Tanium™ Incident Response User Guide

Version 5.5.0

April 09, 2020

The information in this document is subject to change without notice. Further, the information provided in this document is provided "as is" and is believed to be accurate, but is presented without any warranty of any kind, express or implied, except as provided in Tanium's customer sales terms and conditions. Unless so otherwise provided, Tanium assumes no liability whatsoever, and in no event shall Tanium or its suppliers be liable for any indirect, special, consequential, or incidental damages, including without limitation, lost profits or loss or damage to data arising out of the use or inability to use this document, even if Tanium Inc. has been advised of the possibility of such damages.

Any IP addresses used in this document are not intended to be actual addresses. Any examples, command display output, network topology diagrams, and other figures included in this document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Please visit https://docs.tanium.com for the most current Tanium product documentation.

Table of contents

Incident Response overview

Tanium™ Incident Response consists of several solutions that you can deploy to manage incidents across the enterprise.

Incident Response

With the core Incident Response (IR) solution, you deploy a set of IR tools to each endpoint. With these tools on the endpoints, you can:

- Scope and hunt for incidents across the enterprise by searching for evidence from live system activity and data at rest with simple natural language queries.
- Examine and parse dozens of forensic artifacts on Windows, Mac, and Linux systems.
- Identify outliers and anomalies by collecting and comparing data across systems in real time.
- Build saved queries and dashboards to continuously monitor endpoints for malicious activity aligned to key phases of the intrusion lifecycle.

More information:

- Install Tanium Incident Response on page 20
- Deploying IR tools on page 26
- Using IR sensors and packages on page 27

Index

Index the file systems on Tanium Client endpoints that are running Windows, Linux, or macOS operating systems. File system inventory, hashes, and magic numbers are recorded in an SQLite database for investigation of threat indicators.

More information:

- Install Index on page 21
- Indexing file systems on page 36

Live Response

Configure what information to collect from suspicious Windows, Linux, and macOS endpoints for further forensic analysis and data correlation. Investigate potentially compromised systems with a customizable and extensible framework.

More information:

- Installing Incident Response solutions on page 20
- Collecting data with Live Response on page 48

Quarantine

Isolate targeted machines from communicating with unapproved network addresses or IP ranges by applying network quarantine. You can apply a quarantine to Windows, Linux, and macOS endpoints that show evidence of compromise or other suspicious activity. You can use Tanium Quarantine to apply, remove, and test for quarantine.

More information:

- Install Quarantine on page 22
- Isolating endpoints on page 65

Integration with Detect

In cases where a wider search or a search for a large or dispersed data set is required, you can integrate Tanium™ Detect into the hunt. For example, to exhaustively search for hundreds of hashes, or to perform recursive searches in nested directories, use Detect to create a custom IOC intel document for quick scans or background scans. For more information about Detect, see the Tanium Detect User Guide.

Getting started

1. Install Tanium™ Incident Response solutions.
   More information: Installing Incident Response solutions on page 20
2. Deploy IR tools to the endpoints that are running Tanium Client.
   More information: Deploying IR tools on page 26
3. Use IR sensors to scope and hunt incidents, examine forensic artifacts, collect real time data, and monitor endpoints for malicious activity.
   More information: Using IR sensors and packages on page 27
4. Index operating systems.
   Use the Tanium Index solution to index the local file systems on Tanium Client endpoints that are running Windows or Mac OS X operating systems. After the file systems are indexed, you can use sensors to query specific file attributes, such as path, hash, and modified dates.
   More information: Indexing file systems on page 36
5. Collect files from endpoints.
   - Move a set of arbitrary files. You can define this list of files with a comma-separated list.
     More information: Copying IR data to a central location on page 30
   - With Live Response, you can configure what files and what destinations you want to use to collect data from endpoints.
     More information: Collecting data with Live Response on page 48
6. Isolate endpoints.
   You can apply a quarantine on endpoints that show evidence of compromise or other suspicious activity. When applied, the endpoint cannot communicate with any resource other than the Tanium Server.
   More information: Isolating endpoints on page 65

Incident Response requirements

Review the requirements before you install and use Incident Response.

Tanium dependencies

Component	Requirement
Platform	Version 7.2 or later.
Tanium Client	Version 6.0.314.1396 or later.
License	For information about licensing Incident Response, contact your Technical Account Manager (TAM). The license for Incident Response includes the following solutions: - Tanium Incident Response - Tanium Quarantine (Quarantine) - Tanium Live Response (Live Response) - Tanium Index (Index) - Windows Security Patch Management (for more information, see Tanium Knowledge Base)
Tanium™ Trace	Version 2.3.2.0004 or later is required for real-time events on Linux endpoints with Tanium Index 2.0.0 or later.

Third-party software requirements

For Tanium Incident Response, the required third-party software is installed automatically.

However, the IR Gatherer solution has third-party software requirements that are not installed automatically. The related documentation includes instructions to download the software and include it in packages that are distributed to the endpoints.

Endpoints

Supported operating systems

The following endpoint operating systems are supported by Incident Response, Copy tools, Quarantine, Index, and Live Response:

- Windows (A minimum of Windows 7 or Windows Server 2008 R2 with SP1 is required.)

- MacOS
- Linux

See the documentation for each IR solution for specific version numbers.

Disk space requirements

If a solution is not listed, the required disk space is minimal.

IR Solution	Disk Space
Index	1 GB free space

Host and network security requirements

Specific ports and processes are needed to run Incident Response.

Ports

The following ports are required for IR communication.

IR Solution	Port	Direction	Purpose
Live Response	443 (S3), 22 (SFTP/SCP), or 445 (SMB)	Outbound	Outbound connections over ports depending on how the collected data is being transferred.

Security exclusions

If security software is in use in the environment to monitor and block unknown host system processes, your security administrator must create exclusions to allow the Tanium processes to run without interference.

Table 1: Incident Response security exclusions

Target Device	Process
Windows x86	*<Tanium Client>*\Tools\IR\TaniumPersistenceAnalyzer.exe
	<Tanium Client>\Tools\EPI\TaniumExecWrapper.exe
	<Tanium Client>\Tools\IR\TaniumExecWrapper.exe
	<Tanium Client>\Tools\IR\TanFileInfo.exe
	<Tanium Client>\Tools\IR\TaniumHandle.exe
	<Tanium Client>\Tools\IR\TanListModules.exe
	<Tanium Client>\Tools\EPI\TaniumEndpointIndex.exe
	<Tanium Client> \Tools\IR\PowerForensics\PowerForensics.dll
	<Tanium Client>\Downloads\Action_ *nnn*\Winpmem.gb414603.exe [1]
	<Tanium Client>\Downloads\Action_ *nnn*\TaniumFileTransfer.exe[1]
	<Tanium Client>\Python27\TPython.exe(7.2.x clients)
	<Tanium Client>\Python38\TPython.exe(7.4.x clients)
	<Tanium Client>\Python38*.dll(7.4.x clients)

Target Device	Process
Windows x64	*<Tanium Client>*\Tools\IR\TaniumPersistenceAnalyzer.exe
	<Tanium Client>\Tools\EPI\TaniumExecWrapper.exe
	<Tanium Client>\Tools\IR\TaniumExecWrapper.exe
	<Tanium Client>\Tools\IR\TanFileInfo.exe
	<Tanium Client>\Tools\IR\TaniumHandle.exe
	<Tanium Client>\Tools\IR\TanListModules.exe
	<Tanium Client>\Tools\EPI\TaniumEndpointIndex.exe
	<Tanium Client>\Tools\IR\PowerForensics\PowerForensics.dll
	<Tanium Client>\Downloads\Action_*nnn*\Winpmem.gb414603.exe [1]
	<Tanium Client>\Downloads\Action_*nnn*\TaniumFileTransfer.exe[1]
	<Tanium Client>\Python27\TPython.exe(7.2.x clients)
	<Tanium Client>\Python38\TPython.exe(7.4.x clients)
	<Tanium Client>\Python38*.dll(7.4.x clients)

Target Device	Process
Mac OS	*<Tanium Client>*/Tools/EPI/TaniumExecWrapper
	<Tanium Client>/Tools/IR/TaniumExecWrapper
	<Tanium Client>/Tools/EPI/TaniumEndpointIndex
	<Tanium Client>/Downloads/Action_*nnn*/surge-collect[1,2]
	<Tanium Client>/Downloads/Action_*nnn*/surge.dat[1,2]
	<Tanium Client>/Downloads/Action_*nnn*/osxpmem.app/osxpmem[1]
	<Tanium Client>/Downloads/Action_*nnn*/TaniumFileTransfer[1]
	<Tanium Client>/python27/python (7.2.x clients)
	<Tanium Client>/python38/python (7.4.x clients)
Linux x86	*<Tanium Client>*/Tools/EPI/TaniumExecWrapper
	<Tanium Client>/Tools/IR/TaniumExecWrapper
	<Tanium Client>/Tools/EPI/TaniumEndpointIndex
	<Tanium Client>/Downloads/Action_*nnn*/surge-collect[1,2]
	<Tanium Client>/Downloads/Action_*nnn*/surge.dat[1,2]
	<Tanium Client>/Downloads/Action_*nnn*/linpmem-<version>.bin[1]
	<Tanium Client>/Downloads/Action_*nnn*/TaniumFileTransfer[1]
	<Tanium Client>/python27/python (7.2.x clients)
	<Tanium Client>/python38/python (7.4.x clients)

Target Device	Process
Linux x64	`<Tanium Client>/Tools/EPI/TaniumExecWrapper`
	`<Tanium Client>/Tools/IR/TaniumExecWrapper`
	`<Tanium Client>/Tools/EPI/TaniumEndpointIndex`
	`<Tanium Client>/Downloads/Action_nnn/surge-collect`[1,2]
	`<Tanium Client>/Downloads/Action_nnn/surge.dat`[1,2]
	`<Tanium Client>/Downloads/Action_nnn/linpmem-<version>.bin`[1]
	`<Tanium Client>/Downloads/Action_nnn/TaniumFileTransfer`[1]
	`<Tanium Client>/python27/python` (7.2.x clients)
	`<Tanium Client>/python38/python` (7.4.x clients)

[1] = Where *nnn* corresponds to the action ID.

