain with the appropriate role. The EventCollector GPO applies to the EventCollectors group and the EventSource GPO applies to the EventSources group. The members of the GPOs are computer objects. Use groups containing domain computers as opposed to individual computers. If the machine was powered on when added to the group, then the newly added group member requires a reboot for it to be notified of its membership.

2.3.1 Locally Configuring Collector Settings

The event collector needs to be configured to automatically start the Windows Event Collector and Windows Remote Management services. Enabling these services sets the startup type to Automatic (Delay Start). Delay start states that the service will be started after other auto-start services are started plus a short delay. [8] The Windows Remote Management and Windows Event Collector services are automatically configured when using the quickconfig option (discussed in next section). Enabling these services through a GPO is possible, but using a GPO to configure the event collectors does not add a firewall exception. Locally configuring the event collector is recommended. Configuration of the collector can be completed by a domain administrator or a built-in administrator. The recommendation is to use a domain administrator account. The DISA STIG discourages using the built-in Administrator account. It is required that the local administrator and the domain administrator do not have a blank password.

2.3.1.1 Enabling Windows Remote Management

The WinRM command provides an option to automatically configure WinRM. The quick configure (qc) option starts the WinRM service, configures the service to be Delay-Start, creates a listener using any IP address, and enables a firewall exception for WinRM. [9] The port used by WinRM depends on the installed version of WinRM. Port 5985 is used by WinRM 2.0 whereas port 80 is used by WinRM 1.1. To configure WinRM, open a command console with administrator privileges and type:

winrm qc

Enter **y** to have the service status changed to Delay-Start. As an alternative option, all prompts can be suppressed by supplying the –q (quiet) option. Enter **y** to create a listener

An **Access Denied** error may appear when attempting to use quickconfig. A possible reason for this error is the account executing the WinRM command does not have the proper permissions. If the account is a member of the local administrator group, then User Account Control (UAC) filtering prevents access to the WinRM service. [10] Log in as a Domain Administrator account and repeat the quick config command.

The quick configuration option enables WinRM to listen on port 5985 since this guide recommends at least using WinRM 2.0 on all clients. In a network with both WinRM 2.0 and WinRM 1.1, enabling a compatibility listener is recommended. This can be done by typing:

winrm set winrm/config/service @{EnableCompatibilityHttpListener="true"}

---

[8] http://msdn.microsoft.com/en-us/library/windows/desktop/ms685155(v=vs.85).aspx
[9] winrm qc -?
[10] http://msdn.microsoft.com/en-us/library/aa384423.aspx

The compatibility listener binds WinRM to a second port (80) and accepts traffic on this port. Once a WinRM 1.1 client has established a connection with the collector, all ensuing traffic will be redirected to port 5985.

Certain collectors may be running Internet Information Services (IIS) along with WinRM. WinRM can easily share port 80 with IIS. See the WinRM and IIS section for more information.

2.3.1.2 Enabling Windows Event Collector

The Windows Event Collector service has a quick configure (qc) option similar to WinRM's quick configure option. Windows Event Collector service's quick configure option sets the service startup type to Delay-Start and enables the ForwardedEvents channel. [11] The quick configure option is only available for Windows Vista and above. To configure the Windows Event Collector Service:

wecutil qc

Enter **y** to have the service started and the status changed to Delay-Start. Similar to the WinRM command line, all prompts can be suppressed by the /q:true option.

2.3.1.3 Creating Event Subscriptions

Subscriptions are used to organize event collection and where the events come from. An administrator can have custom subscriptions to tailor event logs to easily identify interesting events. A custom subscription can be created by using the GUI or from the command line. Custom subscriptions are discussed in the next section.

The event viewer, shown in Figure 1, allows the configuration of a subscription. Subscriptions can be configured to specify the destination of received logs, the computer groups being collected, the event's ID, and the frequency of event collection. Each subscription can be configured in the Subscription Properties window shown in Figure 2. The Event Viewer console should be opened with administrator privileges. To create a subscription:

1. Open **Event Viewer** (**eventvwr.exe**)
2. Select **Create Subscription…** from the **Actions** panel
3. Provide a **Subscription name**
4. Select the **Source computer initiated** option
5. Select **Computer Groups…** button
    - Click the Add **Domain Computers…**button and enter the group name **EventSources**
    - Click **Check Names** and verify the group name is correct
    - Click **OK**
6. Click **OK**

[11] wecutil qc -?

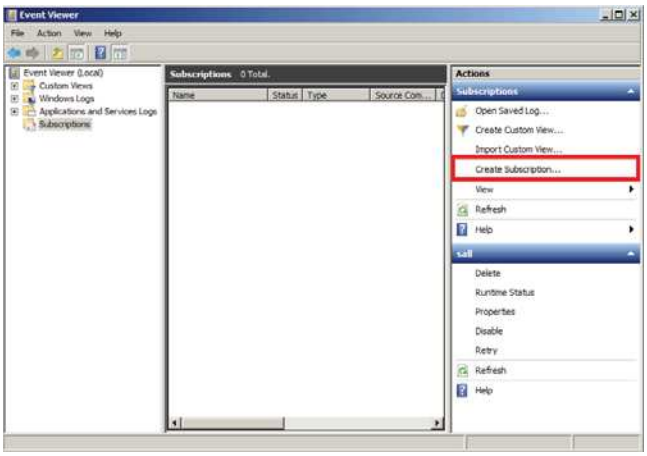


Figure 1: Creating a Subscription

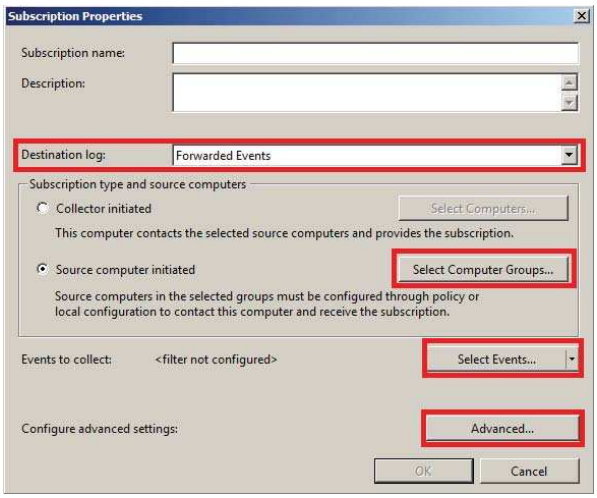


Figure 2: Configuring Subscription Properties

If an error message box appears stating "**the type initializer for 'AdvanceSettings' threw an exception**", then the current account does not have the correct permissions.

Collected Events are stored at a local predefined log location under the **Destination log** drop-down list. The default is **Forwarded Events**.

In the **Query Filter** window, displayed by clicking the **Select Events** button, a variety of events can be chosen for collection based on the event level, origination of log, and event source. Once the setup of filtering events is completed, the XML view of the selected events can be viewed in the **XML** tab. It is possible to edit the XML manually by selecting **Edit query manually** checkbox**.**

7. Click the **Select Events…** button
8. Select **Event Level** options and select all levels
9. Select **By Log**
10. From the drop-down list select…
    a. **Windows Logs > Application**
    b. **Windows Logs > System**
11. Click the **OK** button

The remaining configuration options do not need to be customized as the default setting will collect all events, keywords, task category, and from all users and computers. Any fine-grained customizations to specify the event to collect are discussed in the next section.

The configuration of advanced subscription settings sets the frequency of events being received (forwarded).

12. Click the **Advanced…** button
13. Select **Normal**
    - Leave the protocol drop-down list set to HTTP
14. Click the **OK** button

The Event Delivery Optimization options shown in Figure 3 permits the collection of event logs in 15 minutes (Normal), 6 hours (Minimize Bandwidth), or 30 seconds intervals (Minimize Latency).[12] A custom interval can be set using the wecutil command line utility.

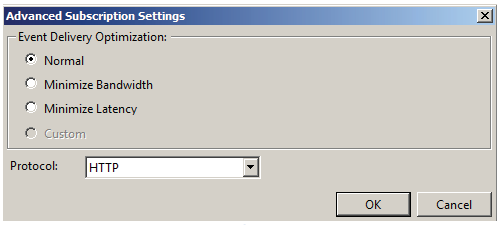


Figure 3: Event Delivery Optimization Configuration

Figure 4: Completed Subscription

2.3.1.3.1 Custom Subscriptions

The general approach to creating subscriptions using the graphical user interface lacks flexibility for custom configuration. It may be desirable to customize the frequency of event delivery and the batch amount of a subscription. A detailed description of the subscription schema is found in the Subscription section of the Appendix.

Customization of subscriptions depends on the administrator's needs and requirements. Several subscriptions have been created and provided in the Subscriptions section of the Appendix. These subscriptions collect events that an enterprise may be interested in collecting from domain computers. The following tables summarize the event IDs and the category they represent for each recommended subscriptions. The Recommended Events to Collect section discusses these events in more detail.

Each subscription focuses on account activity, application and computer failures, computer and applications modification, and security notifications.

Windows Vista and above Events

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

General Event Descriptions	General Event IDs
User Account Locked out	4740
Non-Kerberos Logon Activities	4624 and 4625
Application Crashes	1000
Application Hang	1002
Windows Error Reporting	1001
Blue Screen of Death (BSOD)	1000 and 1001
Crypto Informational Logs	90
Windows Defender Errors	1005, 1006, 1008, 1010, 1012, 1014, 2001, 2003, 2004, 3002, 5008
Windows Integrity Errors	3001, 3002, 3003, 3004, and 3023
EMET Crash Logs	1 and 2
Windows Firewall Logs	2004, 2005, 2009, 2033
MSI Packages Installed	1022 and 1033
Windows Service Manager Errors	7022, 7023, 7024, 7026, 7031, 7032, 7034, and 7045
Windows System Logs	6, 11, 219, 104, 1125, 1127, 10016, 40964, 40968
User Account Added to Privileged Group	4732
Windows File Protection Errors	1, 2, 3, 4, 5, 6, 7, 9, 10, 13, 14, 15, 16, 17
AppLocker and SRP Logs	865, 8003, 8004, 8006, 8007
Windows Update Errors	16, 20, 24, 25, 31, 34, 35
Kernel Driver Signing Errors	5038, 6281

Table 1: Vista and Later Events

Windows XP Events

General Event Descriptions	General Event IDs
User Account Locked out	644
Non-Kerberos Logon Activities	528, 540, 529, 530, 531, 532, 533, 534, 535, 536, 537, 539
Application Crashes	1000 and 1004
Application Hang	1002
Windows Error Reporting	4097
Blue Screen of Death (BSOD)	1001, 1003
Windows Defender Errors	1005, 1006, 1008, 1010, 1012, 1014, 2001, 2003, 2004, 3002, 5008
EMET Crash Logs	1 and 2
Windows Firewall Logs	851, 852, 854
MSI Packages Installed	1022, 11707, 11728
Windows Service Manager Errors	7022, 7031, 7032, 7034
User Account Added to Privileged Group	636
SRP Logs	865
Windows Update Errors	16, 11708

Table 2: Windows XP Events

The logging of cryptographic activities is not available prior to Windows Vista. [13] The alternative option to monitor cryptographic activities for pre-Windows Vista operating systems is to use CAPIMON from the Microsoft Download Center. [14] CAPIMON only monitors CryptoAPIs calls.

2.3.1.4 Creating Custom Views

Large amounts of event data are difficult to organize and view in a meaningful way. The Event Viewer allows users to create custom views that organize event data based on a custom filter. Each view can be used to represent a subscription to help identify events collected using the subscription. Custom Views were introduced in Windows Vista; therefore, this feature is unavailable in Windows XP. [15]

Custom Views should be created on the event collector where all event data is forwarded. To create a custom view:

1. Open Event Viewer and select **Custom Views** in the left panel
2. Right-click and select **Create Custom View…**
3. From the drop-down list titled **Logged**, select a time (e.g., **Last 7 days**)
    a. If a granular time range is needed, select **Custom range …** from the **Logged** drop-down list
4. Select an appropriate **Event level**
5. Select **By log** and select **Forwarded Events** from the **Event logs** drop-down list
6. Enter **Event ID(s)** in the first text area
7. Click **OK**
8. In the **Save Filter to Custom View**, provide a custom view name representing the data being filtered

This creates a custom view under **Custom Views** in the left panel of the Event Viewer. The newly created custom view will not be neatly organized under **Custom Views**. Custom views can be organized by navigating to **%ProgramData%\Microsoft\Event Viewer\Views** and creating a new sub-directory. This newly created directory should have a meaningful name such as “Last 24 hours” to indicate the time period of the events filtered. Creation of the sub-directory requires a privileged account.

To display the new directory when it does not appear after creation under **Custom Views**:

1. Select **Custom Views** in the left panel of the Event Viewer
2. Select **Refresh** in the right panel

Using a directory named “Last 24 hours,” all custom view XML files within the directory should filter events on the condition that the event occurred within the last 24 hours.

An example of a custom view may appear as the following:

---

[13] http://blogs.technet.com/b/pki/archive/2006/12/16/the-easy-way-of-crl-troubleshooting-in-windows-vista.aspx
[14] http://www.microsoft.com/en-us/download/details.aspx?id=1223
[15] http://technet.microsoft.com/en-us/magazine/2006.11.eventmanagement.aspx

```
<ViewerConfig>
        <QueryConfig>
                <QueryParams>
                        <Simple>
                                <BySource>False</BySource>
                                <Channel>ForwardedEvents</Channel>
                                <Level>2</Level>
                                <RelativeTimeInfo>1</RelativeTimeInfo>
                                <EventID>1000</EventID>
                        </Simple>
                </QueryParams>
                <QueryNode>
                        <Name>AppCrash</Name>
                        <QueryList>
                                <Query Id="0">
                                        <Select Path=ForwardedEvents">*[System[(Level=2) and (EventID=1000) and TimeCreated[timediff(@SystemTime) &lt;= 3600000]]]</Select>
                                </Query>
                        </QueryList>
                </QueryNode>
        </QueryConfig>
</Viewerconfig>
```

The preceding XML looks for events containing EventID 1000 at the Error level (Level 2) that occurred in the last hour (3600000 milliseconds).

The preceding steps focused on automatically creating an XPath query to select event data. This does not allow customization of the XPath queries. Manual XPath queries can be entered in the **XML** tab of the **Create Custom View** dialog.

2.4 Configuring Source Computer Policies

Event forwarding policies can be applied to Windows XP and above sources with no additional local configuration. The default policies allow reading of the default log files except for the Security log. The ability to forward the security log will be discussed in the Hardening Event Collection section.

2.4.1 Creating Source Group Policy Objects

Following the configuration of the collector, the collector should currently be in a waiting state to receive events from the sources. The sources are configured similarly with the exception that the Windows Event Collector service does not need to be started and each source needs to be able to read their own event logs. As done on the collector Group Policy Objects (GPO), create a GPO for the event sources with both **Enforced** and **Link Enable** applied.

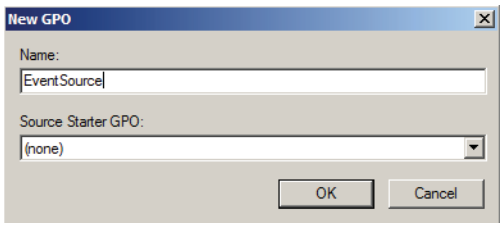


Figure 5: Event Source GPO

2.4.2 Enabling Windows Remote Management Policy

Unlike the approaches used for configuring the collector, WinRM and Event Forwarding will be managed via GPO. Configuring WinRM does not require manually executing the quick configure option. WinRM can be started using a **System Service** policy. The only issue that may arise is enabling the predefined WinRM firewall rule. Previously, the quick configure option automatically enabled this firewall rule. Active Directory provides predefined WinRM firewall rules to avoid executing the WinRM command manually on all source computers. Configuration of firewall rules are discussed later.

