Quest

Quest® InTrust 11.4.1

Preparing for Auditing and Monitoring PowerShell Activity

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.

Attn: LEGAL Dept

4 Polaris Way

Aliso Viejo, CA 92656

Refer to our Web site (https://www.quest.com) for regional and international office information.

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at https://www.quest.com/legal.

Trademarks

Quest, the Quest logo, and Join the Innovation are trademarks and registered trademarks of Quest Software Inc. For a complete list of Quest marks, visit https://www.quest.com/legal/trademark-information.aspx. All other trademarks and registered trademarks are property of their respective owners.

Legend

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

IMPORTANT, **NOTE**, **TIP**, **MOBILE**, or **VIDEO**: An information icon indicates supporting information.

InTrust Preparing for Auditing and Monitoring PowerShell Activity
Updated - December 2019
Version - 11.4.1

Contents

PowerShell Auditing and Real-Time Monitoring Overview

InTrust provides a number of disparate features that help you track and regulate the use of PowerShell in your environment. This document summarizes how InTrust covers these needs and what workflows you can configure to meet them.

To get ready for working with PowerShell logs, make sure the events you are interested in are audited. For details, see Making PowerShell Events Available.

After you have set up PowerShell logging, proceed to configure your InTrust workflows. See the following topics:

- Setting Up Monitoring for Suspicious PowerShell Activity
- Monitoring for PowerShell Downgrades
- Real-Time Collection and Forwarding of PowerShell Activity Data
- Task-Based Gathering of PowerShell Activity Data
- Analyzing PowerShell Events in Repository Viewer

Making PowerShell Events Available

NOTE: Whenever this document refers to PowerShell, the information also applies to PowerShell Core, unless PowerShell Core is specifically mentioned.

It is recommended that the computers you want to watch should have PowerShell 5.1 or later installed, because its logging facilities are far superior to earlier versions of PowerShell.

Configuration Through Group Policy

PowerShell activity must be logged on the computers you are interested in monitoring and gathering from. In the Group Policy Management console, configure the **Computer Configuration | Policies | Administrative Templates | Windows Components | Windows PowerShell** policy for these computers, as follows:

1. Set the **Turn on Module Logging** item to **Enabled** and specify ***** as the module name.
2. Set the **Turn on PowerShell Script Block Logging** item to **Enabled** and select the **Log script block invocation start / stop events** option for it.

Configuration Through the Registry

An alternative way to turn on the necessary logging options is to modify the registry. You can use the snippet below for this purpose.

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Policies\Microsoft\Windows\PowerShell]

[HKEY_LOCAL_
MACHINE\SOFTWARE\WOW6432Node\Policies\Microsoft\Windows\PowerShell\ModuleLogging]
"EnableModuleLogging"=dword:00000001

[HKEY_LOCAL_
MACHINE\SOFTWARE\WOW6432Node\Policies\Microsoft\Windows\PowerShell\ModuleLogging\Modul
eNames]
"*"="*"