[2] = Exception is required if Volexity Surge is used for memory collection.

Internet URLs

If security software is deployed in the environment to monitor and block unknown URLs, your security administrator must whitelist the following URL:

- content.tanium.com

User role requirements

Table 2: Incident Response Advanced user role permissions

Permission	Content Set for Permission	Incident Response Administrator	Incident Response User	Incident Response Read Only User
Ask Dynamic Questions		✓*	✓*	✓*
Read Action	Incident Response	✓	✗	✗

Permission	Content Set for Permission	Incident Response Administrator	Incident Response User	Incident Response Read Only User
Read Package	Incident Response	✓*	✓	✗
Read Saved Question	Incident Response	✓*	✓	✓
Read Sensor	Incident Response	✓*	✓	✓
Write Action	Incident Response	✓	✗	✗
Write Action for Saved Questions	Incident Response	✓	✗	✗
Write Package	Incident Response	✓	✗	✗
Write Saved Question	Incident Response	✓	✗	✗
Write Sensor	Incident Response	✓	✗	✗

‡ To install IR solutions, you must have the reserved role of Administrator.

* Requires permissions for the Interact module to ask questions, see results, and drill-down to endpoints.

Table 3: Index Advanced user role permissions

Permission	Content Set for Permission	Index Administrator	Index User	Index Read Only User
Ask Dynamic Questions		✓*	✓*	✓*
Read Action	Index	✓	✗	✗
Read Package	Index	✓*	✓	✗
Read Saved Question	Index	✓*	✓*	✓
Read Sensor	Index	✓*	✓*	✓
Write Action	Index	✓	✗	✗

Permission	Content Set for Permission	Index Administrator	Index User	Index Read Only User
Write Action for Saved Questions	Index	✓	✗	✗
Write Package	Index	✓	✗	✗
Write Saved Question	Index	✓	✓	✗
Write Sensor	Index	✓	✓	✗

‡ To install IR solutions, you must have the reserved role of Administrator.

* Requires permissions for the Interact module to ask questions, see results, and drill-down to endpoints.

Installing Incident Response solutions

Install Tanium Incident Response

Use the Tanium™ Incident Response solution to scan and hunt for incidents, examine forensic artifacts, and collect system data for analysis.

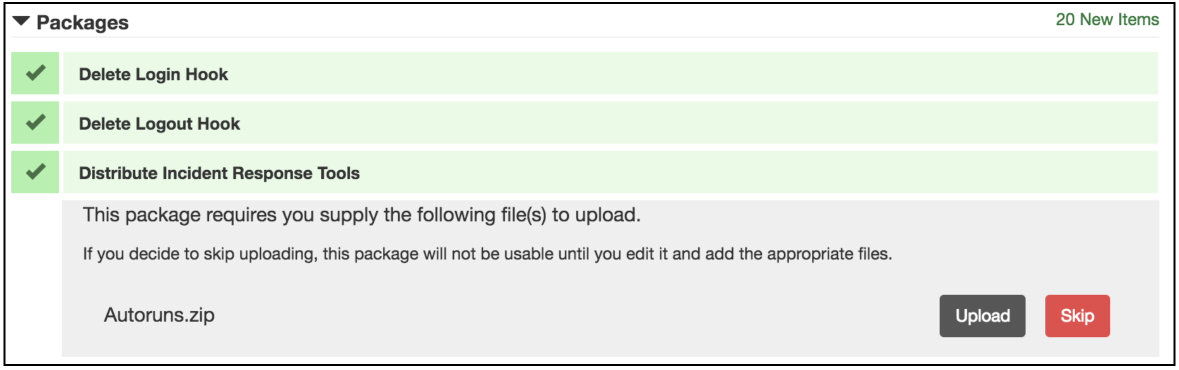
> **Note:** The procedures and screen captures that are in the documentation are for Version 7 and later. Version 6 procedures and screens might vary.

Before you begin

- To use Autoruns content, download the latest version of `Autoruns.zip` file from Autoruns for Windows. Upload this file during the import of the solution.
- You must be assigned the Administrator reserved role to import a Tanium solution module or content pack.

Import the Tanium Incident Response solution

1. From the Main menu, click **Tanium Solutions**.
2. In the Incident Response section, click **Import** *Version*.
3. Review the list of categories, dashboards, saved questions, saved actions, packages, sensors, and content set roles.
4. Upload the `Autoruns.zip` file.



This package requires you supply the following file(s) to upload.

If you decide to skip uploading, this package will not be usable until you edit it and add the appropriate files.

Autoruns.zip

> **IMPORTANT:** Uploading the `Autoruns.zip` file is required for the Autoruns content to work properly.

5. Complete the import.

- For platform version 6.5 and 7.0, click **Proceed with Import**
- For platform version 7.1.314.3071 and later, enable **Include content set overwrite** and click **Import**.

  For more information, see the Tanium Core platform User Guide: Align content for modules.

6. Verify that the IR sensors and packages were installed.
   a. From the Main menu, click **Content > Packages**.
   b. Search for `Incident Response`.
   c. From the Main menu, click **Content > Sensors**.
   d. In the Category column, click the menu button and create a filter that contains `Incident Response`.

What to do next

Deploy the IR tools to the endpoint. For more information, see Deploying IR tools on page 26.

You can also install other IR solutions.

Install Index

Tanium Index is a solution that runs locally on endpoints to gather, compute, and provide information that is useful to detect and investigate threat indicators for files at rest. Index is optimized to minimize endpoint resource utilization and work with journaling file systems when available. The solution performs the following actions:

- Indexes local file system
- Computes file hashes
- Records file attributes and magic numbers

Obtain the Tanium Index solution

To obtain the Tanium Index solution, contact your Technical Account Manager (TAM). Your TAM will provide you with the solution and instructions for importing the solution and any associated content.

After you import Tanium Index, the Index sensors, packages, and scheduled actions are viewable in the console.

What to do next

By default, the actions to distribute Index to the endpoints are disabled. Enable the **Deploy Distribute Tanium Endpoint Index Tools** scheduled action to distribute Index endpoint tools your endpoints. Then, create the custom configuration file and enable the

Distribute Tanium Endpoint Index Config action with the new file. For more information about enabling Index on the endpoints, see Indexing file systems on page 36.

Install Quarantine

Tanium Quarantine is a collection of packages and sensors that you can use to isolate endpoints that show evidence of compromise or other suspicious activity. You can use Quarantine to apply, remove, and test for quarantine. Quarantine is supported on Windows, Linux, and Mac OS X endpoints.

Import the Tanium Quarantine solution

Install the Tanium Quarantine solution by importing the associated content from the Tanium Solutions page.

1. From the Main menu, click **Tanium Solutions**.
2. In the Tanium Content section, select the **Quarantine** row and click **Import Solution**.
3. Review the list of saved actions, packages, and sensors and click **Proceed with Import**.
4. When the import is complete, you are returned to the Solutions page. Verify that the values in the **Available Version** and **Imported Version** columns match.

What to do next

For more information about Quarantine, see Isolating endpoints on page 65.

Upgrading Incident Response

You can upgrade the Tanium™ Incident Response module, or any of the individual solutions.

Upgrade Tanium Incident Response

Before you begin

- To use Autoruns content, download the latest version of `Autoruns.zip` file from Autoruns for Windows. Upload this file during the import of the solution.
- You must be assigned the Administrator reserved role to import a Tanium solution module or content pack.

Import Tanium Incident Response updates

1. From the Main Menu, click **Tanium Solutions**.
2. In the Incident Response section, click **Upgrade to** *Version*.
3. Review the list of saved actions, packages.
    - For platform version 6.5 and 7.0, click **Proceed with Import**
    - For platform version 7.1.314.3071 and later, enable the **Include content set overwrite** checkbox and click **Proceed with Import**.

      For more information, see the Tanium Core platform User Guide: Align content for modules.

   Uploading the `Autoruns.zip` file is required for the Autoruns content to work properly.
4. To confirm the upgrade, return to the Tanium Solutions page and check the Installed version for Incident Response.