The WinRM service can be found by navigating to **Computer Configuration > Policies > Windows Settings > Security Settings > System Services > Windows Remote Management (WS-Management)** in Group Policy Management Editor.

To set the service to automatic:

1. Right-click the **Windows Remote Management (WS-Management)** service and select **Properties**
2. Select the **Define this policy setting** checkbox
3. Select the **Automatic** option
4. Click the **OK** button

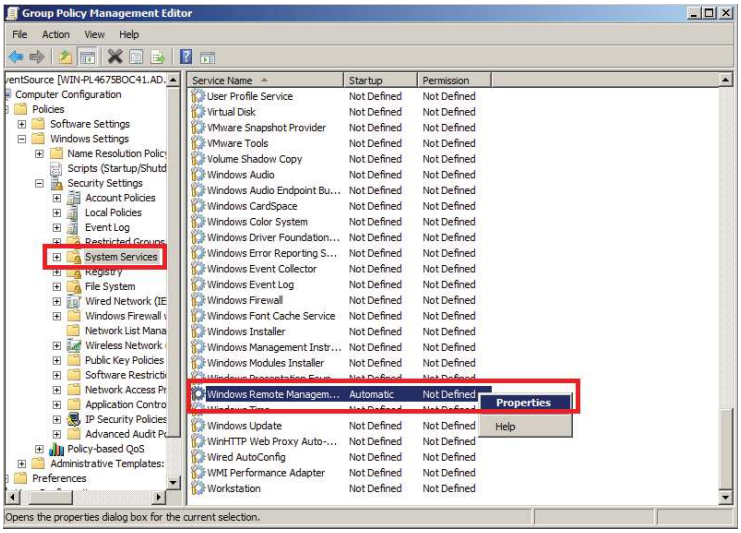


Figure 6: Enabling Windows Remote Management

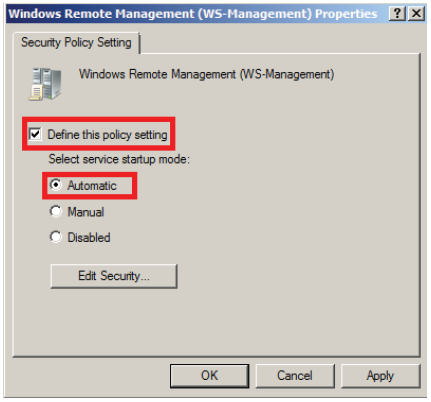


Figure 7: Setting Service Startup Type

Navigate to the WinRM policies located at **Computer Configuration > Policies > Administrative Templates > Windows Components > Windows Remote Management > WinRM Service** in the Group Policy Management Editor.

WinRM requires listeners to be available for inbound connections. The **Allow automatic configuration of listeners** policy shown in Figure 9 instructs WinRM to create listeners on port 5985 or port 80 depending on the WinRM version.

To enable WinRM listeners:

1. Set the **Allow automatic configuration of listeners** policy to **Enabled**
2. Set both **IPv4** and **IPv6 filter** value to ***

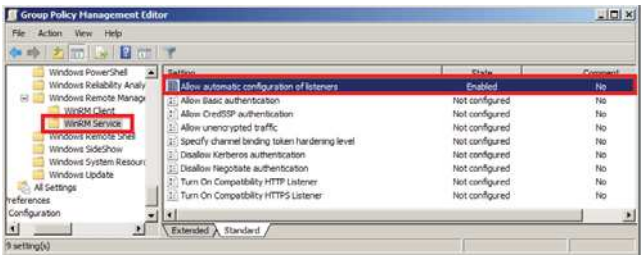


Figure 8: Enabling WinRM listeners

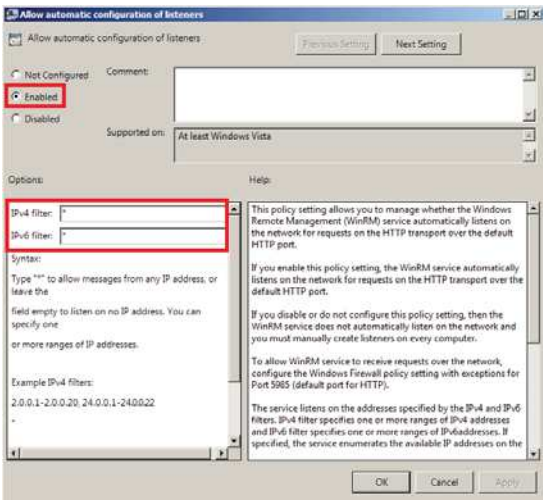


Figure 9: WinRM listener's IP Filter Options

Within the **Allow automatic configurations of listeners** dialog, the **IPv4/IPv6 filter** values should be set to *. This ensures that WinRM starts running and listens on the "any" IP address (IPv4 is 0.0.0.0 and IPv6 is "::") for both protocols. The IPv6 filter is not required to enable a WinRM listener. Enabling an IPv6 listener is an administrative decision. The WinRM service only listens on an IPv4 address when no IPv6 address (or *) is supplied for the filter.

2.4.3 Enabling Event Forwarding Policy

The source needs to be configured to forward events to the targeted subscription manager. The subscription manager (collector) hosts all the subscriptions created on the collector. The source needs to contact the manager to retrieve the list of subscriptions. These subscriptions specify the events to forward. Once the source gathers all the events pertaining to these subscriptions, the events will be delivered to the collector.

The **Configure the server address, refresh interval, and issuer certificate authority of a target** policy sets the configuration settings on how to communicate with the collector. This policy sets the collector's internet address, how often to send events to the collector, and a thumbprint of the client's certificate if using HTTPS. This policy must be enabled to forward events.

Event Forwarding is the main component for enabling event monitoring in an enterprise. Event Forwarding policies can be located by navigating to **Computer Configuration > Policies > Administrative Templates > Windows Components > Event Forwarding**.

To enable Event Forwarding:

1. Set the **Configure the server address, refresh interval, and issuer certificate authority of a target Subscription Manager** policy to **Enabled**
2. Click the **Show…** button

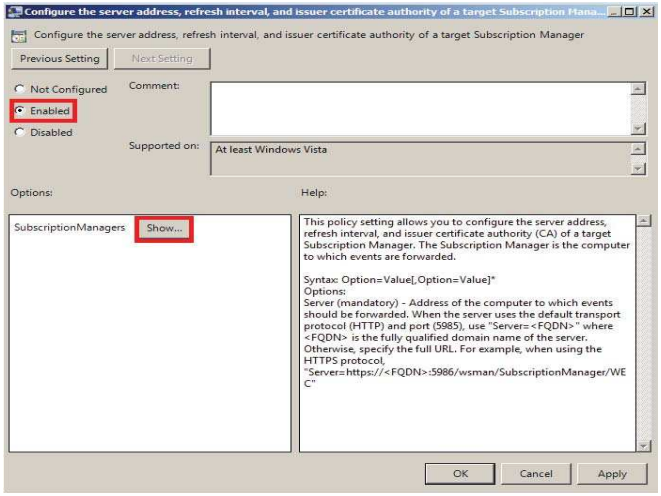


Figure 10: Enable SubscriptionManager

The **SubscriptionManagers** dialog has several options that can be set to configure event forwarding. The only requirement of this policy is to set the **Server** option. Any additional options can be omitted. 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[16]

Figure 11 shows an example Subscription Manager value. The refresh interval should be determined by administrative requirements. Using the default refresh interval is recommended.

In a network solely using WinRM 2.0, the **Server** option needs to specify port 5985, otherwise it will send traffic to port 80.

Server=http://HOSTNAME:5985/wsman/SubscriptionManager/WEC

When both WinRM 2.0 and WinRM 1.1 are intermixed and the collector has enabled compatibility mode, remove the explicit port from the Subscription Manager Uniform Resource Locator (URL).

Server=http://HOSTNAME/wsman/SubscriptionManager/WEC

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

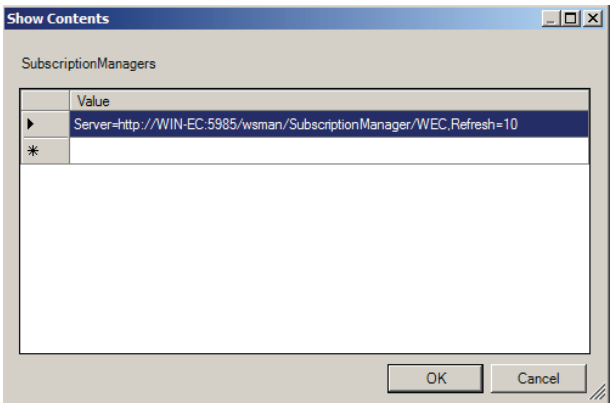


Figure 11: Configuration of SubscriptionManager

Once the **SubscriptionManager** value has been set, click **OK**.

WinRM and the Server option

WinRM will attempt to connect to the collector on port 80 regardless of version. The collector may not accept WinRM client connections on port 80. A compatibility listener for WinRM can be configured to tell WinRM to additionally listen on port 80. A caveat to enabling this option is that an additional port will be open on the server, which is a potential security concern. As an alternative, the subscription manager address should explicitly specify port 5895 for WinRM 1.1 and WinRM 2.0 sources to communicate with a WinRM 2.0 port. This avoids the creation of an additional port and firewall rules.

In the initial configuration, Windows 7 sources are not permitted to read event logs (e.g., Application, Security, Setup and System). The sources need to add the **Event Log Readers** group to Restricted Groups in the EventSource GPO. Restricted groups can be configured by navigating to **Computer Configuration > Policies > Windows Settings > Security Settings > Restricted Groups** in Group Policy Management.

To add the **Event Log Readers** to the Restricted Group Policy:

1. Right-click **Restricted Groups**
2. Select **Add Group…**
3. In the **Add Group** dialog box, click the **Browse…** button
4. Enter **Event Log Readers** in the text area of the **Select Groups** dialog box
5. Click **Check Names**
6. Once **Event Log Readers** appears, click **OK**

The Event Log Reader group can be added locally as an alternative option. In Computer Management, add Network Service to the **Event Log Readers** group. The **Event Log Readers** group is not part of the default groups in Computer Management, but can be added by navigating to **Computer Management > Local and User groups > Groups > Event Log Readers**.

The members of the **Event Log Readers** group are permitted to read event logs. To read the event logs, the Network Service account needs to be added to the **Event Log Readers** group. WinRM runs with Network Service permissions on Windows 7 and Windows XP.

To add the Network Service account to the **Event Log Readers** group:

1. Right-click **Event Log Readers** group and select **Properties**
2. In **Event Log Readers Properties**, select **Add…** in the **Members of this group** section

3. Select **Browse…** and enter **NETWORK SERVICE** in the text area
4. Select **Check Names**
5. Once **NETWORK SERVICE** appears, click **OK**
6. Click **OK** in **Event Log Readers Properties**

The **Event Log Readers** group will be shown in its SID format (S-1-5-32-573), rather than as an easily readable name, until a Windows Server 2008 or Windows 2008 R2 Domain controller has been made the Primary Domain Controller Operations Master role holder of the domain.[17]

For additional Organizational Units (OUs) that contain user workstations, previously created GPOs can be applied against those OUs.

2.5 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 critical security risk and should be disabled for all servers and clients in the domain. If the Windows Remote Shell service is needed for a task, temporarily enable it and then disable it when the task is completed. The registry keys for WinRS can be found in the WinRM Registry Keys and Values section of the Appendix. 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"}

2.6 Firewall Modification

Event collection aids in identifying problems from a remote computer using WinRM. The communication channel opens an additional attack vector on each of the sources and collectors. The role of event forwarding is solely to communicate with the collector. An attacker may attempt to attack or perform reconnaissance of other machines laterally with WinRM services. The isolation of sources and collectors limits an attacker from using this service as a target.

Certain environments may enforce firewall rule merging restrictions for servers. Enforcing these restrictions will hinder the configuration of locally applied WinRM firewall rule exceptions. The removal of rule merging restrictions is encouraged for the collection server.

WinRM should have configured Windows Firewall to allow WinRM connections when using quickconfig. The EventSource GPO firewall policies should be enabled for all profiles. This section serves as a list of

[17] http://support.microsoft.com/kb/243330

alternate methods to enable WinRM firewall exceptions. Windows Firewall with Advanced Security policy should be enabled for all profiles.

Windows Firewall with Advanced Security policies will not be applied to Windows XP clients. The feature is only available on systems running Windows Vista and above. [18] Windows Firewall policies targeting Windows XP are discussed in Windows XP WinRM 2.0 Clients.

2.6.1 Collector Firewall

In Windows Server 2008 R2, Windows Firewall with Advanced Security has two predefined firewall rules that can be enabled from the GUI or the command line. The first predefined rule, **Windows Remote Management (HTTP-In)**, allows network traffic to the local port 5895 on the collector for machines running WinRM 2.0. The second predefined rule, **Windows Remote Management – Compatibility (HTTP-In)**, allows traffic from WinRM 0.5 and WinRM 1.1 to communicate with the collector on port 80. The use of the WinRM compatibility firewall rule should be enabled when a compatibility listener is configured on the collector. These predefined firewall rules should only be enabled for the domain profile only.

2.6.1.1 Graphical User Interface

Windows Firewall with Advanced Security can be managed using two available options: local or group policies. These graphical options are not required since configuration of the firewall was performed during the WinRM setup.

2.6.1.1.1 Windows Firewall with Advanced Security Group Policy

The creation of a firewall policy for WinRM can be set using a predefined rule. Expand **Computer Configuration > Policies > Windows Settings > Security Settings > Windows Firewall with Advanced Security > Windows Firewall with Advanced Security – ADsPath > Inbound Rules**.

To enable WinRM firewall rules:

1. Right-click on **Inbound Rules** and select **New Rule…**
2. Select **Windows Remote Management** from the **Predefined** drop-down list
3. Click the **Next** button
4. Select **Windows Remote Management – Compatibility Mode (HTTP-In)** or **Windows Remote Management (HTTP-In)** depending on environment setup. Select both rules if the network is intermixed with WinRM 2.0 and WinRM 1.1 clients.
5. Click the **Next** button
6. Select **Allow the connection**
7. Click **Finish**

The predefined WinRM rule permits either WinRM 2.0 traffic (port 5985) or compatibility mode traffic (port 80). The option to enable the WinRM rule in compatibility mode or not depends if the environment is intermixed with WinRM 2.0 and WinRM 1.1 clients.

[18] http://technet.microsoft.com/en-us/library/cc748991(WS.10).aspx

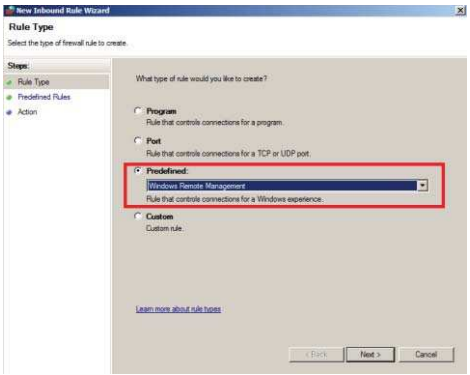


Figure 12: GPO Inbound Firewall Rules

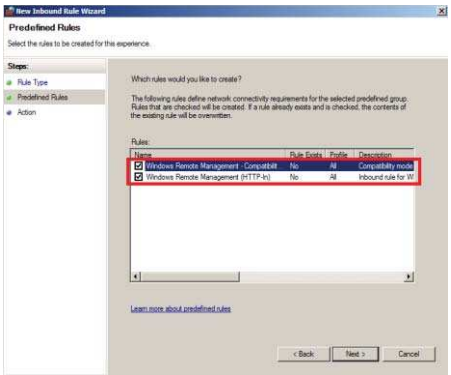


Figure 13: Open Ports for WinRM

The last configuration step for creating the new rule is allowing the connection. Windows Firewall will enable these rules for all profiles and accept traffic from any IP (remote and local) by default.