[HKEY_LOCAL_
MACHINE\SOFTWARE\WOW6432Node\Policies\Microsoft\Windows\PowerShell\ScriptBlockLogging]
"EnableScriptBlockLogging"=dword:00000001
"EnableScriptBlockInvocationLogging"=dword:00000001
```

Save it as a text file with the **reg** extension. Once you have the file, use the **Merge** command on it to enable logging by modifying the registry.

Monitoring for PowerShell Downgrades

PowerShell logging capabilities have improved significantly from version to version. Early PowerShell versions audit much fewer kinds of events that more recent ones. However, PowerShell provides a way to run cmdlets in modes that are compatible with previous versions. Therefore, a common attack strategy is to use the compatibility mode in order to prevent logging of malicious activity. The "PowerShell downgrade attack detected" rule informs you about such threats.

You need the following:

- The real-time monitoring rule
  To monitor computers running PowerShell, use the **Advanced Threat Protection | Windows/AD Suspicious Activity | PowerShell | PowerShell downgrade attack detected** rule as the starting point.
- The site that contains the computers you want to watch
  The predefined "All Workstations" site may suit your needs. If you want to be more specific than that, create your own site and include the computers you need that run PowerShell or PowerShell Core.
- The real-time monitoring policy that applies the rule to the computers you want to protect
  Use the "Windows/AD Security: Detecting Common Attacks" real-time monitoring policy as your template and associate your policy with the correct site. If you have made separate sites for computers running PowerShell and PowerShell Core, include both sites in the policy.

How It Works

The "PowerShell downgrade attack detected" rule works by capturing PowerShell startup events. If PowerShell 2.0 or earlier is invoked, the alert is triggered.

Configuration Details

All of the required elements (the rule, policy and site listed above) are predefined in InTrust and associated with one another. If the existing configuration suits you, you can use the predefined objects directly. However, if you want to make adjustments, consider making copies of the objects, re-associating the copies and proceeding to work with them instead of the originals. InTrust treats all sites, rules and policies the same whether they are predefined or user-defined.

The following procedure assumes that you are working with your personalized copies of the objects listed above and doesn't mention the default predefined objects.

1. In InTrust Manager, under **Configuration | Sites | Microsoft Windows Network**, make sure your site is populated with the computers you want to monitor for PowerShell downgrades.
2. Open the properties of the rule. Click the **Notifications** tab and check that an email message is listed. Edit the message if necessary, as described in the *Message Templates* section of the Notification topic.
3. Click the **General** tab and select the **Enabled** option to activate the rule. After you close rule properties, commit the changes.

4. Open the properties of the real-time monitoring policy. On the **Rules** tab, make sure the necessary rule is specified.
5. On the **Sites** tab, specify the correct site or sites.
6. Select the **E-mail** tab, and click **Add** to specify who will receive the messages. For detailed instructions, see the *Notification Groups* section of the Real-Time Monitoring Overview topic.
7. Select the **General** tab and select the **Activate** option. After you close the properties dialog box, commit the changes. The configuration is now finished; InTrust agents will be installed automatically to the site computers to perform the monitoring.

You can modify such settings as alerting, response actions, rule activity time, or others at any time as necessary.

Setting Up Monitoring for Suspicious PowerShell Activity

This topic helps you minimize the impact of attacks based on PowerShell scripts. InTrust lets you thwart PowerShell-wielding attackers by setting up alerts and emergency response actions for whenever someone uses suspicious PowerShell commands.

You need the following:

- The real-time monitoring rule
  To monitor computers running PowerShell Core 6.0 and later, use the **Advanced Threat Protection | Windows/AD Suspicious Activity | PowerShell | Suspicious PowerShell Core activity** rule as the starting point. For computers running PowerShell 5.1 and earlier, use the **Advanced Threat Protection | Windows/AD Suspicious Activity | PowerShell | Suspicious PowerShell activity** rule.
- The site that contains the computers you want to watch
  The predefined "All Workstations" site may suit your needs. If you want to be more specific than that, create your own site and include the computers you need that run PowerShell or PowerShell Core. If some of these computers run PowerShell and the others PowerShell Core, create two sites for each set.
- The real-time monitoring policy that applies the rule to the computers you want to protect
  Use the "Windows/AD Security: Detecting Common Attacks" real-time monitoring policy as your template and associate your policy with the correct site. If you have made separate sites for computers running PowerShell and PowerShell Core, create a separate real-time monitoring policy for each site.