Upgrade IR solutions

Before you run an upgrade, you might want to back up configuration files that are not preserved during the upgrade process for Index and Live Response. For more information, see Upgrade Tanium Index on page 24 and Preserve configuration files before upgrading Live Response on page 25.

1. From the Main Menu, click **Tanium Solutions**.
2. In the Tanium Content section, select the row and click **Upgrade Solution**.

3. Review the list of saved actions, packages.
   - For platform version 6.5 and 7.0, click **Proceed with Import**
   - For platform version 7.1.314.3071 and later, enable the **Include content set overwrite** checkbox and click **Proceed with Import**.

     For more information, see the Tanium Core platform User Guide: Align content for modules.

     Uploading the `Autoruns.zip` file is required for the Autoruns content to work properly.
4. When you are returned to the Solutions page, check the installed version of the solution.

Upgrade Tanium Index

Before and after upgrading Index, there are some additional steps to take.

Preserve configuration files

The custom Index `config.ini` file in the configuration packages is not preserved when you upgrade the Index solution. You must back up the file before upgrading, and re-add the file to your packages after the upgrade.

1. Save your custom `config.ini` file.
2. Delete any scheduled actions that are going to distribute the `config.ini` file.
   - **Deploy Distribute Tanium Endpoint Index Tools**
   - **Deploy Distribute Tanium Endpoint Index Tools for Mac**
   - **Deploy Distribute Tanium Endpoint Index Config**
   - **Deploy Distribute Tanium Endpoint Index Config for Mac**
3. Upgrade the Tanium Index solution. For more information, see Upgrade IR solutions on page 23.
4. Edit the appropriate Index packages to include the custom `config.ini` file.
5. Create new scheduled actions to distribute the updated packages.

For more information about editing packages to distribute a custom `config.ini` file, see Customize Index endpoint settings on page 40.

Recreate content and deploy tools

After upgrading, you must update the Tanium components that reference Index content.

1. Delete and recreate any saved questions that reference Index sensors.
2. Delete and recreate any scheduled actions that reference Index packages or sensors.
3. If not completed already, re-deploy endpoint tools.

4. (Optional) To capture all hard links on Windows endpoints, initiate a reindex of the file system.
   a. Deploy the **Delete Tanium Endpoint Index database** package.
   b. Use the appropriate saved action to start indexing.

   For Windows endpoints, typically a reindex occurs only if Index has lost its place in the file system; otherwise, Index only checks for new information. For Mac endpoints, deployment causes the directory scan to start from the beginning.

Preserve configuration files before upgrading Live Response

Before you upgrade Live Response, download any customized configuration files. These files are not preserved when you upgrade.

Alternatively, you can host any customized configuration files in a remote location and attach the files to the package.

1. Open the **Live Response - Windows** package.
2. Download all customized collection and transfer configuration files, such as `Custom_Collection.json`, `SCP.json`, and any SSH/Amazon S3 private key files.
3. Upgrade the Live Response solution. For more information, see Upgrade IR solutions on page 23.
4. Open the **Live Response - Windows** package again.
5. Upload the customized configuration files to the package.
6. Create new scheduled actions to distribute the updated packages.

Deploying IR tools

The IR tools are a package that is deployed on the endpoint. The package includes scripts and utilities that enable the functionality of IR module. The tools must be fully deployed for IR to function.

For Windows, the **Distribute Incident Response Tools** package is automatically deployed to endpoints by a scheduled action that is enabled by default. Unless you want to make changes to the package or action schedule, no action is required.

The **Distribute IR Tools (Mac)** and **Distribute IR tools (Linux**) scheduled actions are disabled by default. You must enable these scheduled actions before the tools are deployed to endpoints.

Before you begin

The Tanium Incident Response solution must be installed. For more information, see Install Tanium Incident Response on page 20.

Updating scheduled actions

You can enable, disable, or edit the scheduled actions that deploy IR tools. When the scheduled action is enabled, the IR tools are distributed to any endpoints that do not have them already installed. The frequency of the distribution is defined in the scheduled action.

1. From the Main Menu, click **Actions > Scheduled Actions**.
2. Search for the action **Distribute Incident Response Tools**, **Distribute IR Tools (Mac)**, or **Distribute IR tools (Linux)**.
3. To change the schedule for the action, click **Edit**.
4. To enable the action, select the row and click **More > Enable Action**.

Verify that IR tools are deployed on the endpoints

To check that IR tools are deployed on the endpoints, you can ask the question: `Get Has Incident Response Tools from all machines`. You can then drill down on the rows that are returned to display more information about the endpoints that need to have the IR tools deployed.

Using IR sensors and packages

Use IR sensors for rapid response to and scoping of incidents. Incident response can require computationally-intensive hashing algorithms and extensive file system scans. For this reason, IR sensors are written with a narrow scope to minimize processing and retrieve specific information within seconds. Few search operations are recursive and most sensors perform a hexadecimal search or hash match a single file and target a single directory. This strategy takes advantage of the Tanium linear chaining topology to rapidly deliver critical information at enterprise scale.

About deploying parameterized sensors as actions

Sensors that require extensive computational resources across the security enterprise, for example, sensors that hash files and perform binary searches, are deployed as actions. Deploying parameterized sensors as actions increases the speed of larger tasks, including:

- Searching across directories for binary data
- Matching the hash values of files across many directories
- Hashing and matching executables and their loaded modules

Actions are not processed one at a time. Short actions run at the same time as longer actions. Because they are not strictly queued, shorter actions are not delayed by the execution of more extensive actions.

Actions do not time out. Because the processing time of an action depends on the nature of the task, an action is considered complete when the job begins. The results, however, might not be immediately available.

When you deploy the action, you must provide an IR job ID. Then, you can view results files from Windows-based endpoints with the **Incident Response Job Results** sensor by specifying the job ID as a parameter. You can retrieve and copy job results files to a central location by using one of the platform-specific collection actions.

Table 4: Use Cases for IR content

Task	Question	Package / Sensor
Retrieve a list of all running processes on all endpoints with their hashes	`Get Running Processes with Hash from all machines`	Sensor: **Running Processes with Hash**

Task	Question	Package / Sensor
Retrieve the currently running processes matching a specific MD5 hash	`Get MD5 Hash Match Files Executing from all machines`	Package: **Incident Response - MD5 Hash Match Files Executing**
Display IR job results in Tanium Console	`Get Incident Response Job Results from all machines`	Sensor: **Incident Response Job Results**
Copy IR job results for Windows-based endpoints to a central location	`Get Has Incident Response ID Files from all machines`	Package: **IR Gatherer - Collect Info to Central Server**

Before you begin

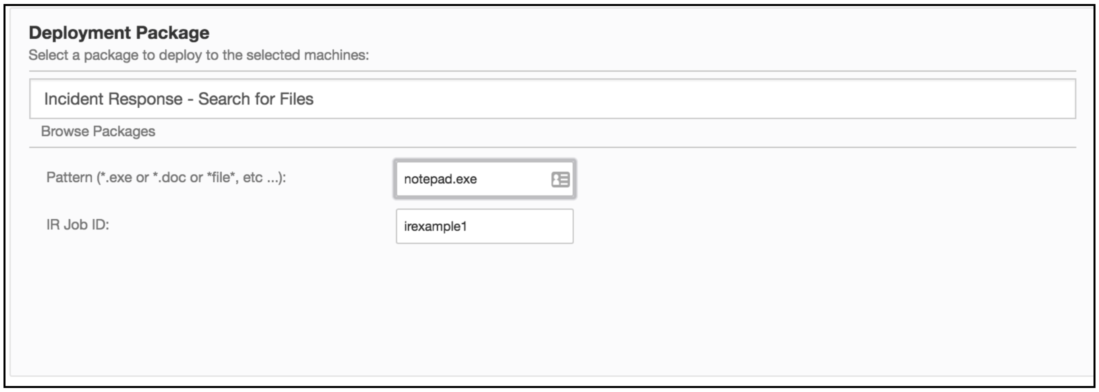
- The Tanium Incident Response solution must be installed. For more information, see Install Tanium Incident Response on page 20.
- The IR tools must be deployed to the endpoints. For more information, see Deploying IR tools on page 26.

Deploy a parameterized sensor as an action

1. Identify the endpoints that you want to target.
   a. Ask a question to return a set of endpoints.
   b. Select the endpoints and click **Deploy Action**.
2. Specify the parameterized sensor.
   a. Type the name of the parameterized sensor in the **Deployment Package** field.

      For example, type: `Incident Response - Search for Files`.

b. Specify parameters for the sensor.