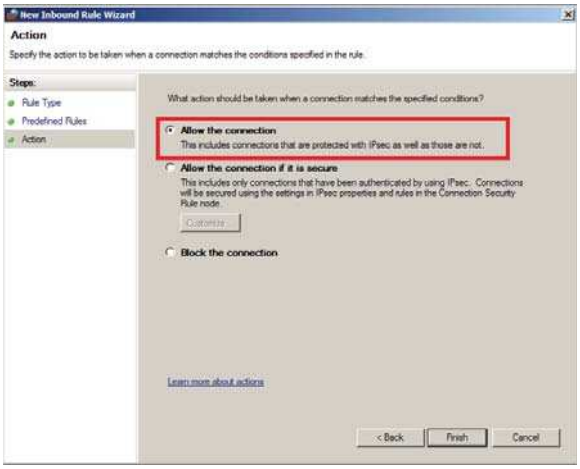


Figure 14: Allow Any Connection to Port

Figure 15: Verify Firewalls are Enabled

2.6.1.1.2 Locally Applied Firewall Rules

WinRM predefined firewall rules can also be enabled locally without a GPO. It is not required to have physical access to the collector as Remote Desktop is available to configure the firewall settings.

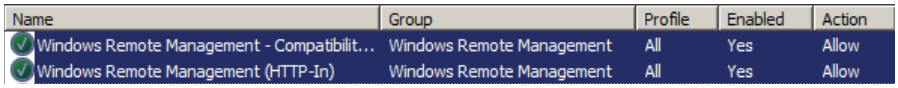


Figure 16: Enabling Predefined Firewall Rules for WinRM

2.6.1.2 Configuring the Firewall using the Command Line

The benefit of executing a firewall command allows the user to avoid navigating through the GUI to find the desired configuration options. The following commands demonstrate how to enable WinRM firewall rules for compatibility mode or non-compatibility mode respectively:

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

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

If an error message "A specific value is not valid" appears, verify the rule's name. The alternative approach is to enter the netsh context, followed by the advfirewall context, and the firewall context. In the firewall context, repeat the command for the specific rule.

2.6.2 Source Firewall

When WinRM is executed with the quickconfig option, it creates a default firewall rule that allows inbound WinRM traffic. The firewall rule automatically sets the required port (80 or 5985) depending on the WinRM version. Configuring WinRM locally on sources is discouraged as using Group Policy is more manageable.

2.6.2.1 WinRM 2.0

Sources using WinRM 2.0 require that port 5985 is allowed through the firewall. The predefined rule **Windows Remote Management (HTTP-In)** should only be enabled on a computer using WinRM 2.0. The steps for enabling the firewall rule via GPO for the sources can be done by following the Windows Firewall with Advanced Security Group Policy section. This rule should be applied to Windows Vista and beyond as it uses Windows Firewall with Advanced Security. WinRM firewall rule for Windows XP sources are detailed in the Windows XP WinRM 2.0 Clients section.

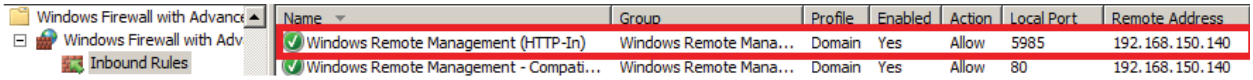


Figure 17: Predefined Rule for WinRM 2.0

Once the WinRM firewall rule is enabled, update the group policy changes using gpupdate. Events should be populating the collector's log. If no events are received, then troubleshooting techniques are provided in the Troubleshooting section.

2.6.2.2 Windows XP WinRM 2.0 Clients

Windows Firewall consists of two profiles: Domain and Standard. [19] The Domain Profile policy does not provide a graphical advantage for configuring firewall rules.

Expand **Computer Configuration > Policies > Administrative Templates > Network > Network Connections > Windows Firewall > Domain Profile.** To enable WinRM firewall rules:

1. Select **Windows Firewall: Define inbound port exceptions** policy
2. Select **Enabled** and click **Show…**

A port exception needs to be manually inserted. The syntax of the port exception is

Port:Transport:Scope:Status:Name

as detailed in the policy. Port specifies the targeted port in which this rule applies. The transport rule specifies **TCP** or **UDP**. The scope details the IP address (or any IP with '*') that can connect to the port. Status indicates if the rule is **enabled** or **disabled**. Name is used to set a name for the rule.

---

[19] http://technet.microsoft.com/en-us/library/bb490626.aspx

3. In the **Show Contents** windows, enter **5985:TCP:*:Enabled:Windows Remote Management**
4. Click **OK** and click **OK**

On the client, this rule will be enabled and allow any connection to port 5985 using TCP.

2.7 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 move laterally between machines and servers by accessing WinRM services. Mitigation to this attack is customizing the predefined rules to only allow connections between collectors and sources. A policy for specifying the IP scope for both source and collector machine is discussed in this section. These configurations apply to the WinRM predefined firewall rules under **Computer Configuration > Policies > Windows Settings > Security Settings > Windows Firewall with Advanced Security > Inbound Rules**.

2.7.1 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**

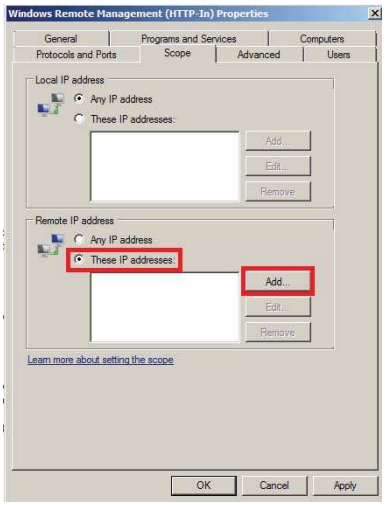


Figure 18: Adding Selective IP addresses

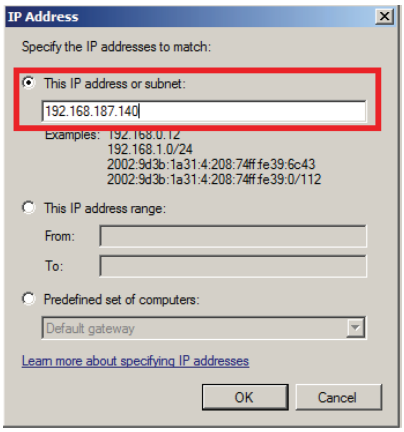


Figure 19: Add IP of Event Collector

Configuring a whitelist, which accepts WinRM traffic only from the collector, is recommended.

2.7.1.1 Windows XP Source Firewall Modifications

The procedures are identical to the Windows XP WinRM 2.0 Clients section with the exception that the scope uses the IP of the collector.

For example, assume the IP of the collector is 192.168.1.2. The port exception rule must be as follows:

5985:TCP:192.168.1.2:Enabled:Windows Remote Management

This rule ensures that connections between Windows XP clients using WinRM are blocked.

2.7.2 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.

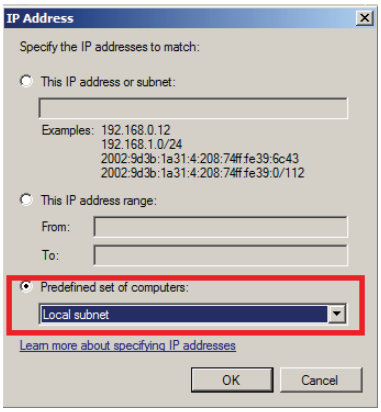


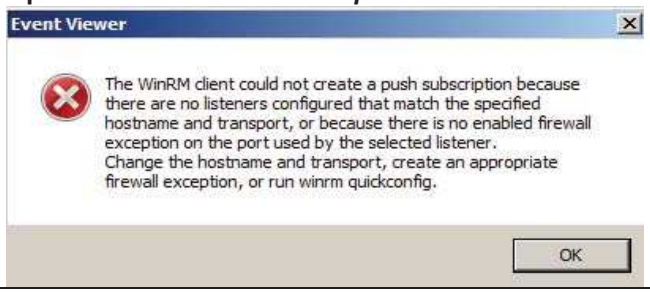
Figure 20: The Event Collector Firewall allowing Local subnet to Connect

Group Policy Firewall Problem

While viewing a subscription in Event Viewer, the following error *may* appear. As the dialog states, a firewall exception needs to be applied. Verify that when you enabled the predefined firewall rules via a Group Policy that the firewall profile for the rule is enabled as well.

A more detailed error message can be obtained by providing the name of the desired subscription (subscriptionID):

wecutil get-subscriptionruntimestatus *SubscriptionID*



2.8 Disabling WinRM and Windows Collector Service

Windows Remote Management (WinRM) and Event Forwarding can be stopped from operating in the network. The collector needs to halt and disable the Windows Event Collector and Windows Remote Management services. These services can be stopped in the Services Microsoft Management Console (MMC) snap-in. The subscriptions created in the Event Viewer should be disabled on the log aggregation server.

To disable collection of events on the server:

1. Open **Services** MMC snap-in
2. Right-click the **Windows Remote Management** service and select **Properties**
3. Change the **Startup type** to **Disabled**
4. In Services status option, select **Stop**
5. Click **OK**
6. Repeat step 1 through 6 for the **Windows Event Collector** service

WinRM can be disabled on each source that was configured by a GPO. The following steps are performed on the Domain Controller for domains using WinRM and Event Forwarding:

1. Open Group Policy Management Editor
2. Navigate to **Computer Configuration > Policies > Windows Settings > Security Settings > System Services**
3. Right-click the **Windows Remote Management** service and select **Properties**
4. Set **Startup** type to **Disabled**
5. Click **OK**
6. Navigate to **Computer Configuration > Policies > Administrative Templates > Event Forwarding**
7. Set the **Configure the server address, refresh interval, and issuer certificate authority of a target Subscription Manager** policy to **Disabled**
8. Click **OK**

Repeat the above steps for additional OUs that use Event Forwarding and WinRM.

2.9 Operating System Based Subscriptions

In an intermixed environment of different Windows operating systems, capturing the same event ID between two different versions of Windows may become problematic. An event ID that appears in Windows XP (e.g., audit events) may have a completely different event ID in Windows Vista and later. This change is due to the eventing system changing in Windows Vista. [20][21] The creation of subscriptions focusing on a single event needs to provide a way to target different Windows versions. Two subscriptions are needed to solve this issue; one for each of the targeted operating system versions.

Separating an entire domain by operating system versions can be daunting so a batch script was developed. See the Operating System Version Separation Script section to complete this task. This script will create two groups, find all computers in the domain based on their operating system version, and add these computers to the newly created group depending on the operating system version. The recommended subscriptions that will capture events based on the Windows operating system version are in the Subscriptions section of the Appendix.

This solution is not required when the network consists of only Windows Vista and later machines.

3 Hardening Event Collection

Windows Remote Management (WinRM) provides security options for authentication and uses other security technologies to encrypt communication channels. This section explains how to securely configure WinRM.

---

[20] http://msdn.microsoft.com/en-us/library/windows/desktop/aa964766(v=vs.85).aspx
[21] http://blogs.msdn.com/b/ericfitz/archive/2009/06/10/mapping-pre-vista-security-event-ids-to-security-events-ids-in-vista.aspx

3.1 WinRM Authentication Hardening Methods

WinRM 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. [22]

The following are acceptable methods used to authenticate with WinRM: [23]

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

The authentication methods for the WinRM client and service can be located by navigating to **Computer Configuration > Policies > Administrative Templates > Windows Components > Windows Remote Management (WinRM).** WinRM Service and WinRM Client authentication methods are respectively shown in Figure 21 and Figure 22.

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.

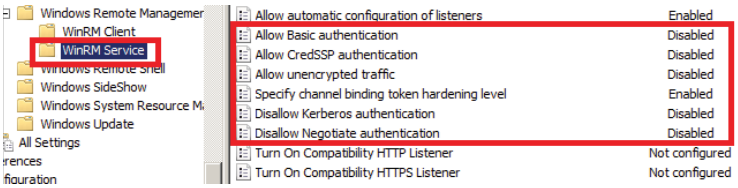


Figure 21: WinRM Service Authentication Policies

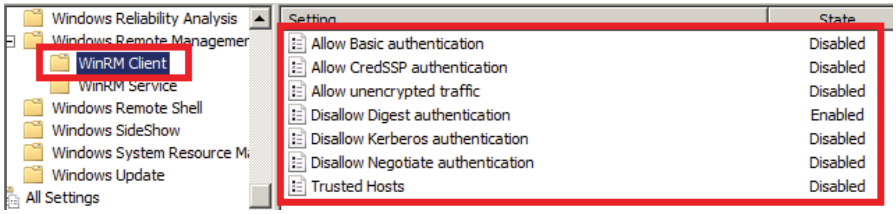


Figure 22: WinRM Client Authentication Policies

The **Allow unencrypted traffic** policy is not part of authentication.

Default value for both Client and Service configuration: **Disabled**

Setting this policy to **Disabled** is recommended.

---

[22] http://technet.microsoft.com/en-us/library/cc775103(v=ws.10).aspx
[23] http://msdn.microsoft.com/en-us/library/windows/desktop/aa384372(v=vs.85).apsx

3.1.1 Basic Authentication

The client can use basic authentication to communicate with a WinRM service. Setting the **Allow Basic authentication** to **Disabled** is recommended.

Default Client Configuration: **True**
Default Service Configuration: **False**

Setting both to **False** is recommended.

3.1.2 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. [23]

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. [24]

The WinRM service does not accept digest authentication as shown in Figure 21. [25][26]

Default Service Configuration: **Not Applicable**
Default Client Configuration: **True**

Setting the client configuration to **False** is recommended.

Setting the **Disallow Digest Authentication** policy to **Enabled** is recommended.

3.1.3 Credential Security Support Provider

Credential Security Support Provider (CredSSP) provides a secure way to delegate a user's credentials from a client to a target server. [23][27][28] The SSP provides the capability of Single Sign-on (SSO) in Terminal Services sessions.[28] This option is only available for WinRM 2.0. Setting the **Allow CredSSP authentication** policy to **Disabled** is recommended.

Default Client Configuration: **False**
Default Service Configuration: **False**

Setting both to **False** is recommended.

---

[24] http://computer-forensics.sans.org/blog/2012/03/09/protecting-privileged-domain-accounts-disabling-encrypted-passwords
[25] http://msdn.microsoft.com/en-us/library/windows/desktop/aa384295(v=vs.85).aspx
[26] http://msdn.microsoft.com/en-us/library/windows/desktop/aa384372(v=vs.85).aspx
[27] ([MS-CSSP]:Credential Security Support Provider (CredSSP) Procotol)
[28] http://technet.microsoft.com/en-us/library/cc749211(WS.10).aspx

3.1.4 Negotiate Authentication

Negotiate authentication is a Security Support Provider (SSP) that provides a client two alternative methods for authentication: Kerberos and NTLM. [29] Negotiate will initially select Kerberos as the default; otherwise, NTLM is used. [23]

Default Client Configuration: **True**
Default Service Configuration: **True**

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. [30] If an error arises stating Negotiate authentication is disabled, a workaround is to use Kerberos locally by specifying the local hostname in the remote switch. [31] Setting the **Disallow Negotiate Authentication** policy to **Enabled** is recommended.

Setting both to **True** is recommended.

3.1.5 Kerberos Authentication

Kerberos version 5 is used as a method of authentication and communication between the service and client. [32][33][34] Setting the **Disallow Kerberos Authentication** policy to **Disabled** is recommended.

Default Client Configuration: **True**
Default Service Configuration: **True**

Setting both to **True** is recommended.

3.1.6 Client Certificate-Based Authentication

Services can verify the connecting client's authenticity by examining its certificate. If the authentication process fails, then the client's connection is revoked.

Default Client Configuration: **True**
Default Service Configuration: **False**

Setting both to **False** is recommended.