- Make sure PowerShell logging is enabled through group policy on the computers you want to monitor
  See Making PowerShell Events Available for details.

NOTE: If you implement this scenario on domain controllers or important member servers, you should whitelist the users who perform administrative actions. This isn't necessary for monitoring Windows workstations.

How It Works

The "Suspicious PowerShell activity" and "Suspicious PowerShell Core activity" rules work by capturing the use of PowerShell keywords from a specific set. These keywords are traces of PowerShell activity that might mean trouble if the activity is done by the wrong people. Keywords like this are typically found in modules that uncover system vulnerabilities, such as PowerSploit. These modules are often designed for testing purposes, but they are just as effective if the intent is malicious. If any of the keywords is detected in the PowerShell command prompt input or implicitly at any level during PowerShell operation, the rule is matched.

The rule has two parameters that control its operation:

- **Keywords**
  This is the list of keywords that the rule watches for, as described above. The default list is designed to cover a wide range of suspicious situations and minimize false positives. You can extend this list to anticipate other threats that you are aware of.

- **User Whitelist**
  These are the users who you trust with PowerShell. The rule will ignore whatever these users do with PowerShell. Use this parameter to specify accounts that have to perform administrative tasks on the monitored computers. Today, a lot of such tasks are performed through PowerShell by Windows without the user having to work with PowerShell directly. Take this into account when whitelisting users; this will help avoid false positives.

The rule can trigger the following response actions to stop an attacker in their tracks and help investigate the situation:

- Log off the offending user, provided that they are logged on interactively
- Disable the user account
- Stop and disable the Windows Remote Management service; if the user is logged on remotely, this will cut off their PowerShell session
- Enable auditing of a variety of events for analysis of that user's subsequent activity

By default, all of these response actions are disabled. You can enable any combination of them as necessary.

Configuration Details

All of the required elements (the rules, policy and site listed above) are predefined in InTrust and associated with one another. If the existing configuration suits you, you can use the predefined objects directly. However, if you want to make adjustments, consider making copies of the objects, re-associating the copies and proceeding to work with them instead of the originals. InTrust treats all sites, rules and policies the same whether they are predefined or user-defined.

The following procedure assumes that you are working with your personalized copies of the objects listed above and doesn't mention the default predefined objects.

1. In InTrust Manager, under **Configuration | Sites | Microsoft Windows Network**, make sure your site is populated with the computers you want to monitor for suspicious PowerShell activity.
2. Open the properties of the rule. Click the **Notifications** tab and check that an email message is listed. Edit the message if necessary, as described in the *Message Templates* section of the Notification topic.
3. Select the **Response Actions** tab, and select the check boxes next to any of the response actions you need.
4. Click the **General** tab and select the **Enabled** option to activate the rule. After you close rule properties, commit the changes.
5. Open the properties of the real-time monitoring policy. On the **Rules** tab, make sure the necessary rule is specified.
6. On the **Sites** tab, make sure the correct site is specified.
7. Select the **E-mail** tab, and click **Add** to specify who will receive the messages. For detailed instructions, the *Notification Groups* section of the Real-Time Monitoring Overview topic.
8. Select the **General** tab and select the **Activate** option. After you close the properties dialog box, commit the changes. The configuration is now finished; InTrust agents will be installed automatically to the site computers to perform the monitoring.

You can modify such settings as alerting, response actions, rule activity time, or others at any time as necessary.

Real-Time Collection and Forwarding of PowerShell Activity Data

This topic explains how you can continuously gather PowerShell events to InTrust repositories and, if necessary, forward it to a SIEM solution of your choice for analysis. The functionality described here is part of the feature set provided by InTrust Deployment Manager. To proceed, run this console and connect to your InTrust organization.

Decide Where to Store the Events

It's up to you if you want to store your PowerShell audit data in one of your existing repositories or a dedicated repository. A dedicated repository is recommended if you intend to forward the incoming data to a SIEM solution.

If you want to create a new repository, go to the **Storage** view, click the **New** button and follow the steps. For details, see Managing Repositories.

Set Up Gathering

You need a dedicated collection for PowerShell events. Go to the **Collections** view and take the following steps:

1. Right-click the **Collections** node, select **New Windows Collection** and follow the steps.
2. On the **Specify Computers** step, supply the computers that you want to collect PowerShell logs from.
3. On the Data Sources and Repository step:
    a. Select the **Windows PowerShell Operational Log** and **Windows PowerShell Core Operational Log** data sources.
    b. Make sure the **If any of the selected data sources cannot be found, consider this an error** option is cleared.
    NOTE: Before Update 1 for InTrust 11.4.1, this option was labeled **Suppress errors from non-existent data sources** and did the opposite. If you are using InTrust 11.4.1 without Update 1, make sure **Suppress errors from non-existent data sources** is selected.
    c. Select the repository you decided on earlier.
4. Finish the steps.

For more details, see Managing Collections.

Set Up Forwarding

If you want to forward your collected PowerShell data, take the following steps:

1. Go to the **Storage** view and select the repository that stores PowerShell data.
2. In the right pane, in the **Forwarding** block of options, click **Edit** and select **Enable forwarding**.
3. Configure your forwarding settings as necessary. For details, see Turning Forwarding On and Off.
4. Click **Apply** to put your changes into effect.

Task-Based Gathering of PowerShell Activity Data

If you just want to archive your PowerShell audit data without real-time awareness of what is going on, you may want to use task-based gathering. This kind of gathering is also the only option if you want to collect data without installing InTrust agents on the audited computers.

The functionality described here is part of the feature set provided by InTrust Manager. To proceed, run this console and connect to your InTrust organization.

To implement the simplest configuration for this scenario, create the following:

- One InTrust site that contains all the computers you want to collect PowerShell events from, under the **Quest InTrust Manager | Configuration | Sites | Microsoft Windows Network** node.
- One gathering policy that defines how to collect logs, under the **Quest InTrust Manager | Gathering | Gathering Policies | Microsoft Windows Network** node.
  When prompted to include data sources in the policy, select the **Windows PowerShell Operational Log** and **Windows PowerShell Core Operational Log** data sources.
- If you want a dedicated store for your PowerShell event data, a new repository under the **Quest InTrust Manager | Configuration | Data Stores | Repositories** node.
- One InTrust scheduled task under the **Quest InTrust Manager | Workflow | Tasks** node.
  The schedule for the task should be enabled and set to a time that is convenient to you—for example, sometime during off-peak hours.
- One gathering job within the scheduled task.
  In the configuration of the gathering job, specify the repository you decided on, the site you created and the policy you created.

After you have set up these configuration objects, click the **Commit** button in the toolbar to put the workflow in effect.

For details about the particular procedures involved in this configuration, see the following topics:

- Auditing Guide
- InTrust Sites
- Creating and Editing Repositories

Analyzing PowerShell Events in Repository Viewer

To match the **Windows PowerShell Operational Log** and **Windows PowerShell Core Operational Log** data sources that are available out of the box, Repository Viewer provides the **Threat Hunting | Windows | PowerShell** search folder with dedicated predefined searches. You can use these searches directly or make custom searches based on them to better suit your needs.

For details about running searches and preparing scheduled reports on your repository data, see Searching for Events in Repository Viewer.

We are more than just a name

We are on a quest to make your information technology work harder for you. That is why we build community-driven software solutions that help you spend less time on IT administration and more time on business innovation. We help you modernize your data center, get you to the cloud quicker and provide the expertise, security and accessibility you need to grow your data-driven business. Combined with Quest's invitation to the global community to be a part of its innovation, and our firm commitment to ensuring customer satisfaction, we continue to deliver solutions that have a real impact on our customers today and leave a legacy we are proud of. We are challenging the status quo by transforming into a new software company. And as your partner, we work tirelessly to make sure your information technology is designed for you and by you. This is our mission, and we are in this together. Welcome to a new Quest. You are invited to Join the Innovation™.

Our brand, our vision. Together.

Our logo reflects our story: innovation, community and support. An important part of this story begins with the letter Q. It is a perfect circle, representing our commitment to technological precision and strength. The space in the Q itself symbolizes our need to add the missing piece — you — to the community, to the new Quest.

Contacting Quest

For sales or other inquiries, visit www.quest.com/contact.

Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to-videos
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product