Deployment Package
Select a package to deploy to the selected machines:

Incident Response - Search for Files

Browse Packages

Pattern (*.exe or *.doc or *file*, etc ...): notepad.exe

IR Job ID: irexample1

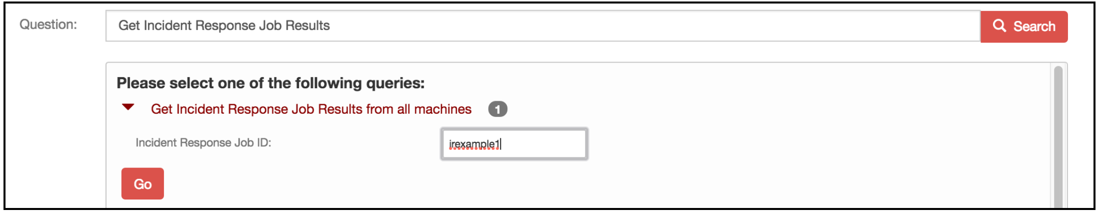
For the **Incident Response - Search for Files** sensor, indicate a **Pattern** of files to match and the **IR Job ID**.

> **IMPORTANT:** The **IR Job ID** can be any value that you choose. Use this value to get the results of the action. The value must be unique. If two actions share the same job ID, the files identified by those actions might be destroyed. Remember the value so that you can retrieve the job results later.

c. Complete deployment of the action. Click **Deploy Action**.

3. Get the results of the parameterized sensor action.
   a. Ask the question: `Get Incident Response Job Results from all machines`
   b. Specify the **Incident Response Job ID**.

   The value for the job ID is the same value that you specified when you deployed the action.



Question: Get Incident Response Job Results

Please select one of the following queries:

Get Incident Response Job Results from all machines 1

Incident Response Job ID: irexample1

   c. Click **Go**.

Reference: IR sensors and packages

For details about the parameters for each IR sensor and package, see [Tanium Support Knowledge Base: Tanium IR Reference]() (login required).

Copying IR data to a central location

Tanium™ Incident Response includes Tanium Copy Tools. Use these tools to copy files that you specify from endpoints to a central location. When you run a Copy action, you must specify the target endpoints for the operation and the method of transport.

Before you begin

- IR must be installed on the server. For more information, see Install Tanium Incident Response on page 20.
- IR tools must be deployed the endpoints. For more information, see Deploying IR tools on page 26.

Set up a copy location and service account

You must have a server location to which you are copying the IR data.

> **IMPORTANT:** Assign write-only access to the account used by IR Gatherer when performing copy operations. Read and append access for this account are not necessary and present a security risk due to IR Gatherer operating in a potentially hostile environment.
>
> The accounts used for file transfer should expire as soon as possible after use because log data might include the user names and passwords for these access accounts.

Copy location file transfer methods

The following methods of transfer are available:

File Transfer Protocol (FTP) / Secure Copy Protocol (SCP)

Requires a user account limited to write access. Do not assign read, append and delete permissions to the user. An account that expires soon after creation is preferred.

Secure File Transport Protocol (SFTP)

Requires a user account limited to write access. Do not assign read, append and delete permissions. An account that expires soon after creation is preferred.

Server Block Message (SMB) Protocol

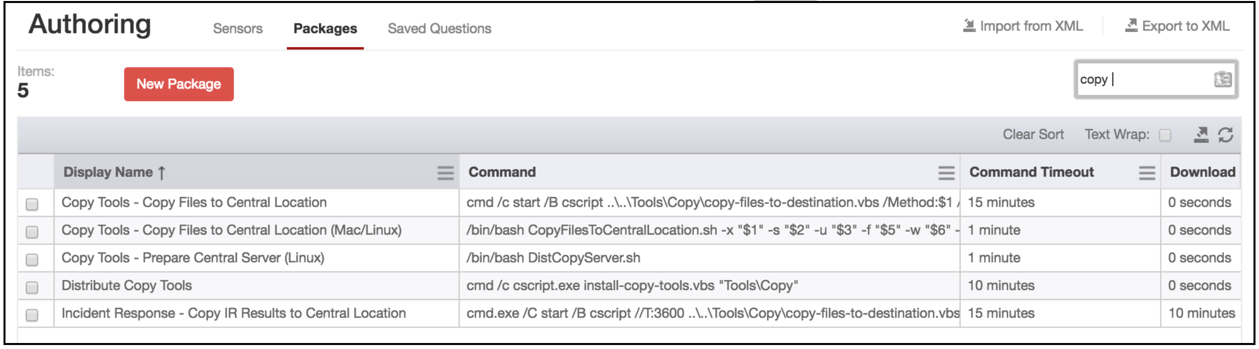
(Windows only) A `\\server\share` location, ideally a Distributed File System (DFS) location, that allows write access to the Domain Computers group. Do not enter user name and password information for the SMB transfer type.

Configure the Copy Tools packages

You can customize settings in the **Copy Tools - Copy Files to Central Location** and **Copy Tools - Copy Files to Central Location (Mac/Linux)** packages that are applied any time that package is deployed as an action.

Open the package to edit

From the Main menu, click **Content > Packages**. Type `copy` in the search box.



	Display Name ↑	Command	Command Timeout	Download
□	Copy Tools - Copy Files to Central Location	cmd /c start /B cscript ..\..\Tools\Copy\copy-files-to-destination.vbs /Method:$1 /	15 minutes	0 seconds
□	Copy Tools - Copy Files to Central Location (Mac/Linux)	/bin/bash CopyFilesToCentralLocation.sh -x "$1" -s "$2" -u "$3" -f "$5" -w "$6" -	1 minute	0 seconds
□	Copy Tools - Prepare Central Server (Linux)	/bin/bash DistCopyServer.sh	1 minute	0 seconds
□	Distribute Copy Tools	cmd /c cscript.exe install-copy-tools.vbs "Tools\Copy"	10 minutes	0 seconds
□	Incident Response - Copy IR Results to Central Location	cmd.exe /C start /B cscript //T:3600 ..\..\Tools\Copy\copy-files-to-destination.vbs	15 minutes	10 minutes

Select the package that you want to update and click **Edit**.

Update package timeouts

When you deploy a Copy Tools package as an action on endpoints, the minimum expiration time for the action is the sum of the **Command Timeout** and **Download Timeout** values. You can change the default values to increase or decrease the timeout when you deploy the action.

Field	Description
Command Timeout	The interval of time, in minutes, before the package command expires. By default, the command times out after 15 minutes.
Download Timeout	The interval of time, in minutes, before the download operation times out. By default, the download operation times out after 10 minutes.
Ignore Action Lock	Enable locked clients to run actions that include this package. For more information about the Action Lock setting, see Tanium Knowledge Base: Action Lock.

Save the package

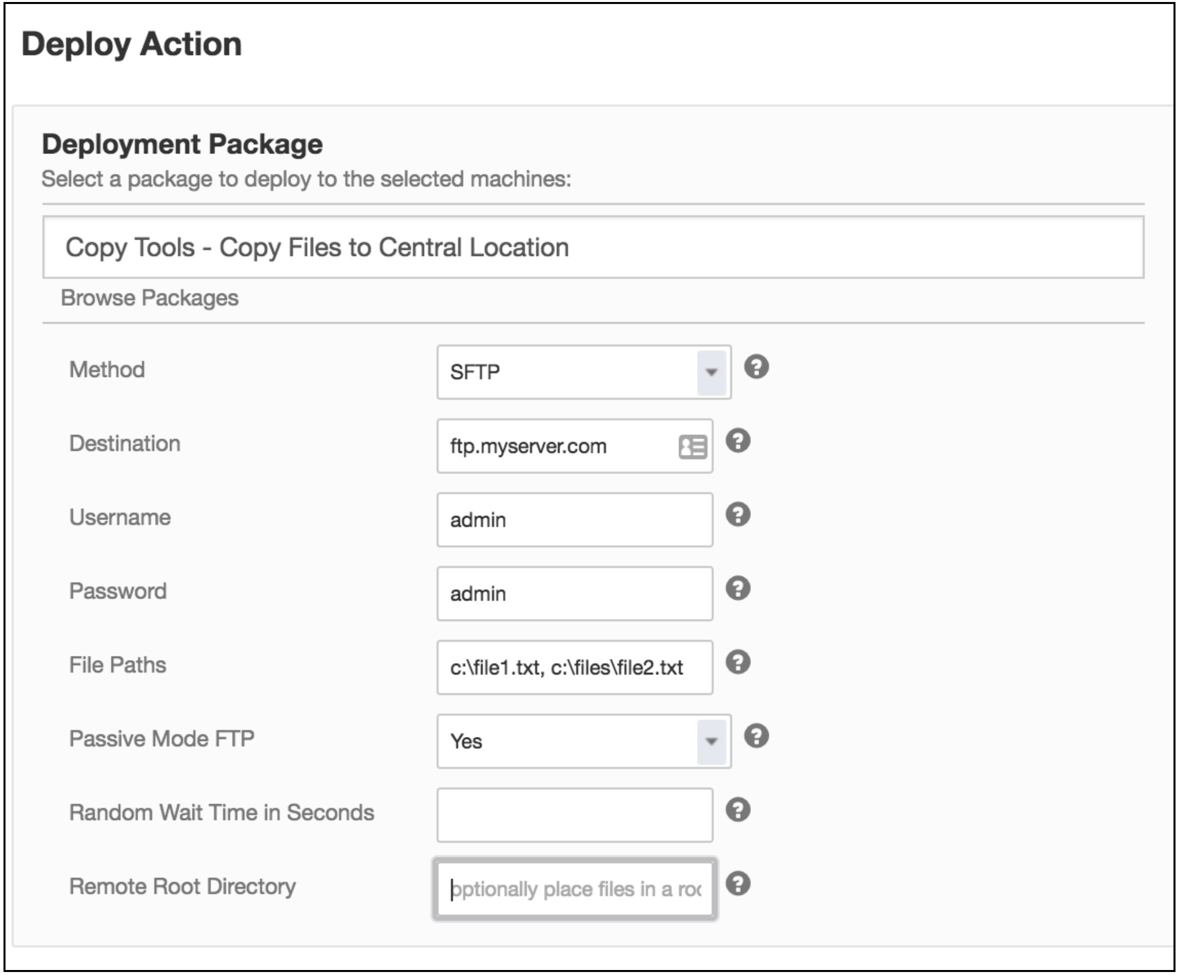
After you configure other settings and parameters, click **Save**.