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"}[35]

---

[29] http://technet.microsoft.com/en-us/library/cc755084(v=ws.10).aspx
[30] http://msdn.microsoft.com/en-us/library/windows/desktop/aa384295(v=vs.85).aspx
[31] WinRM errorcode 0x803380E1
[32] http://www.ietf.org/rfc/rfc1510.txt
[33] http://technet.microsoft.com/en-us/library/cc772815(v=ws.10).aspx
[34] http://technet.microsoft.com/en-us/library/cc753173(v=ws.10).aspx
[35] If you get an error regarding Negotiate authentication failed after applying hardening authentication methods, see Troubleshooting section in Appendix and the Negotiate Authentication section.

Accessing each source to manually configure this setting is not recommended. This authentication recommendation should be set on the collector.

3.1.7 Channel Binding Token

A common threat amongst NTLML, NTLMv2, and Kerberos authentication methods is a Man-in-the-Middle (MitM) attack. [36] 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. [37][38][39]

CBT ensure 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: [40]

- **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

Default Service Configuration: **Relaxed**

Setting the **Specify channel binding token hardening level** policy to **Strict** is recommended.

This option is not available for WinRM 1.1 and earlier. This is one of the reasons for using WinRM 2.0.

3.1.8 Trusted Host

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. These computers, when using WinRM, will not be authenticated. [26]

Default Client Configuration: **False**

Setting the **Trusted Hosts** policy to **Disabled** is recommended.

---

[36] Securing Windows Networks: Security Advice From The Front Line by Robert Hensing – Microsoft PSS Security; http://it.northwestern.edu/bin/docs/windows_network.ppt
[37] http://msdn/microsoft.com/en-us/library/vstudio/dd767318(v=vs.90).aspx
[38] http://blogs.technet.com/b/srd/archive/2009/12/08/extended-protection-for-authentication.aspx
[39] http://tools.ietf.org/html/rfc5056
[40] **Specify channel binding token hardening level** policy within **Windows Remote Management > WinRM Service** on Windows Server 2008 R2.

3.2 Security Log in Windows XP

By default, Windows XP does not audit security activities. [41] There are several different audit policies which can be enabled to audit security logs.

Expand **Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > Audit Policy.** The DISA STIG states to enable auditing of the following policies: [42]

Policy	Type
Audit account logon events	Success, Failure
Audit account management	Success, Failure
Audit directory service access	Failure
Audit logon events	Success, Failure
Audit object access	Failure
Audit policy change	Success
Audit privilege use	Failure
Audit process tracking	No auditing
Audit system events	Success

Table 3: DISA STIG Auditing Policies Recommendations

Logging of account activities, system events, and process activities can be enabled with this policy. Initially, WinRM runs as Network Service and cannot read the security logs. WinRM needs to run as the local system. [43][44] This alteration modifies the context in which the WinRM service runs as to a higher privilege principal. Services running as a higher privilege principal are generally targeted by attackers. The events stored in the security log outweigh the security implications for allowing collection. This method should only be configured when Windows XP clients are in the log collection network.

3.2.1 Allowing Read Permission

Unless an update is installed, Windows operating systems earlier than Windows 7 are not able to use the Group Policy Preference feature that was introduced in Windows Server 2008. Microsoft has released the KB943729 update to install the Group Policy Preference feature for Windows Vista, Window XP and Windows Server 2003. The previously mentioned feature is automatically installed by Windows Update. [45] Ensure that every Windows Server Update Services (WSUS) servers, if applicable, have these updates available.

When KB943729 is installed using Windows Update, the message “Installing Group Policy Preference Client Side Extensions for Windows XP (KB943729) (update n of N)... done!” will be shown in the Windows Installer text area. All KB packages generate a text file in %WINDIR% named after the installed KB package number (e.g., KB943729.txt).

Expand the EventSources GPO and navigate to **Computer Configuration > Preferences > Control Panel Settings > Services**

---

[41] http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/event_overview_01.mspx
[42] DISA STIG: Windows XP Security Technical Implementation Guide Version 6. Group ID (Vulid): V-6850
[43] http://blogs.msdn.com/b/wmi/archive/2009/04/06/forwarding-security-related-events-from-xp-win2k3-vista-using-winrm-wsman-event-forwarding.aspx
[44] http://blogs.technet.com/b/wincat/archive/2009/06/23/forwarding-security-events-from-windows-xp-server-2003-and-vista-server-2008.aspx
[45] http://support.microsoft.com/kb/943729

To configure a WinRM service to run in the context of **Local System account**:

1. Right-click the **Services** area, select **New**, then **Service**
2. In the **New Service Properties**, set **Startup** to **Automatic (Delayed Start)**
3. Click the **…** button and select **Windows Remote Management (WS-Management)**
4. From the **Service action** dropdown list, select **Start service**
5. Under the **Log on as** area, select **Local System account**
6. Click **OK**

Verify the WinRM Service Log On As Value

A process needs to be executed with the permissions of a logon session. WinRM is run as **NT AUTHORITY\NetworkService**.

A reader can verify a process's logon session in numerous ways. Only two methods will be mentioned. The logon session of a process can be discovered by the succeeding examples:

1. Services: **LogOn** tab of the service's properties
2. Registry keys: **HKLM\SYSTEM\CurrentControlSet\Services\WinRM\ObjectName**

3.3 Secure Socket Layer and WinRM

The WinRM traffic between the collector and source is encrypted using a Windows Integrated Authentication or HTTPS. [25][46][83] 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. [47][48][76] WinRM with SSL requires certificates to authenticate the collector and source. The general steps consist of configuring the listening port, creating certificates for collectors and sources, configuring the subscription manager, creating certificates, and configuring subscriptions. A more detailed explanation of configuring WinRM to use SSL is provided by Microsoft. [16][49]

4 Recommended Events to Collect

This section contains a basic set of events recommended for central collection and review by administrators. The presence of a collected event is not necessarily malicious, and should be reviewed in the appropriate context. Event logs provide a record of activities that can be referenced when malicious activity is discovered on a workstation.

4.1 Application Whitelisting

Application whitelisting events should be collected to look for applications that have been blocked from execution. Any blocked applications could be malware or users trying to run unapproved software. Software Restriction Policies (SRP) is supported on Windows XP and above. The AppLocker feature is available for Windows Vista and above enterprise editions only. Application Whitelisting events can be collected if SRP or AppLocker are actively being used on the network.

---

[46] http://support.microsoft.com/kb/2019527
[47] http://msdn.microsoft.com/en-us/library/cc251580(v=prot.20).aspx
[48] http://technet.microsoft.com/en-us/security/advisory/974926
[49] http://support.microsoft.com/kb/2019527

Windows 7

	ID	Level	Event Log	Event Source
AppLocker Block	8004	Error	Application	Microsoft-Windows-AppLocker
SRP Block	865	Warning	Application	Microsoft-Windows-SoftwareRestrictionPolices

Table 4: Windows 7 Whilelisting Events

Windows XP

	ID	Type	Event Log	Event Source
SRP Block	865	Warning	Application	Software Restriction Policies

Table 5: Windows XP Whitelisting Events

4.2 Application Crashes

Application crashes may warrant investigation to determine if the crash is malicious or benign. Categories of crashes include Blue Screen of Death (BSOD), Windows Error Reporting (WER), Application Crash and Application Hang events. If the organization is actively using the Microsoft Enhanced Mitigation Experience Toolkit (EMET), then EMET logs can also be collected.

Windows 7

	ID	Level	Event Log	Event Source
App Error	1000	Error	Application	Application Error
App Hang	1002	Error	Application	Application Hang
BSOD	1000,1001	Error	System	Microsoft-Windows-WER-SystemErrorReporting
WER	1001	Informational	Application	Windows Error Reporting
EMET	1 2	Warning Error	Application Application	EMET

Table 6: Windows 7 Application Events

Windows XP

	ID	Type	Event Log	Event Source
App Error	1000,1004	Error	Application	Application Error
App Hang	1002	Error	Application	Application Hang
BSOD	1003 1001	Error Informational	System	System Error Save Dump
WER	4097	Informational	Application	DrWatson
EMET	1 2	Warning Error	Application Application	EMET

Table 7: Windows XP Application Events

4.3 System or Service Failures

System and Services failures are interesting events that may need to be investigated. Service operations normally do not fail. If a service fails, then it may be of concern and should be reviewed by an administrator. If a Windows service continues to fail over and over on the same machines, then this may indicate that an attacker is targeting a service.

Windows 7

	ID	Level	Event Log	Event Source
Windows Service Fails or Crashes	7000, 7001, 7022, 7023, 7024, 7026, 7031, 7032, 7034	Error	System	Service Control Manager
Windows Update Failed	25,31	Error	Microsoft-Windows-WindowsUpdateClient/Operational	Microsoft-Windows-WindowsUpdateClient
Hotpatching Failed	1009	Informational	Setup	Microsoft-Windows-Servicing
DCOM invalid permission	10016	Error	System	Microsoft-Windows-DistributedCOM

Table 8: Windows 7 System Events

Windows XP

	ID	Type	Event Log	Event Source
Windows Service Fails or Crashes	7034	Error	System	Service Control Manager
Windows Update Failed	20	Error	System	Windows Update Agent
MSI Installation Failed	11708 11923	Informational Error	Application	MsiInstaller
DCOM invalid permission	10016	Error	System	DCOM

Table 9: Windows XP System Events

4.4 Windows Firewall

If client workstations are taking advantage of the built-in host-based Windows Firewall, then there is value in collecting events to track the firewall status. For example, if the firewall state changes from on to off, then that log should be collected.

Windows 7

	ID	Level	Event Log	Event Source
Firewall Rule Add	2004	Informational	Microsoft-Windows-Windows Firewall With Advanced Security/Firewall	Microsoft-Windows-Windows Firewall With Advanced Security
Firewall Rule Change	2005	Informational	Microsoft-Windows-Windows Firewall With Advanced Security/Firewall	Microsoft-Windows-Windows Firewall With Advanced Security
Firewall Rules Deleted	2006, 2033	Informational	Microsoft-Windows-Windows Firewall With Advanced Security/Firewall	Microsoft-Windows-Windows Firewall With Advanced Security
Firewall Failed to load Group Policy	2009	Error	Microsoft-Windows-Windows Firewall With Advanced Security/Firewall	Microsoft-Windows-Windows Firewall With Advanced Security

Table 10: Windows 7 Firewall Events

Windows XP

	ID	Type	Event Log	Event Source
Windows Firewall port exception list change	852	Success Audit	Security	Security
Windows Firewall application exception list change	851	Success Audit	Security	Security
The Windows Firewall logging settings have changed	854	Success Audit	Security	Security

Table 11: Windows XP Firewall Events

The above events for the listed versions of the Windows operating system are only applicable to modifications of the local firewall settings.

The Windows Firewall raises certain events when modification of its configuration occurs. Deletion of a firewall rule on Windows XP will raise an event based on the rule type. However, no unique firewall rule deletion event is recorded. When an application (851) or port (852) firewall rule changes, the respective event will annotate the cause of the change in the **Change type** field. The **Change type** field will state **Remove** when a rule is deleted.

4.5 Clearing Event Logs

It is unlikely that event log data would be cleared during normal operations and it is likely that a malicious attacker may try to cover their tracks by clearing the event log. When the event log gets cleared, it is suspicious. Centrally collecting events has the added benefit of making it much harder for an attacker to cover their tracks. Using redundant collection servers can also reduce concern of a single point of failure.

Windows 7

	ID	Level	Event Log	Event Source
Event Log was Cleared	104	Informational	System	Microsoft-Windows-Eventlog
Audit Log was Cleared	1102	Informational	Security	Microsoft-Windows-Eventlog

Table 12: Windows 7 Log Activity Events

Windows XP

	ID	Type	Event Log	Event Source
Audit Log was Cleared	517	Success Audit	Security	Security

Table 13: Windows XP Log Activity Events

Unlike Windows 7, Windows XP does not generate an event for clearing logs except for the security log. [50]

4.6 Software and Service Installation

As part of normal network operations, new software and services will be installed, and there is value in logging monitoring this activity. Administrators can review these logs for newly installed software or system services and verify that they do not pose a risk to the network.

Windows 7

	ID	Level	Event Log	Event Source
New Kernel Filter Driver	6	Informational	System	Microsoft-Windows-FilterManager
New Windows Service	7045	Informational	System	Service Control Manager
New MSI File Installed	1022, 1033	Informational	Application	MsiInstaller

Table 14: Windows 7 Software and Service Events

Windows XP

	ID	Type	Event Log	Event Source
New MSI File Installed	1022 11707 11728	Informational	Application	MsiInstaller
New Hofix Installed	4377	Informational	System	NtServicePack
New Security Update Installed	19	Informational	System	Windows Update Agent

Table 15: Windows XP Software and Service Events

4.7 Account Usage

User account information can be collected and audited. Tracking local account usage can help detect Pass the Hash activity and other unauthorized account usage. Additional information such as remote desktop logins, users added to privileged groups, and account lockouts can also be tracked. User accounts being promoted to privileged groups can be audited very closely to ensure that users are in fact supposed to be in a privileged group. Unauthorized membership in privileged groups is a strong indicator that malicious activity has occurred.

[50] http://technet.microsoft.com/en-us/library/cc957086.aspx

Windows 7

	ID	Level	Event Log	Event Source
Account Lockouts	4740	Informational	Security	Microsoft-Windows-Security-Auditing
User Added to Privileged Group	4728, 4732, 4756	Informational	Security	Microsoft-Windows-Security-Auditing
Successful User Account Login	4624	Informational	Security	Microsoft-Windows-Security-Auditing
Failed User Account Login	4625	Informational	Security	Microsoft-Windows-Security-Auditing
Account Login with Explicit Credentials	4648	Informational	Security	Microsoft-Windows-Security-Auditing

Table 16: Windows 7 Account Activity Events

Windows XP

	ID	Type	Event Log	Event Source
Account Lockouts	644	Success Audit	Security	Security
Failed Login	529	Failure Audit	Security	Security
Successful Login	528	Success Audit	Security	Security
User Initiated Logoff	551	Success Audit	Security	Security
Account Login with Explicit Credentials	552	Success Audit	Security	Security
Successful Network Login	540	Success Audit	Security	Security
User Account Created	624	Success Audit	Security	Security
Change Password Attempt	627	Success Audit	Security	Security
User Added to Privileged Group	632, 636, 660	Success Audit	Security	Security

Table 17: Windows XP Account Activity Events

4.8 Kernel Driver Signing

Introduction of kernel driver signing in the 64-bit version of Windows Vista significantly improves defenses against malicious drivers or activities in the kernel. Any indication of a protected driver being altered may indicate malicious activity or a disk error and should warrant investigation.

Windows 7

	ID	Level	Event Log	Event Source
Detected an invalid image hash of a file	5038	Informational	Security	Microsoft-Windows-Security-Auditing
Detected an invalid page hash of an image file	6281	Informational	Security	Microsoft-Windows-Security-Auditing
Code Integrity Check	3001, 3002, 3003, 3004, 3023	Warning, Error	Microsoft-Windows-CodeIntegrity/Operational	Microsoft-Windows-CodeIntegrity
Failed Kernel Driver Loading	219	Warning	System	Microsoft-Windows-Kernel-PnP

Table 18: Windows 7 Kernel Driver Signing Events

4.9 Pass the Hash Detection

Tracking user accounts for detecting Pass the Hash (PtH) requires creating a custom view with XML to configure more advanced filtering options. The event query language is based on XPath. The recommended **QueryList** below is limited in detecting PTH attacks. These queries focus on discovering lateral movement by an attacker using local accounts that are not part of the domain. The **QueryList** captures events that show a local account attempting to connect remotely to another machine not part of the domain. This event is a rarity so any occurrence should be treated as suspicious.

These XPath queries below are used for the Event Viewer's **Custom Views**.