Target endpoints

To target endpoints, you can ask a question, then drill down and deploy an action to a set of endpoints. When you are targeting endpoints, be careful not to overload the copy location. Verify that the count field in the results for your endpoint targeting is not too high. For more information about targeting endpoints, see Tanium Interact User Guide: Using Deploy action.

Copy with the general purpose action

1. Use the **Copy Tools - Copy Files to Central Location** and **Copy Tools - Copy Files to Central Location (Mac/Linux)** actions for general purpose copy operations of a comma-separated list of files to be copied from the specified endpoints.



Deploy Action

Deployment Package

Select a package to deploy to the selected machines:

Copy Tools - Copy Files to Central Location

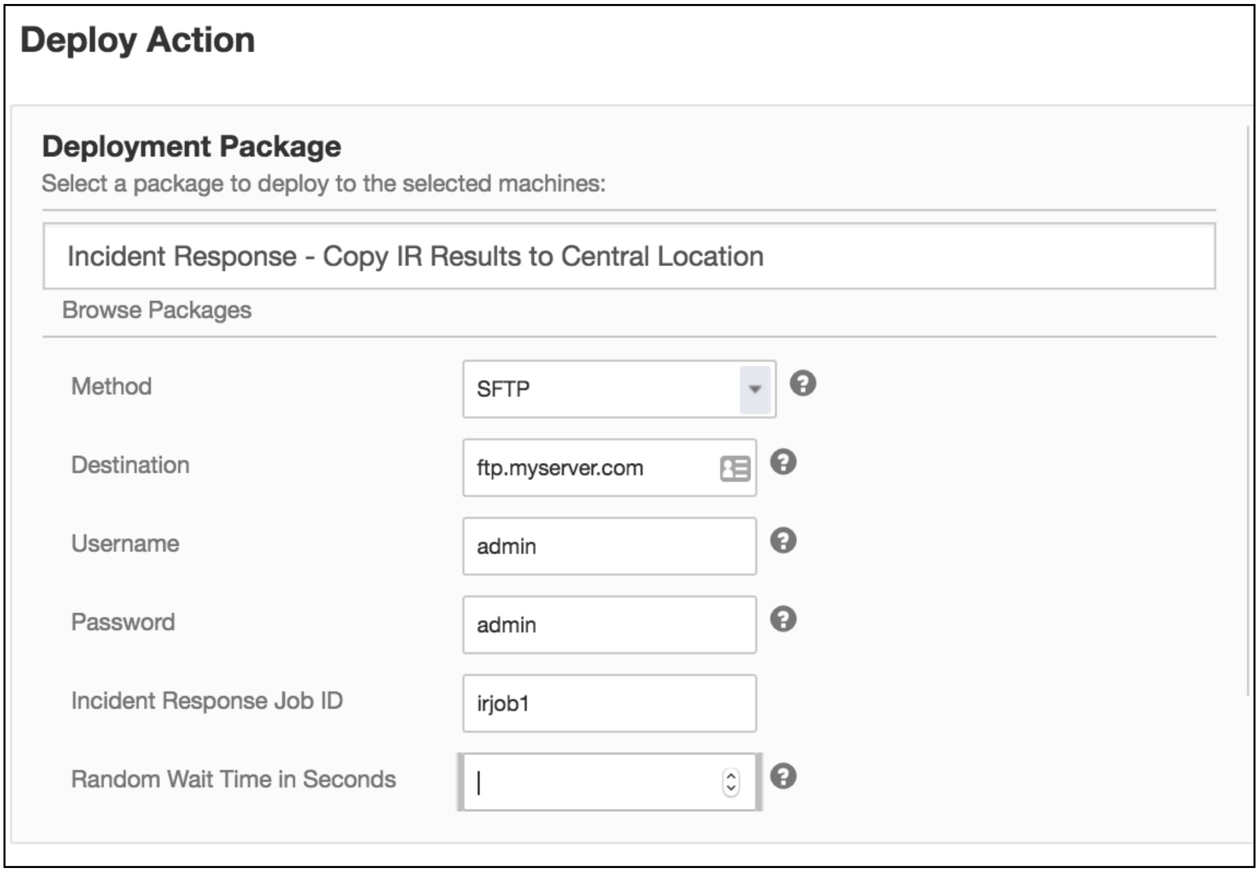
Browse Packages

Method	SFTP
Destination	ftp.myserver.com
Username	admin
Password	admin
File Paths	c:\file1.txt, c:\files\file2.txt
Passive Mode FTP	Yes
Random Wait Time in Seconds	
Remote Root Directory	optionally place files in a roo

2. Choose the transfer method, and specify the server and login information for your copy location. If you are using the SMB transfer method, do not enter user name or password information.
3. In the **File Paths** field, indicate a comma-separated list of absolute paths to files that you want to copy from each endpoint.
4. Choose a setting for how often the copy operation runs to prevent the copy destination from being overloaded. Choose from one of the following settings:
    - **Random Wait Time in Seconds** field: Enter the maximum number of seconds to wait before sending the files. The actual time when the endpoint runs the action is a random wait time between zero (no delay) and the specified count of seconds. For best results, as the number of endpoints in the security network increase, increase the maximum number of seconds that are specified.
    - **Schedule Deployment** section: Use the **Distribute over time** option to randomize the package copy process to smooth the distribution and avoid spikes in traffic.
5. Click **Show Preview to Continue** to preview the targeted endpoints on which you are deploying the action.
6. Click **Deploy Action**.

Copy by IR Job ID

For file copy operations that are required after an incident, use the **Incident Response - Copy IR Results to Central Location** action.



Deploy Action

Deployment Package

Select a package to deploy to the selected machines:

Incident Response - Copy IR Results to Central Location

Method	SFTP
Destination	ftp.myserver.com
Username	admin
Password	admin
Incident Response Job ID	irjob1
Random Wait Time in Seconds	

This action requires an IR job ID. Use the IR job ID that was created during the deployment of any IR packages that require the use of an IR job ID. For more information about IR job IDs, see About deploying parameterized sensors as actions on page 27.

File copy results

Both actions copy the specified files to a directory in the copy destination, for example: `<remote_root_directory>/YYYY-MM-DD-hh-mm-ComputerName/<file_paths>`. *<remote_root_directory>* is the value of the **Remote Root Directory** field and *<file_paths>* are the files that you specified in the **File Paths** field. The copied files retain the original directory structure.

Indexing file systems

Tanium Index 2.5.4.0001

Use Tanium™ Index to index the local file systems on Tanium Client endpoints that are running Windows, Linux, and macOS operating systems. Index is optimized to minimize endpoint resource utilization and work with journaling file systems, when available. The solution indexes local file systems, computes file hashes, and gathers file attributes and magic numbers. This information is recorded in an SQLite database for detection and reporting of threat indicators for files at rest.

Overview

Index creates and maintains an inventory of the file system on an individual endpoint with the following steps:

1. Perform initial inventory on page 36
2. Detect file system changes on page 37
3. Compute file hashes on page 37
4. Calculate magic number on page 38

Perform initial inventory

The file system inventory is saved in the SQLite database on the endpoint.

Windows

On Windows endpoints, Index uses the Master File Table (MFT) for the initial inventory, and only indexes local fixed drives.

Linux

On Linux endpoints, a platform-independent method is used to index the file system on the endpoint.

macOS

On macOS endpoints, a platform-independent method is used to index the file system on the endpoint.

Note: Index does not cross NFS mounts.

Detect file system changes

Any new file system changes are captured in the database.

Windows

On Windows systems, after the initial indexing is complete, changes are detected within a few seconds of when the change occurs if the drive is in journaled mode. If the drive is not journaled, Index operates in platform-independent mode, and changes are detected during the next indexing pass.

Index starts an asynchronous thread that checks the USN journal for changes, and updates the inventory in the database. If a file is modified, any existing saved hashes for that file are removed. When a file creation or modification is detected, the file is indexed to include the file name, file size, file creation time, file modification time, and directory name.

For files that have one or more NTFS hard links, Index records all hard links by the associated file reference number. This ensures that even if a hard link is removed, the remaining hard links are still in the database.

After initial indexing, Index does not detect changes made to only the attributes of a file, such as creation or modification timestamps. If the contents of a file are modified, Index records the new file modification time stamp, but does not update the file creation time stamp.

Linux and macOS

If the Tanium Recorder is deployed and operational on the endpoint, Index gets file change events from the Recorder usually within a minute of the change.
If the Recorder is not available, Index uses the platform-independent indexing method. With the platform-independent indexing, changes take longer to pick up because Index gets file changes by traversing the directory tree.

Compute file hashes

After the initial inventory of the file system is complete, Index computes and stores the hashes of files in the database. The Index hashing thread sleeps for the configured rescan interval. This interval is one hour by default. When the thread wakes up, it calculates hash values for files in the database that do not have hashes.

Index can record any combination of three different hash types: MD5, SHA-1, or SHA-256. You can disable calculation of hashes if desired.

Calculate magic number

The magic number is the first 4 bytes of the file. You can use the magic number to identify for many types of files. Magic numbers are recorded during the hashing pass for files that do not have a magic number entry.

Client system requirements

Operating system

- **Windows Workstation**: Windows XP with SP3 and later
- **Windows Server**: Windows Server 2003 with SP2 and later
- **Linux:** See Tanium Client Deployment Guide: Host system requirements (with Index 2.0.0 and later)
- **Mac OS X**: Mac OS X 10.8 and later (with Index 1.1.1.2 and later)

Disk space

To install Index normally, a minimum of 1 GB of free space must be available on the drive where Tanium Client is installed.

How much space the Index installation uses varies depending on how much space is used on the local disks that are being indexed. The actual space that is required for the Index database is proportional to the number of files and directories on the local disks and what hashes are configured. For a rough estimate, the Index database uses approximately 1 MB of space for each 1 GB of drive space that is used.

For more information about calculating the amount of space that is required for the Index database, ask your TAM.

CPU usage

Index monitors the CPU usage on the endpoint and throttles if needed . Index does this by measuring the amount of resources it is using and adjusting length of time between operations. It is normal to see Index use a bit more than it is configured limit from time to time.

The CPU usage is set via the `CPUUsageLimit` parameter in the Index configuration file. We suggest setting it to 5-7%.

Before you begin

- The endpoints must have Tanium Client installed. For more information, see Tanium Client Deployment Guide.

- Install the Tanium Index solution. For more information, see Install Index on page 21. If you are upgrading, see Upgrading Incident Response on page 23.
- (Linux endpoints) For live file event monitoring with Trace, the endpoint packages for Trace must have file event recording enabled. Check for the following sensor results:
  - Run the **Tanium Trace Status** sensor and verify that it returns `No issues found`.
  - Run the **Tanium Trace Endpoint Filters** sensor and verify that no file events are listed.
- Exempt the following process from antivirus or other host-based security solutions: `<Tanium Client>\Tools\EPI\TaniumEndpointIndex.exe`. For more information about AV exclusions for Tanium, see Tanium Support Knowledge Base: Security Software Exceptions (login required).

Deploy Index tools to endpoints

Deploy the latest Tanium endpoint Index tools to the appropriate endpoints with a scheduled action. You must target the endpoints by operating system. One way to target by operating system is to create OS-specific computer groups.

The actions that deploy Index to the endpoints are disabled by default.

1. From the Main menu, click **Actions > Scheduled Actions**.
2. Select the appropriate tool deployment action and click **Edit**.
   - **Deploy Distribute Tanium Endpoint Index Tools**
   - **Deploy Distribute Tanium Endpoint Index Tools for Linux**
   - **Deploy Distribute Tanium Endpoint Index Tools for Mac**
3. Specify the scheduling details and target systems for the endpoint package distribution.

   Selecting a **Reissue** interval ensures that endpoints that come online later get the Index tools.
4. Choose an action group of endpoints for the package.
5. Click **Show Preview to Continue**. Review the list of targeted endpoints and adjust the action group if necessary.
6. Click **Save Action**.

The action runs at the specified time or interval to distribute the **Distribute Tanium Endpoint Index Tools**, **Distribute Tanium Endpoint Index Tools For Linux**, and **Distribute Tanium Endpoint Index Tools For Mac** packages to the targeted endpoints.

The tools are deployed by default to the `<Tanium Client>\Tools\EPI` directory. An SQLite 3 database is used to store file indexes and the associated file hashes in the following location: `<Tanium Client>\Tools\EPI\EndpointIndex.db`.

Verify deployment on endpoints

Verify that Index is installed on endpoints and confirm that the Index tools are up to date.

The **Index Has Latest Tools** sensor returns the version of Index that is installed on the endpoint and whether the tools are up to date. This sensor returns two pieces of data:

Version number

> The version of Index that is on the endpoint.

Package required

> Specifies a number of Linux, Mac, or Windows endpoints that have a version of Index deployed on the endpoint that does not match the version of the solution that is imported on the server. You need to deploy the Index package to those endpoints.

Customize Index endpoint settings

Customize Index configuration settings with the **Distribute Tanium Endpoint Index Config**, **Distribute Tanium Endpoint Index Config For Linux**, or **Distribute Tanium Endpoint Index Config For Mac** packages. The default packages contain a sample configuration file to use as a template to customize the Index settings to your environment.

1. From the Main menu, go to **Content > Packages**.
2. Select the appropriate package and click **Edit**.
3. Add your `config.ini` file.
    a. To download the `sample_config.ini`, click Download ⬇.
    b. Update the file to the settings that you want and save it as `config.ini`. The following table provides a description of the settings in the config.ini files for Index and the operating systems that they apply to:

Setting	Windows	Linux	Mac
CPUUsageLimit Indexing pauses if the calculated CPU usage exceeds the configured CPU usage limit value during the file system inventory and computation of hashes.	✓	✓	✓
CPUViolationThreshold A percentage of CPU use that alerts a CPU use violation.	✗	✗	✓*
CPUViolationInterval The duration in seconds for a CPU use violation to be tolerated before triggering a violation.	✗	✗	✓*
StopAfterCPUViolations The number of reported CPU use violations to allow before stopping index. Setting this value to 0 disables the automatic stopping of indexing.	✗	✗	✓*
TakeSampleOnCPUViolation Captures a full stack trace when index detects a resource use violation.	✗	✗	✓*
HashesToCalculate specifies which hash types to calculate and store for each file hashed. Valid values are MD5, SHA1, SHA256, and none. To disable the calculation of hashes, set HashesToCalculate=None.	✓	✓	✓
MaxFileSizeToHashMB Limits hashing to only files whose size is less than specified. This is important because hashing is resource intensive for large files, for example, VMDK and other virtualization resources.	✓	✓	✓
FileIndexesPerThrottle The number of files to index before checking if throttling is necessary. Throttling ensures that the overall CPU utilization averages out to the defined CPU usage limit.	✓	✓	✓
FileHashUpdatesPerThrottle The number of updates to the hash digest to perform before checking if throttling is necessary.	✓	✓	✓

Setting	Windows	Linux	Mac
RescanInterval When the hashing phase is complete, this is the length of time that Index monitors for file changes before going to the next indexing phase.	✓	✓	✓
ScanFilePermissions List the permissions that are associated with files when an index is performed.	✓*	✓*	✓*
TrackChanges Enables real time file change notifications.	✓	✓	✗
logging.loggers.root.level The level of logging to apply to index operations.	✓	✓	✓
FilesystemTypesToExclude Exclude local (and otherwise indexable) file systems from indexing based on file system type. Provide a comma-separated list of file system types.	✗	✓*	✗
RecorderDBRescanInterval Sets the frequency of reading pending file change events from the Event Recorder.	✗	✓*	✓*
RecorderDBEventsPerRead Sets the maximum number of file change events to read from the Event Recorder in each batch.	✗	✓*	✓*
RecorderDirectory Specifies a non-standard Event Recorder installation location.	✗	✓*	✓*
ExcludeFromIndexing Create exclusions to keep specific files and paths out of file system indexes.	✓*	✓*	✓*
ExcludeFromHashing Create exclusions from hashing to exclude specific files and paths from having hash values calculated.	✓*	✓*	✓*

* = The setting is not enabled by default.

c. Click **Add** to upload the customized `config.ini` file.

4. Click **Save**.

5. Deploy the **Distribute Tanium Endpoint Index Config**, **Distribute Tanium Endpoint Index Config For Linux**, or **Distribute Tanium Endpoint Index Config For**

Mac packages.

   a. Click **Actions > Scheduled Actions**.
   b. Select the package that you want to deploy and click **Edit**.
   c. Edit the deployment details and target the package distribution to a specific action group.

By default, the config.ini file is located in the `<Tanium Client>\Tools\EPI` directory.

To ensure that updates to the modified `config.ini` file are preserved when you upgrade Index, see Upgrade Tanium Index on page 24.

Start indexing

To start indexing on the endpoints that have Index tools installed, use the **Deploy Start Indexing**, **Deploy Start Indexing For Linux**, and **Deploy Start Indexing For Mac** saved actions. To ensure that indexing gets restarted when a computer restarts, configure the saved action as a scheduled action. For example, you might schedule **Deploy Start Indexing** to run every 30 minutes.