Windows 7

The successful use of PtH for lateral movement between workstations would trigger event ID 4624, with an event level of Information, from the security log. This behavior would be a **LogonType** of 3 using NTLM authentication where it is not a domain logon and not the ANONYMOUS LOGON account. To clearly summarize the event that is being collected:

EventID	Log	Level	LogonType	Authentication Package Name
4624	Security	Information	3	NTLM

In the **QueryList** below, substitute the <DOMAIN NAME> section with the desired domain name.

```
<QueryList>
 <Query Id="0" Path="ForwardedEvents">
  <Select Path="ForwardedEvents">
    *[System[(Level=4 or Level=0) and (EventID=4624)]]
    and
    *[EventData[Data[@Name='LogonType'] and (Data='3')]]
    and
    *[EventData[Data[@Name='AuthenticationPackageName'] = 'NTLM']]
    and
    *[EventData[Data[@Name='TargetUserName'] != 'ANONYMOUS LOGON']]
    and
    *[EventData[Data[@Name='TargetDomainName'] != '<DOMAIN NAME>']]
</Select>
 </Query>
</QueryList>
```

A failed logon attempt when trying to move laterally using PtH would trigger an event ID 4625. This

would have a **LogonType** of 3 using NTLM authentication where it is not a domain logon and not the ANONYMOUS LOGON account. To clearly summarize the event that is being collected:

EventID	Log	Level	LogonType	Authentication Package Name
4625	Security	Information	3	NTLM

```
<QueryList>
 <Query Id="0" Path="ForwardedEvents">
  <Select Path="ForwardedEvents">
   *[System[(Level=4 or Level=0) and (EventID=4625)]]
   and
   *[EventData[Data[@Name='LogonType'] and (Data='3')]]
   and
   *[EventData[Data[@Name='AuthenticationPackageName'] = 'NTLM']]
   and
   *[EventData[Data[@Name='TargetUserName'] != 'ANONYMOUS LOGON']]
   and
   *[EventData[Data[@Name='TargetDomainName'] != '<DOMAIN NAME>']]
</Select>
 </Query>
</QueryList>
```

Windows XP

The **QueryList** for Windows 7 cannot be directly applied for Windows XP without certain significant changes. Successful login and failed login event IDs are different along with the amount of detailed information regarding the login event. In Windows XP a successful network login and failed login will generate events 540 and 529, respectively.

EventID	Log	Level	LogonType	Authentication Package Name
540	Security	Success Audit	3	NTLM
529	Security	Failure Audit	3	NTLM

```
<QueryList>
 <Query Id="0" Path="ForwardedEvents">
  <Select Path="ForwardedEvents">
   *[System[(Level=4 or Level=0) and (EventID=540) and Security[@UserID!='S-1-5-7']]]
   and
   *[EventData[Data[2]!='<DOMAIN NAME>' and Data[4]=3 and Data[6]='NTLM']]
</Select>
 </Query>
</QueryList>
```

```
<QueryList>
 <Query Id="0" Path="ForwardedEvents">
  <Select Path="ForwardedEvents">
   *[System[(Level=4 or Level=0) and EventID=529 and Security[@UserID!='S-1-5-7']]
   and
   *[EventData[Data[2]!='<DOMAIN NAME>' and Data[4]=3 and Data[6]='NTLM']]
</Select>
 </Query>
</QueryList>
```

4.10 Remote Desktop Logon Detection

Remote account activity events are not easily identifiable using the GUI. When an account remotely connects to a client, a generic successful logon event is created. A custom **Query Filter** can aid in clarifying the type of logon that was performed. The query below shows logins using Remote Desktop. Remote Desktop activity should be monitored since only certain administrators should be using it, and they should be from a limited set of management workstations. Any Remote Desktop logins outside of expected activity should be investigated.

The XPath queries below are used for the Event Viewer's **Custom Views**.

EventID	Log	Level	LogonType	Authentication Package Name
4624	Security	Information	10	Negotiate
528	Security	Information	10	Negotiate

Windows 7

```
<QueryList>
 <Query Id="0" Path="ForwardedEvents">
  <Select Path="ForwardedEvents">
    *[System[(Level=4 or Level=0) and (EventID=4624)]]
    and
    *[EventData[Data[@Name='AuthenticationPackageName'] = 'Negotiate']]
   and
    *[EventData[Data[@Name='LogonType'] and (Data='10')]]
</Select>
 </Query>
</QueryList>
```

Windows XP

```
<QueryList>
 <Query Id="0" Path="ForwardedEvents">
  <Select Path="ForwardedEvents">
    *[System[(Level=4 or Level=0) and (EventID=528)]]
    and
    *[EventData[Data[6] = 'Negotiate' and Data[4]=10]]
</Select>
 </Query>
</QueryList>
```

The only difference between the above two queries is the event ID. Windows 7 and Windows XP respectively have event ID 4624 and 528 to indicate a successful logon. A **LogonType** with the value of 10 indicates a Remote Interactive logon. [51]

5 Event Log Retention

It is recommended that the forwarded events log file on the server designated as the central point for log collection is set to a size of approximately 1GB and enable the **Archive the log when full, do not overwrite events** policy to control the behavior when the event log has reach capacity. The theoretical maximum log file size for the forwarded events log on Windows Server 2008 R2 is 2 terabytes[52], 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 (once a day) and setup archiving, or alternatively feed the log data into some larger Security Information Event Management (SIEM) system.

Client workstations and servers should be following the existing DISA STIG for setting the size of other log files (Application, System, Setup, and Security). [53][54][55]

---

[51] http://msdn.microsoft.com/en-us/library/windows/desktop/aa380129(v=vs.85).aspx
[52] http://technet.microsoft.com/en-us/library/hh125924(v=ws.10).aspx
[53] DISA STIG: Windows XP Security Technical Implementation Guide Version 6. Group ID (Vulid): V-1118
[54] DISA STIG: Windows 7 Security Technical Implementation Guide Version 1. Group ID (Vulid): V-26579, V-26580, V-26581, V-26582
[55] DISA STIG: Windows Server 2008 R2 Security Technical Implementation Guide Version 1. Group ID (Vulid): V-26579, V-26580, V-26581, V-26582

The maximum log file sizes are intended for the server whose role is the event collection server of the domain. Clients machine do not need to specify a maximum log size or retention policy. When Event Forwarding is properly configured, all events will be sent to the collector for archiving.

6 Final Recommendations

The central collection of event information helps enterprises gain a better view of activities occurring on the network. Collecting targeted events has the benefit of reducing network and storage requirements while providing useful audit information. Targeted event collection reduces the burden and time required for administrators to review logs which may lead to administrators detecting unapproved or malicious activities.

7 Appendix

7.1 Subscriptions

Event Forwarding on Windows uses subscriptions to specify which events from a set of computers to collect. The Operating System Based Subscriptions section discussed the issue of collecting events from an intermixed network and recommends a possible solution. This section discusses the details of subscriptions and custom subscriptions for Windows 7 and Windows XP computers.

The sample subscription files in this section can be copied as XML files and loaded into the event collector using the command line tool, wecutil.exe. Each of the sample subscriptions do not specify whom is permitted to use the subscriptions (**AllowedSourceDomainComputers** is blank). The creation of the sample subscription can be completed by executing the following commands in order:

1. **wecutil cs <*xml_file_path*>.xml**
    a. An error stating **The subscription fails to activate** will appear so ignore it
2. **wmic path Win32_group where name='EventSources' get sid**
    a. Store this value temporarily

If computers have been separated by their operating system version and placed into groups, execute **wmic path Win32_group where name='new_group_name' get sid** instead of step 2 above. The *new_group_name* needs to be either *vplus* or *previs* depending on targeted group. If a different group was created to represent the different Windows operating system versions, use that name in lieu of *vplus* or *previs*.

3. Obtain the value of the S*ubscriptionId* element from the subscription XML file
4. Using the SID value found in step 2, correct the subscriptions configuration by executing wecutil **ss S*ubscriptionId* /adc:O:NSG:BAD:P(A;;GA;;;sid_value)S:**
5. To verify that no issues are present, execute **wecutil rs S*ubscriptionId***

The parameter /adc of wecutil is used to set a Security Descriptor Definition Language (SDDL) for the targeted subscription. SDDL is discussed in the Security Descriptor Definition Language section.

7.1.1 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 all events in the Application log from a source (client). The targeted sources are the Domain Computers group and the Domain

Controllers group. This subscription example is for testing purposes as it will collect a large amount of events and is not recommended for production use. The example below conforms to the MS-WSMV: Web Services Management Protocol Extensions for Windows Vista, as the subscription was created on Windows Server 2008 R2. [56]

```
<?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"><Select Path="Application">*[System[(Level=0  or Level=
1 or Level=2 or Level=3 or Level=4 or Level=5)]]</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>
```

The following table details each node of the above subscription: [56]

[56] wecutil ss -?

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 [57]
Heatbeat	Used to validate the client's connectivity with subscription [58]
ReadExistingEvents	Notifies the subscription to read all events matching the filter [57]
TransportName	Indicates that either HTTP or HTTPS will be used
ContentFormat	Specifies how the event data will be given to the subscription manager [57]
Locale	Language that the response is translated too [57]
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

Table 19: Subscription XML Description

7.1.2 Sample Windows 7 Recommended Subscription Files

The sample subscriptions below are provided as an example implementation of the recommended events to collect from the Recommended Events to Collect section of this guide. This sample implementation targets event collected from Windows 7 workstations.

The collection of cryptographic events is not permitted unless the operational log of CAPI2 is enabled. A registry value called **Enabled** (REG_DWORD) for each event log indicates if the events are logged. This registry value is found at the following location:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\Channels\Microsoft-Windows-CAPI2/Operational

[57] ([MS-WSMV]: Web Services Management Protocol Extensions for Windows Vista)
[58] (Web Services for Management (WS-Management) Specification)

The **Enabled** registry value should have the value of 1 to enable the log. There is no GPO policy to enable a specific log; however, it is possible to use the Group Policy Preferences to update a registry value via GPO.

WinRM is not permitted to read the CAPI2 event log as only the built-in administrators and the Authenticated Users groups can. Updating the **ChannelAccess** registry value to include the **Event Log Readers** group is recommended. This grants the members of the **Event Log Readers** group permission to read the CAPI2 operational log. Using the same method for enabling the CAPI2 log, append **(A;;CC;;;S-1-5-32-573)** to the existing SDDL. The complete SDDL should be:

O:BAG:SYD:(A;;0x7;;;BA)(A;;0x2;;;AU)(A;;CC;;;S-1-5-32-573)

Logon Using Explicit Credentials (ExpCreds.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>ExpCreds</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description></Description><Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems>
            <MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings>
            <Heartbeat Interval="40000"/>
        </PushSettings>
    </Delivery><Query><![CDATA[
<QueryList><Query Id="0" Path="Security">
    <Select Path="Security">
          *[System[(Level=4 or Level=0) and (EventID=4648)]]
    </Select>
  </Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>Events</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
<AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

Account Locked Out (LockedOut.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>LockedOut</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>User Account Locked Out</Description><Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
        <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems>
            <MaxLatencyTime>1000</MaxLatencyTime>
        </Batching><PushSettings>
            <Heartbeat Interval="40000"/>
        </PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="Security">
    <Select Path="Security">*[System[(Level=4 or Level=0) and (EventID=4740)]]</Select>
  </Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>Events</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
<AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

Account Logons (Logons.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>Logons</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Non-Kerberos Success/Failed Logons</Description><Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime></Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="Security">
    <Select Path="Security">
          *[System[(Level=4 or Level=0) and (EventID=4624 or EventID=4625)]]
          and
         *[EventData[Data[@Name='AuthenticationPackageName'] != 'Kerberos']]
          and
          *[EventData[Data[@Name='TargetUserName'] != 'ANONYMOUS LOGON']]
    </Select></Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName>
    <ContentFormat>Events</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
<AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

Application Crash, Hangs and Windows Error Reporting (AppCrash.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>AppCrash</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>AppCrash and AppHang and WER Logs. Targets: Windows XP+</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
           <MaxItems>1</MaxItems>
            <MaxLatencyTime>1000</MaxLatencyTime></Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery><Query><![CDATA[
<QueryList><Query Id="0" Path="Application">
    <Select Path="Application">*[System[Provider[@Name='Application Error'] and (Level=2)]]</Select>
  </Query><Query Id="1" Path="Application">
    <Select Path="Application">*[System[Provider[@Name='Application Hang'] and (Level=2)]]</Select>
  </Query><Query Id="2" Path="Application">
    <Select Path="Application">*[System[Provider[@Name='Windows Error Reporting'] and (Level=4 or Level=0)]]</Select>
  </Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>RenderedText</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
<AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

Blue Screen of Death (BSODErr.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>BSODErr</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Windows Blue Screen of Death. Targets: Windows XP+</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push">
        <Batching><MaxItems>1</MaxItems>
            <MaxLatencyTime>1000</MaxLatencyTime></Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery><Query><![CDATA[
   <QueryList><Query Id="0" Path="System">
    <Select Path="System">*[System[Provider[@Name='Microsoft-Windows-WER-SystemErrorReporting'] and (Level=2)]]</Select>
<Select Path="System">*[System[Provider[@Name='System Error'] and (EventID=1003)]]</Select>
<Select Path="System">*[System[Provider[@Name='Save Dump'] and (EventID=1001)]]</Select>
</Query>
</QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName>
    <ContentFormat>Events</ContentFormat>
    <Locale Language="en-US"/>
    <LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
  <AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

Windows Kernel Driver Errors (KernelDriverDetect.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>KernelDriverDetect</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Windows Kernel Driver Errors. This will capture Kernel Signing issues on 64-bit versions of Windows Vista+.</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode><Delivery Mode="Push">
        <Batching><MaxItems>1</MaxItems>
        <MaxLatencyTime>1000</MaxLatencyTime></Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings></Delivery><Query>
        <![CDATA[
<QueryList>
  <Query Id="0" Path="Microsoft-Windows-CodeIntegrity/Operational">
    <Select Path="Microsoft-Windows-CodeIntegrity/Operational">*[System[(Level=2 or Level=3) and (EventID=3001 or EventID=3002 or
EventID=3003 or EventID=3004 or EventID=3023)]]</Select></Query>
  <Query Id="1" Path="Security">
    <Select Path="Security">*[System[(Level=0 or Level=4) and (EventID=5038 or EventID=6281)]]</Select></Query>
  <Query Id="2" Path="System">
    <Select Path="System">*[System[Level=3 and EventID=219]]</Select></Query>
</QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>Events</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
    <AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

Windows Defender Errors (DefenderErr.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>DefenderErr</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Windows Defender Errors. Targets: Windows XP+</Description><Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="Microsoft-Windows-Windows Defender/Operational">
    <Select Path="Microsoft-Windows-Windows Defender/Operational">*[System[(Level=2 or Level=3)]]</Select>
  </Query>
  <Query Id="1" Path="System">
    <Select Path="System">*[System[Provider[@Name='WinDefend'] and (Level=2 or Level=3)]]</Select>
  </Query>
</QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>Events</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
    <AllowedSourceDomainComputers></AllowedSourceDomainComputers>
</Subscription>
```

EMET Crash Logs (EMETLogs.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>EMETLogs</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>EMET Crash Logs. Targets: XP+</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery>
    <Query><![CDATA[
<QueryList>
  <Query Id="0" Path="Application">
    <Select Path="Application">*[System[Provider[@Name='EMET'] and (Level=2  or Level=3)]]</Select>
    <Suppress Path="Application">*[System[(EventID=0)]]</Suppress>
  </Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>RenderedText</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
    <AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

MSI Packages Installed (MsiPackages.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>MsiPackages</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>MSI Packages Installed. Targets: Windows XP+</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="Application">
    <Select Path="Application">*[System[Provider[@Name='MsiInstaller'] and (Level=4 or Level=0) and (EventID=1022 or EventID=1033 or
EventID=11707 or EventID=11728)]]</Select>
  </Query>
  <Query Id="1" Path="System">
    <Select Path="System">*[System[Provider[@Name='NtServicePack'] and (EventID=4377)]]</Select>
    <Select Path="System">*[System[Provider[@Name='FilterManager'] and (EventID=6)]]</Select>
    <Select Path="System">*[System[Provider[@Name='Windows Update Agent'] and (EventID=19)]]</Select>
</Query>
</QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>RenderedText</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
    <AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

Windows Service Manager Errors (Service.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>NTService</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Windows Service Manager Errors</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="System">
    <Select Path="System">*[System[(Level=1  or Level=2 or Level=3 or Level=4 or Level=0) and (EventID=7022 or EventID=7023 or EventID=7024
or EventID=7026 or EventID=7031 or EventID=7034 or EventID=7032 or EventID=7045)]]</Select>
  </Query>
</QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>Events</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
    <AllowedSourceDomainComputers></AllowedSourceDomainComputers>
</Subscription>
```