For more information about these actions, see:

- Tanium Knowledge Base Index Reference: Start indexing
- Tanium Knowledge Base Index Reference: Start indexing For Linux
- Tanium Knowledge Base Index Reference: Start indexing For Mac

Check Index status

To check indexing status, use the **Index Status** sensor. For more information about the status values, see Tanium Knowledge Base Index Reference: Index Status.

Query indexed files

Use the **Index Query File** sensors to get details about files that have been indexed.

The **Index Query File Details** sensors return **Created** and **Last Modified** time stamps. The time stamps in the results make the strings that are returned for each file unique. To reduce the overall number of strings, use the following workflow:

1. Start with one of the following sensors that are less likely to return as many unique strings:
   - **Index Query File Path Using Name**
   - **Index Query File Path and Hash**

    - **Index Query File Exists**
    - **Index Query File Hash Recently Changed**
    - **Index Query File Count**
    - **Index Query File Permissions***
2. After getting results from the sensors above, you can drill down to get more details with the following sensors:
    - **Index Query File Details**
    - **Index Query File Details Using Name**
    - **Index Query File Details by Last Modified**
    - **Index Query File Details Using Name Sort By Largest**
    - **Index Query File Permissions***

* = To use the **Index Query File Permissions** sensor, the `ScanFilePermissions` setting in the Index config.ini must be enabled and set to `true`.

For more information about these sensors, see Tanium Knowledge Base Index Reference: Sensors.

Find files in a blacklist

You can provide a blacklist of hashes and compare that list with the hashes that are computed by Tanium Index. You can use MD5, SHA1, or SHA256 hashes. Index is not able to find blacklist hashes for files excluded from hashing by name or file size.

1. Edit the package.
    a. In the Tanium Console, go to **Content > Packages**.
    b. Select the **Distribute Index Query Blacklist**, **Distribute Index Query Blacklist For Linux**, or **Distribute Index Query Blacklist For Mac** package.
    c. Click **Edit**.
2. Update the `blacklist.txt` file.
   The file contains a list of hashes that are separated by commas or carriage returns. If the hashes are separated by commas, group the hashes of the same type together.

   > **Tip:** The blacklist has been successfully tested with over 100,000 entries, but start with a smaller number of hashes and update the blacklist on a regular basis.

    a. To download the current file, click Download ⤓.
    b. Remove the file that is currently in the package 🗑.

   c. Edit the `blacklist.txt` file.
   d. Click **Add** to upload the updated `blacklist.txt` file.
3. Click **Save**.
4. In the Tanium Console, use an operating system-based question to locate computers on which to deploy the Package. Drill down to the endpoints and click **Deploy Action**. Choose the **Distribute Index Query Blacklist**, **Distribute Index Query Blacklist For Linux**, or **Distribute Index Query Blacklist For Mac** package.
5. Perform comparison of deployed blacklist with hashes computed by Index.
   Use the **Get Index Query Find Blacklist Matches** sensor. This sensor returns a list of the file paths and hashes that are listed in the blacklist.

Troubleshoot

Index not running

By default, the configuration deployment packages contain a sample configuration file: `sample_config.ini`. If you did not replace this file with a customized `config.ini` file, the **Index Status** sensor returns: `Missing Config File`. Verify that you have replaced the `sample_config.ini` file with a customized `config.ini` file. See Customize Index endpoint settings on page 40.

Files and directory paths reported by Index are different compared to other methods

You might notice a difference in the files and directory paths that are reported by Index versus other methods. Windows uses hard links, symbolic links, and junctions for some of the files that the user sees. Enumerating the files with the Master File Table (MFT) shows the source path of the first hard link of a file, but not all of the hard links. As a result, scanning the MFT can yield different directory paths for files than a typical directory traversal.

Links in the MFT can cause problems with finding the full path of file. If you search for a file that seems to be "missing" from the System32 directory, you might find the file in a different location, such as the `C:\Windows\SysWow64` directory. Another example, the `C:\Users\all users` directory, is symbolically linked to the `C:\ProgramData` directory. Index follows the link and records the files in the database under `C:\ProgramData`. The hashes of these files are correct and match the linked files that are in the directories that are visible to users.

The differences caused by links in the MFT are rarely an issue for forensic analysis. Indicators of Compromise (IOC) rarely have a full path as an indicator item, and instead use a file name and MD5 hash. The blacklists of files that you can get from the government also include MD5 hashes.

Beginning with Index version 1.7.0, hard links in Windows can now be tracked. For more information, see Hard links not recorded.

Hard links not recorded

If you see that Index records only the first hard link for a file, and not the other hard link peers, you can verify which machines have the configuration to record all hard links. Use Interact to ask `Get Index Status from all machines`. If you see the message "Delete Index Database To Enable Hard Link Tracking", this means that these endpoints have a version of Index that can track hard links, but Index needs to reindex the filesystem to get all of the information needed to track the hard links in the filesystem. To enable hard link tracking, deploy the **Delete Tanium Endpoint Index database** package and initiate a reindex.

Performing reindexing message

If Index is started after not running for a while, either because it was stopped or the endpoint has been offline, you might see an entry in the `TaniumEndpointIndex.log` file that is similar to the following message:

```
[2019-04-11 11:40:28 GMT] [Information] [MFTScanner C:] [3932] 17635450400
not found in journal. Performing reindexing...
```

This message indicates that when Index restarted, the USN Journal no longer had the last Update Sequence Number (USN), so indexing restarted.

To reduce the likliehood of reindexing, use the **Deploy Start Indexing** scheduled action to restart Index every 30 minutes if indexing is disabled. Using this scheduled action:

- Catches new endpoints.
- Catches endpoints that are coming online after a restart.
- Starts indexing on these new and restarted endpoints.

Missing hash or magic number for file

Some files might be in the Index database with no hashes or magic number. This situation can happen for the following reasons:

- The file is inventoried, but the initial hashing pass is not complete.
- The file is changed, but the `RescanInterval` timer has not initiated the file to be rehashed.
- The file is locked, so Index cannot get read lock to hash it.
- The file is larger than the configured `MaxFileSizeToHashMB` value.

- The file was excluded from hashing with the `ExcludeFromHashing` regular expression.

Reference: Log settings

Log level

The levels for the `logging.loggers.root.level` property in the `config.ini` file are in the following list. The levels are listed from least to most verbose:

- none (turns off logging)
- fatal
- critical
- error
- warning
- notice
- information (default)
- debug
- trace (includes the most messages)

If you set the level to `debug` or `trace`, expect verbose output in the log file. Most of the information in these levels is meant for debugging by Tanium technical teams. Do not change logging level - even lower - unless directed by your TAM.

Log file rotation

Log files are capped at 10 MB. When the file reaches 10 MB, the file gets compressed and moved to `TaniumEndpointIndex.log.timestamp.gz`. Index keeps up to ten log files, removing older log files. The log files are in the `<Tanium Client>\Tools\EPI` directory.

Dump (.dmp) files

`TaniumEndpointIndex_[0-9].dmp` log files are created if the Index process crashes. Files rotate, with `TaniumEndpointIndex_0.dmp` always being the most recent.

Reference: Index sensors and packages

For details about the parameters for each Index sensor and package, see [Tanium Knowledge Base: Index Reference]().

Collecting data with Live Response

Tanium Live Response 1.1.2

A critical step in the incident response process is the collection of data from compromised endpoints for further forensic analysis. With Live Response, you can collect extensive data from endpoints.

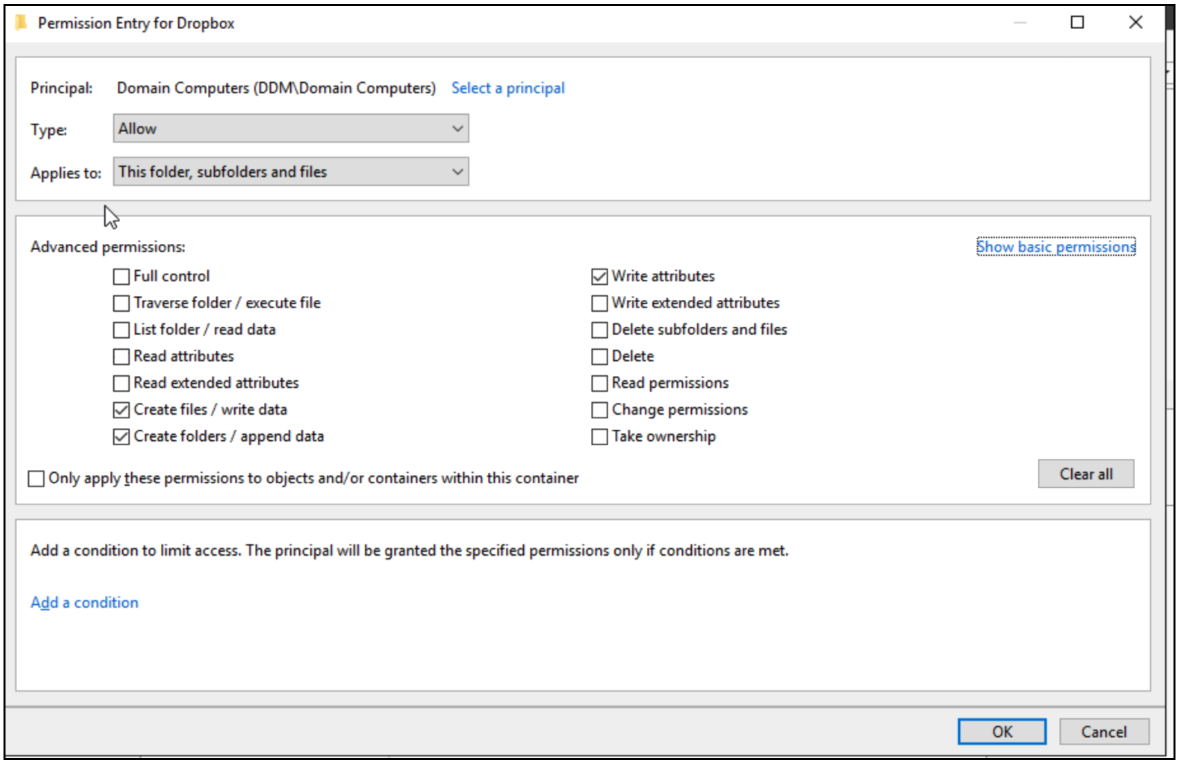
The Live Response package collects forensic information from endpoints, and transfers the results to the specified location. The **Live Response** packages contain configuration files that specify what data to collect, and where to copy the data.