System Log Errors (SysLogs.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>SysLogs</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Windows System Logs</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery>
    <Query><![CDATA[
<QueryList>
  <Query Id="0" Path="System">
    <Select Path="System">*[System[(Level=2 or Level=3 or Level=4 or Level=0) and (EventID=10016 or EventID=11 or EventID=104 or EventID=6 or
EventID=1127 or EventID=1125 or EventID=40964 or EventID=40968)]]</Select>
  </Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>Events</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
<AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

User Account Added to Privileged Group (UserToPriv.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>UserToPriv</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>User Account Added to Privileged Group events. Targets: Windows XP+</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems>
            <MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings>
            <Heartbeat Interval="40000"/>
        </PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="Security">
    <Select Path="Security">*[System[Provider[@Name='Microsoft-Windows-Security-Auditing'] and (EventID=4728 or EventID=4732 or
EventID=4756)]]</Select>
<Select Path="Security">*[System[Provider[@Name='Security'] and (EventID=632 or EventID=636 or EventID=660)]]</Select>
  </Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>Events</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
<AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

Windows File Protection Errors (WFP.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>WFP</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Windows File Protection Errors. Targets: Windows Vista+</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems>
            <MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings>
            <Heartbeat Interval="40000"/>
        </PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="Microsoft-Windows-CorruptedFileRecovery-Client/Operational">
    <Select Path="Microsoft-Windows-CorruptedFileRecovery-Client/Operational">*[System[(Level=2 or Level=4 or Level=0)]]</Select>
    <Select Path="Microsoft-Windows-CorruptedFileRecovery-Server/Operational">*[System[(Level=2 or Level=4 or Level=0)]]</Select>
  </Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>Events</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
<AllowedSourceDomainComputers></AllowedSourceDomainComputers>
</Subscription>
```

Applocker and Software Restriction Policies Blocks (WhitelistingLogs.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>WhitelistingLogs</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>AppLocker and SRP Logs. Targets: Windows XP+</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching><MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="Application">
    <Select Path="Application">*[System[Provider[@Name='Microsoft-Windows-AppLocker'] and (Level=2 or Level=3) and (EventID=8003 or
EventID=8004 or EventID=8006 or EventID=8007)]]</Select>
  </Query>
  <Query Id="1" Path="Application">
    <Select Path="Application">*[System[Level=3 and EventID=865]]</Select>
  </Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>RenderedText</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
<AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

Windows Update Errors (WinUpdateErr.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>WinUpdateErr</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Windows Update Errors</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="Microsoft-Windows-WindowsUpdateClient/Operational">
    <Select Path="Microsoft-Windows-WindowsUpdateClient/Operational">*[System[(Level=2 or Level=3)]]</Select>
    <Select Path="Setup ">*[System[Provider[@Name='Microsoft-Windows-Servicing'] and (EventID=1009)]]</Select>
    <Suppress Path="Application">*[System[(EventID=29)]]</Suppress>
  </Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>Events</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
    <AllowedSourceDomainComputers></AllowedSourceDomainComputers>
</Subscription>
```

Windows Firewall with Advanced Security (WinFAS.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>WinFAS</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Windows Firewall and Windows Firewall with Advanced Security Logs</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="Microsoft-Windows-Windows Firewall With Advanced Security/Firewall">
    <Select Path="Microsoft-Windows-Windows Firewall With Advanced Security/Firewall">*[System[(Level=2 or Level=4 or Level=0) and
(EventID=2009 or EventID=2033 or EventID=2004 or EventID=2005)]]</Select>
  </Query>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>Events</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
    <AllowedSourceDomainComputers></AllowedSourceDomainComputers>
</Subscription>
```

Log Deletion (LogDel.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>LogDel</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Windows Firewall Logs</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0">
    <Select Path="Security">*[System[Provider[@Name='Microsoft-Windows-Eventlog'] and (Level=4 or Level=0) and EventID=1102]]</Select>
    <Select Path="System">*[System[Provider[@Name='Microsoft-Windows-Eventlog'] and (Level=4 or Level=0) and EventID=104]]</Select>
</Query>
<Query Id="1" Path="Security">
<Select Path="Security">*[System[Provider[@Name='Security'] and (Level=4 or Level=0) and EventID=517 and Task=1]]</Select></Query>
</QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>RenderedText</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
    <AllowedSourceDomainComputers></AllowedSourceDomainComputers>
</Subscription>
```

Cryptographic Log (CryptoLogs.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>CryptoLogs</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Crypto Informational Logs</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="Microsoft-Windows-CAPI2/Operational">
    <Select Path="Microsoft-Windows-CAPI2/Operational">*[System[(Level=4 or Level=0) and EventID=90]]</Select></Query>
</QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>RenderedText</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
    <AllowedSourceDomainComputers></AllowedSourceDomainComputers>
</Subscription>
```

7.1.3 Sample Windows XP Recommended Subscriptions Files

A subset of the subscriptions for Windows 7 sources is applicable to Windows XP sources. The subscriptions provided in this section are tailored for Windows XP sources.

The following list of Windows 7 subscriptions can also target Windows XP sources equally:

- Application Crash, Hangs and Windows Error Reporting (AppCrash.xml)
- EMET Crash Logs (EMETLogs.xml)
- Applocker and Software Restriction Policies Blocks (WhitelistingLogs.xml)
- Log Deletion (LogDel.xml)
- MSI Packages Installed (MsiPackages.xml)
- Blue Screen of Death (BSODErr.xml)
- User Account Added to Privileged Group (UserToPriv.xml)

When the collector receives the events from Windows XP sources, the event viewer may not render a well-formatted message for the event. Subscriptions targeting Windows XP sources will have the **ContentFormat** set to **RenderedText**. If this is not the case, then the **ContentFormat** can be set to **RenderedText** by executing the following command:

wecutil ss *SubscriptionId* /cf:RenderedText /e:true

Logon Using Explicit Credentials (XP_ExpCreds.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>XP_ExpCreds</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description></Description><Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems>
            <MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings>
            <Heartbeat Interval="40000"/>
        </PushSettings>
    </Delivery><Query><![CDATA[
<QueryList><Query Id="0" Path="Security">
    <Select Path="Security">
          *[System[(Level=4 or Level=0) and EventID=552]]
    </Select>
  </Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>RenderedText</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
<AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

Account Locked Out (XP_LockedOut.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>XP_LockedOut</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>User Account Locked Out</Description><Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
        <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems>
            <MaxLatencyTime>1000</MaxLatencyTime>
        </Batching><PushSettings>
            <Heartbeat Interval="40000"/>
        </PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="Security">
    <Select Path="Security">*[System[(Level=4 or Level=0) and EventID=644]]</Select>
  </Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>RenderedText</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
<AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

Account Logons (XP_Logons.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>XP_Logons</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Non-Kerberos Success/Failed Logons</Description><Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime></Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="Security">
    <Select Path="Security">
          *[System[(Level=4 or Level=0) and (EventID=529 or EventID=528 or EventID=540)]]
    </Select></Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>RenderedText</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
<AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

System Log Errors (XP_SysLogErr.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>XP_SysLogsErr</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Windows System Logs</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery>
    <Query><![CDATA[
<QueryList><Query Id="0">
<Select Path="System">*[System[Provider[@Name='Service Control Manager'] and (Level=2 or Level=4 or Level=0) and (EventID=20 or EventID=7034
or EventID=10016 or EventID=11708 or EventID=11923]]</Select>
<Select Path="System">*[System[Provider[@Name='DCOM'] and (EventID=10016)]]</Select>
  </Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>RenderedText</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
<AllowedSourceDomainComputers> </AllowedSourceDomainComputers>
</Subscription>
```

Windows Update Errors (XP_WinUpdateErr.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>XP_WinUpdateErr</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Windows Update Errors</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="System">
    <Select Path="System">*[System[Provider[@Name='Windows Update Agent'] and (EventID=20)]]</Select>
    <Suppress Path="Application">*[System[(EventID=29)]]</Suppress>
  </Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>RenderedText</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
    <AllowedSourceDomainComputers></AllowedSourceDomainComputers>
</Subscription>
```

Windows Firewall (XP_WinF.xml)

```
<Subscription xmlns="http://schemas.microsoft.com/2006/03/windows/events/subscription">
    <SubscriptionId>XP_WinF</SubscriptionId>
    <SubscriptionType>SourceInitiated</SubscriptionType>
    <Description>Windows Firewall Logs</Description>
    <Enabled>true</Enabled>
    <Uri>http://schemas.microsoft.com/wbem/wsman/1/windows/EventLog</Uri>
    <ConfigurationMode>Custom</ConfigurationMode>
    <Delivery Mode="Push"><Batching>
            <MaxItems>1</MaxItems><MaxLatencyTime>1000</MaxLatencyTime>
        </Batching>
        <PushSettings><Heartbeat Interval="40000"/></PushSettings>
    </Delivery><Query><![CDATA[
<QueryList>
  <Query Id="0" Path="Security">
    <Select Path="Security">*[System[(Level=4 or Level=0) and (EventID=851 or EventID=852 or EventID=854)]]</Select>
  </Query></QueryList>
        ]]></Query>
    <ReadExistingEvents>true</ReadExistingEvents>
    <TransportName>http</TransportName><ContentFormat>RenderedText</ContentFormat>
    <Locale Language="en-US"/><LogFile>ForwardedEvents</LogFile>
    <AllowedSourceNonDomainComputers></AllowedSourceNonDomainComputers>
    <AllowedSourceDomainComputers></AllowedSourceDomainComputers>
</Subscription>
```

7.1.4 Operating System Version Separation Script

The script below provides a solution to the issue of grouping an intermixed network as mentioned in the Operating System Based Subscriptions section. This script needs to be run on the domain controller with

domain administrator privileges. It is assumed the reader has followed this guidance document in its entirety.

```
@echo off
setlocal

REM This batch script will search for all computer in the current domain and separate them by OS version.
REM Needs to be run by a domain administrator
REM
REM Limitations:
REM This script assumes you have followed the guide. It assumes there is an EventSources group in the domain.
REM The scope of this script is not intended for multiple domains or multiple forests.
REM The scope is for the current (single) domain.
REM This will not add members of the EventCollectors groups to the new groups created.
REM Assumes there are no Windows Server 2003 R2 Collectors. Issues will arise if there are.
REM
REM Note:
REM This will add the collector (assuming collector is in same domain) to a group. This does not affect the ability to
REM read or receive logs on the collector

REM Steps:
REM  Search for all Domain controllers in forest
REM  1. Create two groups one for vista and beyond (vplus) and another for Windows XP (previs)
REM  1a. Get domain names (DN)
REM  2. Search for all Vista+ computers and add them to the vplus group
REM  3. Repeat step 2 but for Windows XP

REM First Remove members of EventSources
for /f "usebackq tokens=*" %%c in (`dsquery * domainroot -filter "(&(objectclass=group)(name=EventSources))"`) do (
            for /f "usebackq tokens=*" %%d in (`dsquery * domainroot -filter "(memberOf=%%c)"`) do ( dsmod group %%c -rmmbr %%d)
)

REM Get base of current domain

for /f "usebackq tokens=*" %%c in (`dsquery * domainroot -scope base`) do (set dc=%%c)

if not defined dc (

            echo Error getting domain
            exit /b 1
)

REM  Add Groups to User Container

set container="CN=Users,"
set nvgroup="CN=vplus,"
set nxgroup="CN=previs,"
set vplus=%nvgroup%%container%%dc%
set previs=%nxgroup%%container%%dc%


dsadd group %vplus% -secgrp yes -scope g
dsadd group %previs% -secgrp yes -scope g

REM Search for all computers in the domain that are not Domain Controllers and are version 6.0 and above
REM  Vista and above; Get EventCollector group
set collector=""

for /f "usebackq tokens=*" %%d in (`dsquery * domainroot -filter "(&(objectclass=group)(!(primaryGroupID=516))(name=EventCollectors))"`) do (set collector=%%d)


for /f "usebackq tokens=*" %%c in (`dsquery.exe * domainroot -filter
"(&(objectCategory=computer)(!(primaryGroupID=516))(objectClass=Computer)(operatingSystemVersion>=6.0)(!(memberof="%collector%")))"`) do (
            REM Verify if this computer is part of the EventCollectors group, so do not add to new group
            dsmod group %vplus% -addmbr %%c
)


REM Search for all computers in the domain that are not Domain Controllers or not members of EventCollectors group and are version 6.0 and above
REM  XP and below (NT5.1 and NT5.2)
for /f "usebackq tokens=*" %%c in (`dsquery.exe * domainroot -filter
"(&(objectCategory=computer)(!(primaryGroupID=516))(objectClass=Computer)(operatingSystemVersion>=5.1)(operatingSystemVersion<=5.9))"`) do (
            dsmod group %previs% -addmbr %%c
)

REM Now add these newly created groups to be a member of the EventSources group

REM Get DN of EventSources
for /f "usebackq tokens=*" %%c in (`dsquery group domainroot -name "EventSources"`) do (set esgrp=%%c)

REM Add new groups to Eventsources
dsmod group %esgrp% -addmbr %vplus%,%previs%

endlocal
```

7.2 Event ID Definitions

This guidance document has given a list of event IDs to be aware of when monitoring activity. This list is not complete nor should it be the only set of events to be collected. Each environment will most likely focus on specific events or currently using a third party application for event monitoring.

Microsoft's Events and Errors Message Center web site provides a central location for identifying event IDs for each Windows operating system. [59] Effective use of this resource requires an event ID, or some other information about the event, is known beforehand.

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 [60]

- 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

- Windows Server 2008: Events and Errors http://technet.microsoft.com/en-us/library/cc754424(v=ws.10).aspx

Windows Server 2008 Component-Based Servicing events

- Update and package related 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

Windows Vista, Windows Server 2008, Windows 7 and Windows Server 2008 R2 security audit events are provided by Microsoft either by a support article or a downloadable Excel file. [61][62][63] The Windows operating system, beginning with Windows Vista, provides a command line tool, wevtutil, to list all event IDs raised by a publisher along with the event's message. [64]

The Windows Events Command Line Utility can obtain information regarding event logs and publishers. [65]

wevtutil gp Microsoft-Windows-Security-Auditing /ge:true /gm:true[66]

---

[59] http://www.microsoft.com/technet/support/ee/ee_advanced.aspx
[60] http://blogs.msdn.com/b/ericfitz/archive/2007/10/12/list-of-windows-server-2003-events.aspx
[61] http://www.microsoft.com/en-us/download/details.aspx?id=17871
[62] http://support.microsoft.com/kb/947226
[63] http://www.micrsoft.com/en-us/download/details.aspx?id=21561
[64] wevtutil /?
[65] http://technet.microsoft.com/en-us/library/cc732848(WS.10).aspx
[66] http://blogs.microsoft.com/b/ericfitz/archive/2007/07/31/documentation-on-the-windows-vista-and-windows-server-2008-security-events.aspx

The above command will get the publisher (gp/get-publisher), obtain information on events that the publisher uses, and produce readable messages for each event. This command can be applied to any publisher to obtain a list of all their events.

The mapping of security event IDs between Windows XP and the latest versions of Windows can be revealed by a simple addition or subtraction of 4096_{10}/$0x1000_{16}$. [67] This rule is not applicable to events dealing with successful and failed logons. [67]

7.3 Windows Remote Management Versions

There have been four versions of WinRM since its introduction in Windows Server 2003 R2. The following table correlates WinRM version to Windows operating system version. [68]

Version	Support
WinRM 0.5	Windows Server 2003 R2*
WinRM 1.0	Windows Vista
WinRM 1.1	Windows Vista SP1 Windows Server 2008 Windows Server 2003 SP1** Windows Server 2003 SP2** Windows Server 2003 R2** Windows XP SP2**
WinRM 2.0	Windows 7 Windows Server 2008 R2 Windows Server 2008 SP1*** Windows Server 2008 SP2*** Windows Vista SP1*** Windows Vista SP2*** Windows XP SP3***
WinRM 3.0	Windows 8 Windows 7 SP1**** Windows Server 2008 SP1**** Windows Server 2008 SP2****

Table 20: WinRM Version Correlation

* = Installed from the Add/Remove System Components feature within the Hardware Management feature
** = Install WS-Management v1.1. [69]
*** = Installed as part of the Windows Management Framework Core package. [70][71] This update requires at least Microsoft .NET Framework 2.0 Service Pack 1. [72]
**** = Installed as part of Windows Management Framework 3.0. This update requires at least Microsoft .NET Framework 4.0. [73]

Installation packages for WinRM can be found in knowledge base articles, shown below.

---

[67] http://blogs.msdn.com/b/ericfitz/archive/2009/06/10/mapping-pre-vista-security-events-ids-to-security-events-ids-in-vista.aspx
[68] http://technet.microsoft.com/en-us/library/ff520073(v=ws.10).aspx
[69] https://www.microsoft.com/en-us/download/details.aspx?id=21900
[70] https://www.microsoft.com/en-us/download/details.aspx?id=9864
[71] https://www.microsoft.com/en-us/download/details.aspx?id=16818
[72] https://www.microsoft.com/en-us/download/details.aspx?id=16614
[73] https://www.microsoft.com/en-us/download/details.aspx?id=34595

WinRM Version (KB#)	Supported OS	KB URIs
WinRM 1.1 (KB936059)	Windows Server 2003 SP1 Windows Server 2003 SP2 Windows XP SP2 Windows XP SP3*	http://support.microsoft.com/kb/936059 +
WinRM 2.0 (KB968930)	Windows Server 2003 SP2 Windows Server 2008 Windows Server 2008 SP2 Windows Vista SP1 Windows Vista SP2 Windows XP SP2* Windows XP SP3	http://support.microsoft.com/kb/968930 + * Requires Microsoft Windows Installer 3.1 * Requires .NET Framework 2.0 SP1
WinRM 3.0 (KB2506146)	Windows 7 SP1 Windows Server R1 SP1 Windows Server 2008 SP2	http://support.microsoft.com/kb/2506146 + * Requires .NET Framework 4.0 * Update comes with Release Notes

Table 21: WinRM Version Update URLs

Microsoft published a knowledge base article (KB936059)[74] and an update for WinRM 1.1.[75] The knowledge base article offers additional post-installation information to the update that is not mentioned in this document. The actual update can be applied to Windows XP SP2, Windows Server 2003 SP1, Windows Server 2003 SP2, and Windows 2003 Server R2.

7.4 WinRM 2.0 Configuration Settings

The quick configuration option of WinRM uses the following default configuration settings on Windows Server 2008 R2. [26][76] Default values of WinRM configuration settings are shown and referenced in this document for convenience. [26] The following WinRM command displays the configuration setting of WinRM

winrm get winrm/config

It produces the following example output:

[74] http://support.microsoft.com/kb/936059
[75] https://www.microsoft.com/en-us/download/details.aspx?id=21900
[76] ([MS-WSMV]: Web Services Management Protocol Extensions for Windows Vista, 2012)

```
Config
    MaxEnvelopeSizekb = 150
    MaxTimeoutms = 60000
    MaxBatchItems = 32000
    MaxProviderRequests = 4294967295
    Client
        NetworkDelayms = 5000
        URLPrefix = wsman
        AllowUnencrypted = false
        Auth
            Basic = true
            Digest = true
            Kerberos = true
            Negotiate = true
            Certificate = true
            CredSSP = false
        DefaultPorts
            HTTP = 5985
            HTTPS = 5986
        TrustedHosts
    Service
        RootSDDL = O:NSG:BAD:P(A;;GA;;;BA)S:P(AU;FA;GA;;;WD)(AU;SA;GWGX;;;WD)
        MaxConcurrentOperations = 4294967295
        MaxConcurrentOperationsPerUser = 15
        EnumerationTimeoutms = 60000
        MaxConnections = 25
        MaxPacketRetrievalTimeSeconds = 120
        AllowUnencrypted = false
        Auth
            Basic = false
            Kerberos = true
            Negotiate = true
            Certificate = false
            CredSSP = false
            CbtHardeningLevel = Relaxed
        DefaultPorts
            HTTP = 5985
            HTTPS = 5986
        IPv4Filter = *
        IPv6Filter = *
        EnableCompatibilityHttpListener = false
        EnableCompatibilityHttpsListener = false
        CertificateThumbprint
    Winrs
        AllowRemoteShellAccess = true
        IdleTimeout = 180000
        MaxConcurrentUsers = 5
        MaxShellRunTime = 2147483647
        MaxProcessesPerShell = 15
        MaxMemoryPerShellMB = 150
        MaxShellsPerUser = 5
```

Each of field of the above output is described in the following sections.

7.4.1 Protocol Settings

These settings are configurable options for the WS-Management protocol used by WinRM.

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**

Table 22: Protocol Settings

7.4.2 Client Configuration

The following parameters configures on 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.

Table 23: WinRM Client Configuration

7.4.3 WinRM Service

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**
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**

Table 24: WinRM Service

7.4.4 WinRS

Windows Remote Shell (WinRS) is turned on by default. The recommendation is to disable it. Each of the parameters for WinRS will use the default value if no policy is configured. [77][26]

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.

Table 25: WinRS Configuration Settings

7.5 WinRM Registry Keys and Values

Throughout this document, registry keys can be used for verification purposes only. Do not to modify any registry keys as this may cause unforeseen problems and possible system corruption. The registry keys and values below apply to both Windows XP and Windows 7 when configured using GPO. These keys are found by viewing the following GPO Administrative Template (ADM) files located at %WINDIR%\system32\GroupPolicy\Adm on Windows XP:

- 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.

[77] http://msdn.microsoft.com/en-us/library/windows/desktop/ee309367(v=vs.85).aspx

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

Table 26: WinRM, WinRS, WSMAN and Event Forwarding Registry Values

7.6 Installation Batch Script

Microsoft does not provide an MSI to install Windows Remote Management, therefore it is require to download the update. A batch script was developed to simplify the task of installing the WinRM update. The updates that will be installed must reside in a shared folder that all targeted clients can access.

```
install.bat Z:\PATH_TO_UPDATES\
```

The script does not accept a hostname, only a directory path. After the script completes, a reboot of the client is needed. The batch script should be executed as a startup script. This script should be saved as install.bat for the purpose of this guide. This script should be run once and when a new Windows XP workstation is added to the network.

```
@ECHO OFF
SETLOCAL
SETLOCAL ENABLEDELAYEDEXPANSION
::This script will only target KB968930 (WINRM2.0) and .NET 2.0 for x86. All other versions should be added
::by manually modifying this script.
::NOTE: Only have the needed executables in the directory since these will be copied to the client
::locally

IF "%1"==="" (
echo Error Parameter missing
echo install.bat NETWORKPATH_TO_EXECUTABLE
GOTO END
)
::This is where the WinRM executable resides
:: Need to be able to distinguish between Names and letter drives
:: SET NetworkUpdatesPath=\\%1
SET NetworkUpdatesPath=%1
::Setup the installation location
SET LocalUpdatesPath=%PROGRAMFILES%\Common Files\Updates
::Setup the Log file of this batch file
::SET UpdateLog="\\%1\UpdateLog-%COMPUTERNAME%.txt"
SET UpdateLog="%1\UpdateLog-%COMPUTERNAME%.txt"
ECHO %DATE% %TIME% %COMPUTERNAME% Update processing started >>%UpdateLog%

IF NOT EXIST "%NetworkUpdatesPath%" (
  ECHO %DATE% %TIME% %COMPUTERNAME% Network updates directory does not exist >>%UpdateLog%
  GOTO END
)
IF NOT EXIST "%LocalUpdatesPath%" (
  ECHO Creating local updates directory: "%LocalUpdatesPath%"
  ECHO %DATE% %TIME% %COMPUTERNAME% Creating local updates directory >>%UpdateLog%
  mkdir "%LocalUpdatesPath%" >>%UpdateLog% 2>&1
  ECHO. >>%UpdateLog%
  IF NOT !ERRORLEVEL!==0 (
                    ECHO Error creating directory
                    ECHO %DATE% %TIME% %COMPUTERNAME% Error creating local updates directory >>%UpdateLog%
                    GOTO END
  ) ELSE (
                    ECHO Successfully created local updates directory
                    ECHO %DATE% %TIME% %COMPUTERNAME% Successfully created local updates directory >>%UpdateLog%
  )
)
FOR %%I IN ("%NetworkUpdatesPath%\*") DO (
                    SET File=%%~nxI
                    SET Process=0
                    IF %%~xI==.exe SET Process=1
                    IF !Process!==1 (
                                        ECHO Processing: "!File!"
                                        ECHO !DATE! !TIME! %COMPUTERNAME% Processing: "!File!" >>%UpdateLog%
                                        SET FILE1=
                                        SET FILE2=
                                        FOR /F "usebackq delims==" %%J IN (`dir /TW "%NetworkUpdatesPath%\!File!" ^| find "!File!"`) DO SET FILE1=%%J
                                        IF EXIST "%LocalUpdatesPath%\!File!" (FOR /F "usebackq delims==" %%J IN (`dir /TW "%LocalUpdatesPath%\!File!" ^| find "!File!"`) DO SET FILE2=%%J)
                                        REM FC /B "%%~fI" "%LocalUpdatesPath%\!File!" >>%UpdateLog% 2>&1
                                        REM IF NOT !ERRORLEVEL!==0 (
                                        IF NOT "!FILE1!"=="!FILE2!" (
                                                            ECHO   Found updated file "!File!"
                                                            ECHO !DATE! !TIME! %COMPUTERNAME%   Found updated file "!File!" >>%UpdateLog%
                                                            ECHO   Copying "!File!" to local updates directory
                                                            ECHO !DATE! !TIME! %COMPUTERNAME%   Copying "!File!" to local updates directory >>%UpdateLog%
                                                            COPY "%%~fI" "%LocalUpdatesPath%\!File!" /Y >>%UpdateLog% 2>&1
                                                            REM ECHO. >>%UpdateLog%
                                                            IF !ERRORLEVEL!==0 (
                                                                                IF %%~xI==.exe (
                                                                                                    SET FILENAME=%%~nxI
                                                                                                    IF /I "!FILENAME!" EQU "NetFx20SP1_x86.exe" (
                                                                                                                        ECHO Running: "%LocalUpdatesPath%\!File!" /quiet /norestart
                                                                                                                        ECHO !DATE! !TIME! %COMPUTERNAME%   Running: "%LocalUpdatesPath%\!File!" /quiet /norestart >>%UpdateLog%
                                                                                                                        "%LocalUpdatesPath%\!File!" /quiet /norestart >>%UpdateLog% 2>&1
                                                                                                                        IF ERRORLEVEL 1 (
                                                                                                                                            ECHO Failed to install  !FILENAME!
                                                                                                                                            GOTO END
                                                                                                                        )

                                                                                                                        ECHO. >>%UpdateLog%

                                                                                                    )ELSE (
                                                                                                                        IF /I "!FILENAME!" EQU "WindowsXP-KB968930-x86-ENG.exe" (
                                                                                                                                            ECHO Running: "%LocalUpdatesPath%\!File!" /quiet /norestart
                                                                                                                                            ECHO !DATE! !TIME! %COMPUTERNAME%   Running: "%LocalUpdatesPath%\!File!" /quiet /norestart >>%UpdateLog%
                                                                                                                                            "%LocalUpdatesPath%\!File!" /quiet /norestart >>%UpdateLog% 2>&1

                                                                                                                                            IF ERRORLEVEL 1 (
                                                                                                                                                                ECHO Failed to install  !FILENAME!
                                                                                                                                                                GOTO END
                                                                                                                                            )

                                                                                                                                            ECHO. >>%UpdateLog%
                                                                                                                        )
                                                                                                    )
                                                                                ) ELSE (
                                                                                                    ECHO   Error copying file
                                                                                                    ECHO !DATE! !TIME! %COMPUTERNAME%   Error copying file >>%UpdateLog%
                                                                                )
                                                            ) ELSE (
                                                                                ECHO   !File! not updated
                                                                                 ECHO !DATE! !TIME! %COMPUTERNAME%   "!File!" not updated >>%UpdateLog%
                                                            )
                                        )
                                        )
                    )
)
IF EXIST "%NetworkUpdatesPath%\Epo" (
                    IF NOT EXIST "%LocalUpdatesPath%\Epo" (
                                        ECHO %DATE% %TIME% %COMPUTERNAME% Creating local Epo directory >>%UpdateLog%
                                        mkdir "%LocalUpdatesPath%\Epo" >>%UpdateLog% 2>&1
                                        ECHO. >>%UpdateLog%
                                        ECHO %DATE% %TIME% %COMPUTERNAME% Copying Epo directory >>%UpdateLog%
                                        copy "%NetworkUpdatesPath%\Epo\*.*" "%LocalUpdatesPath%\Epo\*.*" /Y >>%UpdateLog% 2>&1
                                        ECHO. >>%UpdateLog%
                                        ECHO %DATE% %TIME% %COMPUTERNAME% Running Epo installer >>%UpdateLog%
                                        CD "%LocalUpdatesPath%\Epo" >>%UpdateLog% 2>&1
                                        FramePkg.exe /Install=Agent /ForceInstall /Silent >>%UpdateLog% 2>&1
                                        ECHO. >>%UpdateLog%
                                        ECHO %DATE% %TIME% %COMPUTERNAME% Finished installing Epo >>%UpdateLog%
                    )
)
:END
ECHO Done
ECHO %DATE% %TIME% %COMPUTERNAME% Done >>%UpdateLog%
exit /b %errorlevel%
```

7.7 WinRM LogOnAs Correction Batch Script

This script is used to correct the issue with WinRM on Windows Vista and beyond where the **LogOnAs** value is changed to local system after applying the steps in the Security Log in Windows XP section. This script should be run as a startup script. The script should be saved as LogOnAs_detect.bat for this guide.

```
:: WinRM does not need to be Localsystem on Vista+ in an intermixed environment of Windows XP and Windows Vista+.
:: If it is, WinRM will not start. Therefore, WinRM LogOnAs must be set to "NT AUTHORITY\NetworkService".

::This startup script will do the following
:: 1. Detect OS version
:: 2. Determine if LogOnAs modification for WinRM is needed
:: 2a. If Windows XP, exit as the LogOnAs needs to be Localsystem to read security logs
:: 2b. If LogOnAs is correct for that current version of Windows, exit.
:: 3. If so, update WinRM registry key
:: 4. Start WinRM

@echo off
setlocal
set minVer=6.0

::Detect OS Version/Extract OS Version number
::http://msdn.microsoft.com/en-us/library/ms724834(VS.85).aspx
for /f "usebackq tokens=*" %%c in (`reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion" /v CurrentVersion`) do (
          set regtmp=%%c
)

for /f "tokens=3" %%c in ("%regtmp%") do (
          set tversion=%%c
)

::If version < 6.0 (e.g., Windows XP), exit
if %tversion% lss %minVer% (
          echo No Change Needed.
          goto end
)

::Check if LogOnAs is set to LocalSystem indicating it needs change
for /f "usebackq tokens=*" %%c in (`reg query HKLM\SYSTEM\CurrentControlSet\services\WinRM /v ObjectName`) do (
          set regtmp=%%c
)

::This would get "NT" if NT AUTHORITY\NetworkService is set
for /f "tokens=3" %%c in ("%regtmp%") do (
          set curLogOnAs=%%c
)

if /I "%curLogOnAs%" neq "LocalSystem" (
          echo No Change Needed.
          goto end
)

echo Change Needed

::Do modifications on Vista+ machines

:: Stop any currently running instances of WinRM
sc stop winrm

::Timeout is not available in Windows XP
::Wait 10 seconds to stop WinRM
timeout /t 10 /nobreak


reg add HKLM\SYSTEM\CurrentControlSet\services\WinRM /v ObjectName /t REG_SZ /d "NT AUTHORITY\NetworkService" /f


sc start WinRM


:END

endlocal
```

7.7.1 Security Descriptor Definition Language

The language in the **AllowedSourceDomainComputers** node is called Security Descriptor Definition Language (SDDL). [78] A subscription can be customized to target single or multiple users, computers, or groups. If Windows clients are grouped based on OS version, subscriptions can also target a specific operating system version (e.g., Windows XP or Windows 7). Creating additional groups specifying each version of the Windows operating system in the domain is recommended. These groups' SID value can be used to configure the targeted subscription's SDDL.

Microsoft provided the SDDL structure as shown: [79]

---

[78] http://msdn.micosoft.com/en-us/library/windows/desktop/aa379567(v=vs.85).aspx
[79] http://msdn.microsoft.com/en-us/library/aa379570(v=vs.85).aspx

O: Owner_SID
G: Group_SID
D: DACL_FLAGS(string_ace1)(string_ace2).... (string_aceN)
S: SACL_FLAGS(string_ace1)(string_ace2).... (string_aceN)

string_ace are optional Access Control Entries.

ACE has the following structure: [80]
(AceType;AceFlags;Rights;ObjectGuid;InheritObjectGuid;AccountSID;resource_attribute)

There is also an option to use conditional ACE; however, that will not be discussed here. [81]

In the sample subscription configuration file the SDDL use was:
O:NSG:NSD:(A;;GA;;;DC)(A;;GA;;;NS)

The breakdown of **O:NSG:NSD:** is shown below:

SID and Flags	Description
O:	Network Service
G:	Network Service
D:	None

String_ACE breakdown of (A;;GA;;;DC) (A;;GA;;;NS)

String_ACE1	String_ACE2
(A;;GA;;;DC)	(A;;GA;;;NS)
AceType = "A" = ACCESS_ALLOWED_ACE_TYPE	AceType = "A" = ACCESS_ALLOWED_ACE_TYPE
AceFlags = None	AceFlags = None
Rights = "GA" = GENERIC_ALL	Rights = "GA" = GENERIC_ALL
ObjectGuid = None	ObjectGuid = None
InheritObjectGuid = None	InheritObjectGuid = None
AccountSID = "DC" = Domain Computer	AccoutSID = "NS" = Network Service

7.8 Troubleshooting

Issues may arise such as communication errors between the collectors and sources, authentication errors, and subscriptions errors. WinRM issues can be investigated using certain command line options. Understanding the WinRM capabilities and behaviors can be demystified by using the help option of WinRM. [82] If any troubleshooting is performed when the authentication recommendations provided in this guide are used, then append the–remote:TARGET option to the winrm command. The TARGET should be the local hostname if the issue involves the local machine.

---

[80] http://msdn.microsoft.com/en-us/library/aa374928(v=vs.85).aspx
[81] For curious readers, more information can be found at: http://msdn.microsoft.com/en-us/library/dd981030.aspx.
[82] winrm help

The listing below is not an exhaustive list to identify all issues with WinRM. These commands are helpful to diagnose common errors. [83][84][85]

winrm e winrm/config/listener

WinRM can enumerate all listeners that WinRM is currently using.

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).

The identify option provide 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.

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

winrm invoke restore 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

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

---

[83] http://blogs.technet.com/b/jonjor/archive/2009/01/09/winrm-windows-remote-management-troubleshooting.aspx
[84] http://msdn.microsoft.com/en-us/library/windows/desktop/ee309364(v=vs.85).aspx
[85] http://msdn.microsoft.com/en-us/library/windows/desktop/aa384295(v=vs.85).aspx#enabling_auth_options

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
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

7.8.1 Operational Logs

While troubleshooting an issue, it is natural for one to look at the logs to help to identify a problem. Event Forwarding and WinRM have operational logs that can be viewed in the Event Viewer or by using the command line tool wevtutil.exe. When WinRM is installed on a Windows XP client, an operational log is created.

The operational log files for the Event Collector, Event Forwarding, and WinRM services can be found by navigating to **Applications and Services Logs** in the Event Viewer on Windows Vista and later. The list below shows the location of the operational logs under **Applications and Services Logs**:

- **Microsoft > Windows > EventCollector > Operational**
- **Microsoft > Windows > Eventlog-ForwardPlugin > Operational**
- **Microsoft > Windows > Windows Remote Management > Operational**

The **Eventlog-ForwardPlugin** and **Windows Remote Management** operational logs are the locations that the local WinRM service will log to. 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 (e.g., Windows XP log file format).

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 /?

WinRM logs all activities to **Microsoft-Windows-Forwarding/Operational** in the Event Viewer on Windows XP.

7.8.2 Verify Compatibility HTTP(S) 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.

winrm e winrm/config/listener

An example output is shown below:

```
Listener [Source="GPO"]
    Address = *
    Transport = HTTP
    Port = 5985
    Hostname
    Enabled = true
    URLPrefix = wsman
    CertificateThumbprint
    ListeningOn = 127.0.0.1, 192.168.187.140, ::1, fe80::5efe:192.168.187.140%12 , fe80::a055:762d:1d27:6229%11

Listener [Source="Compatibility"]
    Address = *
    Transport = HTTP
    Port = 80
    Hostname
    Enabled = true
    URLPrefix = wsman
    CertificateThumbprint
    ListeningOn = 127.0.0.1, 192.168.187.140, ::1, fe80::5efe:192.168.187.140%12 , fe80::a055:762d:1d27:6229%11
```

7.8.3 WinRM Errors

There are numerous errors that WinRM can generate. Microsoft provides a table to easily identify common errors and solutions related to WinRM. [86] A list of event IDs associated with WinRM that applies to Windows Vista and above can be found on Microsoft's TechNet site. [22]

7.8.3.1 Creation of Subscription Errors

There are numerous possible errors that may arise during subscription creation. The following errors are possible common errors that may be encountered:

wecutil cs Subscriptions\Logons.xml

One possible error message:
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 get-subscriptionruntimestatus 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.

Another possible error message:

---

[86] http://social.technet.microsoft.com/wiki/contents/articles/13444.windows-server-2012-server-manager-troubleshooting-guide-part-ii-troubleshoot-manageability-status-errors-in-server-manager.aspx#Troubleshoot_manageablility_status_errors

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_{10} in the **Eventlog-ForwardingPlugin/Operational** log. The error code states that a cluster resource is not available. [87] 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.

7.8.3.2 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 part of local administration group, **WinRMRemoteWMIUsers__,** or domain administrator [88]
    - The administrator password **cannot** be **blank**
- Incorrect username or password
- WMI operations need permissions to allow secure connections [89]
- Windows Firewall service needs to be running

7.8.4 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 a XPath query was incorrect. The following table distinguishes differences between Windows XP and later versions of the Windows operating system:

---

[87] (Microsoft Corporation, 2012)
[88] http://msdn.microsoft.com/en-us/library/aa384295(v=vs.85).aspx
[89] http://msdn.microsoft.com/en-us/library/aa384424(v=vs.85).aspx

ID	Level	Event Log	Event Source	Operating System Version
101	Warning (3)	Microsoft-Windows-Forwarding/Operational	EventForwarder-Operational	Windows XP
101	Warning (3)	Eventlog-ForwardingPlugin/Operational	Eventlog-ForwardingPlugin	Windows Vista+

Table 27: XPath Errors based on OS Version

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. [90]

Discovering the error in event ID 101 on Windows XP can be found by combining two PowerShell commands:

(Get-EventLog –LogName "Microsoft-Windows-Forwarding/Operational" –EntryType "Warning" –Message "*one or more channels*" –newest 1 | ConvertTo-Xml -NoTypeInformation).Save("ABSOLUTE_PATH_TO_FILE.xml")

This command creates an XML file that provides the XML output of latest event ID 101. The XML output has a Property node named **ReplacementStrings** containing the erroneous query and the error code.

7.8.5 Windows XP Security Logs

The Windows XP security log not being forwarded to the collector may be the result of misconfiguration. Verify the following for the correct configuration:

- WinRM is running as Local system (LogOnAs value)
- If applicable, verify that the XPath query is correct
- Connection between source and collector

Within the **Microsoft-Windows-Forwarding/Operational** log, an error may have been generated regarding the description of an event has not been found.

7.9 WinRM and IIS

Windows Server 2008 R2 introduced a feature called WinRM IIS Extension. [91] 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.

---

[90] http://msdn.microsoft.com/en-us/library/windows/desktop/ms681384(v=vs.84).aspx
[91] http://technet.microsoft.com/en-us/library/dd759166.aspx

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. [92] WinRM IIS Extension is not installed by default and must be added via Server Manager as shown in Figure 23.

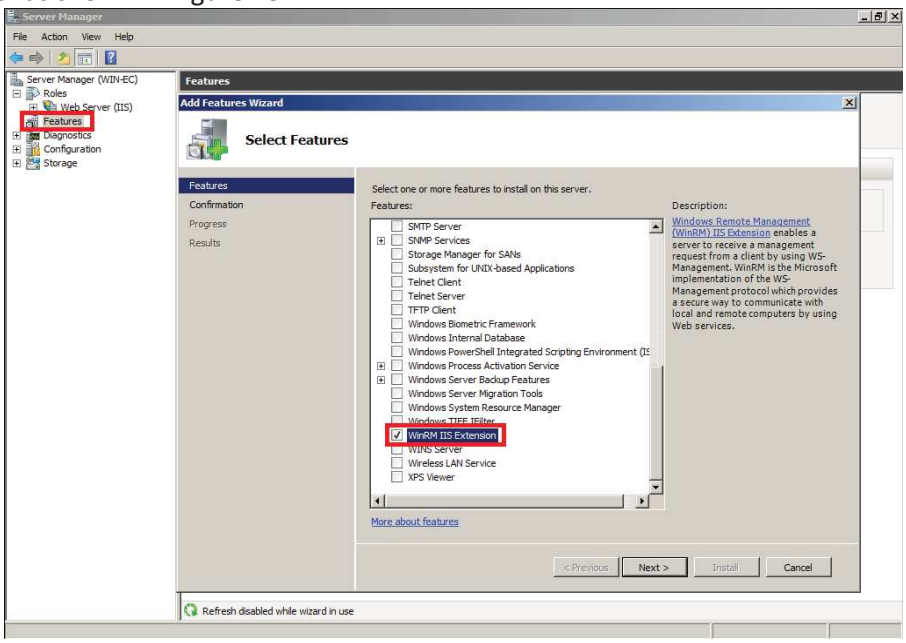


Figure 23: WinRM IIS Extension in Server Manager

7.10 Windows Server 2003 R2

This guide did not focus on Windows Server 2003 R2. The configuration of a collector on this operating system is similar to Windows Server 2008 R2. Certain features are not configurable in Windows Server 2003 R2 (e.g., Channel Binding Token or a policy for configuring listeners). Using Windows Server 2008 R2 as a policy for configuring listeners is recommended. Event Forwarding policies for sources are not available in Windows Server 2003 R2 as shown in Figure 24 and Figure 25. The feature is only available on Windows Vista and later. [93]

[92] winrm get wmi/root/cimv2/Win32_Service?Name=WinRM
[93] http://blog.technet.com/b/askds/archive/2011/08/29/the-security-log-haystack-event-forwarding-and-you.aspx

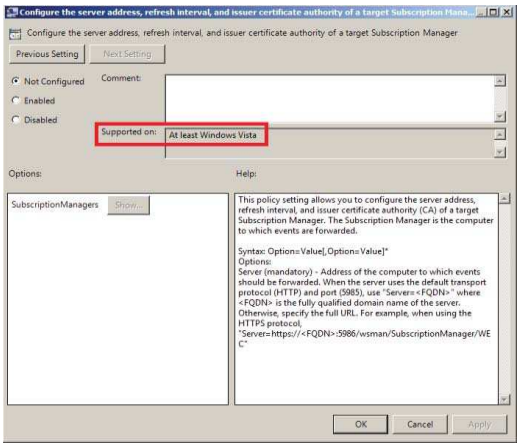


Figure 24: Subscription Manager Policy Supported OS Version

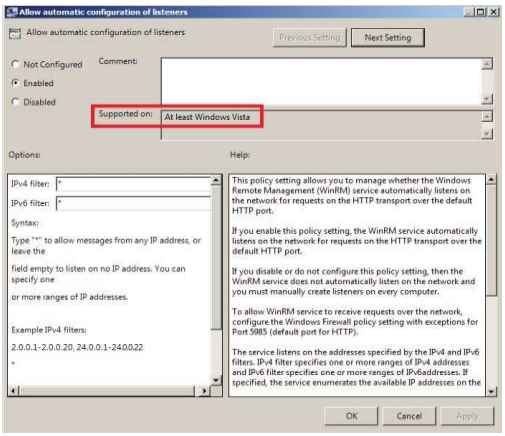


Figure 25: WinRM Listener Policy Supported OS Version

7.10.1 Installation of WinRM 1.1 on Windows Server 2003 R2

The installation of WinRM 1.1 on a Windows Server 2003 R2 can be done using an update from Microsoft. The update URL can be found in Windows Remote Management Versions section. It is possible to install WinRM using the Windows Server 2003 R2 installation CD; however, this will install WinRM 0.5. [94] Installing and using WinRM 2.0 is recommended.

8 Works Cited

Distributed Management Task Force, Inc. (2008, 02 12). *Web Services for Management (WS-Management) Specification.* Retrieved 10 01, 2012, from Distributed Management Task Force, Inc.: http://www.dmtf.org/standards/published_documents/DSP0226_1.0.0.pdf

Microsoft Corporation. (2012, 07 12). *[MS-CSSP]:Credential Security Support Provider (CredSSP) Procotol.* Retrieved 10 01, 2012, from Microsoft MSDN: http://msdn.microsoft.com/en-us/library/cc226764(v=prot.20).aspx

Microsoft Corporation. (2012, 07 15). *[MS-ERREF]: Windows Error Codes.* Retrieved 10 01, 2012, from Microsoft MSDN: http://msdn.microsoft.com/en-us/library/cc231196.aspx

Microsoft Corporation. (2012, 7 5). *[MS-WSMV]: Web Services Management Protocol Extensions for Windows Vista.* Retrieved 10 01, 2012, from Microsoft MSDN: http://msdn.microsoft.com/en-us/library/cc251526(prot.20).aspx

Microsoft Corporation. (2011, 10 8). *An update is available for the Windows Remote Management feature in Windows Server 2003 and in Windows XP.* Retrieved 10 01, 2012, from Microsoft Support: http://support.microsoft.com/kb/KB936059

Microsoft Corporation. (2012, 10 16). *Setting up a Source Initiated Subscription.* Retrieved 10 01, 2012, from MSDN Library: http://msdn.microsoft.com/en-us/library/bb870973(VS.85).aspx

[94] http://technet.microsoft.com/en-us/library/cc781099.aspx