Before you begin

- Installing Incident Response solutions on page 20.
- If you are upgrading, see Preserve configuration files before upgrading Live Response on page 25.

Configure a copy location and endpoints

- You must have a copy location to save the forensic data to be collected. The server that receives information from Live Response can be an Amazon S3 Bucket, or able to communicate over SFTP, SCP, or SMB (Windows only - SMB destinations are not included in Live Response packages for macOS and Linux.) protocols. A best practice is to use Linux SFTP/SCP destinations for SCP/SFTP transfers.
- For an SMB copy location, the endpoint and user must have permissions to mount the server as a share and write to the directory. For authenticated domain computers, configure the destination directory as a write-only share.



Permission Entry for Dropbox

Principal: Domain Computers (DDM\Domain Computers) Select a principal

Type: Allow

Applies to: This folder, subfolders and files

Advanced permissions:

- □ Full control
- □ Traverse folder / execute file
- □ List folder / read data
- □ Read attributes
- □ Read extended attributes
- ☑ Create files / write data
- ☑ Create folders / append data
- ☑ Write attributes
- □ Write extended attributes
- □ Delete subfolders and files
- □ Delete
- □ Read permissions
- □ Change permissions
- □ Take ownership

□ Only apply these permissions to objects and/or containers within this container

Add a condition to limit access. The principal will be granted the specified permissions only if conditions are met.

Required advanced permissions :

- Read attributes
- Create files / write data
- Create folders / append data
- Write attributes

- For Amazon S3 Bucket copy locations, ensure that clients are synchronized with a time server. Transfers fail if the client time differs from the server time by more than 15 minutes.

  For more information on using Amazon S3 Buckets with Live Response, see How to create an AWS S3 Bucket for use with Live Response (login required).

- Do not use SMB transfer destinations when a system has been quarantined by Tanium. Live Response uses domain authentication for transfers. When a system is quarantined it cannot reauthorize with the domain and authentication fails.

Configure the Live Response package

Before you deploy the **Live Response** package, customize the transfer and collector configurations.

You can upload multiple JSON files to the package with different configurations. Select the appropriate configuration when you deploy the package.

IMPORTANT: Custom configuration files are not saved when you upgrade Live Response. For more information, see Preserve configuration files before upgrading Live Response on page 25.

Edit the Live Response package

1. Open the package to edit.
   a. From the Main menu, click **Content > Packages**.
   b. In the search box, type `live response`.
   c. Select a **Live Response** package and click **Edit**.
2. (Optional) Update the package timeouts.
   When you deploy the Live Response package as an action on endpoints, the minimum expiration time for the action is the sum of the **Command Timeout** and **Download Timeout** values. You can change the default values to increase or decrease the timeout when you deploy the action.
   These timeouts affect only the transfer of the Live Response package to the endpoints. Live Response runs in detached mode, so file transfers are not associated with the completion of the action.

Update the transfer configuration files

Collected files are sent to network destinations that you specify in transfer configuration files.

1. Add information about a transfer destination.
   The Files section of the package contains a sample transfer configuration file (`SMB.json`, `SCP.json`, `SFTP.json`, `S3.json`) for each supported transfer method. Download the files for the destinations that you want to configure, and update the contents to specify the details about your transfer destination. For more information on the configuration file format, see Reference: Transfer configuration on page 52.
2. Add files that are required to verify the identity of the destination, such as:
   - A known_hosts file for SSH-based transfer mechanisms, such as SCP or SFTP
   - RSA files, if you are using an RSA key
   - S3 secret key file, if you are using an Amazon S3 Bucket

(Optional) Update collector configuration

The collector configuration controls the data that gets collected. Choose from one of the following configurations when you deploy the **Live Response** package:

- `Standard_Collection.json`: Use for default data.
- `Extended_Collection.json`: Use to collect the same data as `Standard_Collection.json`, plus more file based artifacts, such as the kernel, the Master File Table, USN Journal, event logs, registry hive files, and so on.
- `Memory_Collection.json`: Use for memory acquisition.
- `Custom_Collection.json`: Use as an example if you are adding your own PowerShell scripts to Live Response.

For more information about what gets collected for each file, see Default data modules on page 61.

(Optional) Set default values

In the **Parameters** section, select a parameter. You can choose default values that are selected when you deploy the package.

Collect data from endpoints

To collect data from endpoints, deploy the Live Response package.

> **IMPORTANT:** To prevent resource overload on endpoints, only issue this action manually. Do not create a scheduled action.

1. Target endpoints for data collection. Use an operating system-based question, for example: `Get Computer Name from machines with Is Windows containing "True"`.
2. Select the endpoints from which you want to collect data and click **Deploy Action**.
3. In the **Deployment Package** field, type `Live Response - Windows`.
4. Define the collector and transfer configurations.
5. In the **Base Directory** field, provide a directory name where files are placed as they are collected. This directory is created under the Remote Path value that you provide in the destination you are using for the Live Response package. For example, if you provide a Base Directory of `MyCollection` for an SSH destination where the Remote Path is FileCollection, the result would be `/home/`*`username`*`/FileCollection/MyCollection` since the remote path provided in SSH destinations is relative to the home directory of the present user. Depending on the type of destination, the location of the Remote Path can vary. For example, in SMB destinations it is explicit; whereas in SSH destinations it is relative to the home directory of the present user.

6. Optionally select **Flatten Output Files** if you want all collected files placed in one directory where the filename includes the original path, but does not retain the folder structure.
7. Click **Show Preview to Continue**.
8. After you preview the list of endpoints to which the action is being deployed, click **Deploy Action**.

Live Response tests the connection by writing a `LRConnectionTest`file to the destination. If the write fails, the action tries the other destinations in the transfer configuration in the order they are listed in the configuration file. If all the connection tests fail, the Live Response action does not proceed.

The Tanium Server shows the package as complete almost immediately after the package is downloaded on the endpoints. This completion is not accurate because Live Response runs in detached mode. File transfers continue after the action completes.

The actual time to complete the transfer depends on the endpoint activity and connection speed between the endpoint and the destination system.

Collect logs

In addition to the standard action logs on the endpoint (`<Tanium Client>\Downloads\Action_###\Action_####.log`), a log file of Live Response activities included in the same directory. This file follows the naming convention: `YYYYMMDDhhmm_LR.log`.

When Live Response completes, the `YYYYMMDDhhmm_LR.log` is copied to the destination. The action log is not copied to the destination.

Use both the action log and the Live Response log file to troubleshoot problems. The action log captures messages written to standard error (stderr).

Reference: Transfer configuration

Live Response includes the following example configuration files for file transfer:

- `S3.json`
- `SCP.json`
- `SFTP.json`
- `SMB.json`

All transfer configuration files must contain one connection string.

```
{
        "dest":[
                "scp://<...>"
        ]
}
```

IMPORTANT: The password field for SCP and SFTP in the configuration files support URL encoding. For example, replace `#` with `%23`. Other special characters in URLs include:

- space - `%20`
- & - `%26`
- # - `%23`
- ? - `%3F`
- : - `%3A`
- = - `%3D`
- @ - `%40`
- % - `%25`

View supported protocols and options for file transfers

To see all supported protocols and protocol-specific options, you can run the `taniumfiletransfer.exe` file. The Live Response package contains the `taniumfiletransfer_32.exe` and `taniumfiletransfer_64.exe` files. When the package is deployed, the file that is appropriate for the bitness of the endpoint is copied to the endpoint and renamed to `taniumfiletransfer.exe`.

Download the `taniumfiletransfer_32.exe` or `taniumfiletransfer_64.exe` file from the **Live Response - Windows** package.

To see a list of supported protocols, run one of the following commands, depending on the bitness you are using:

```
taniumfiletransfer_64 protocol
```

```
taniumfiletransfer_32 protocol
```

To see details about scp protocol, including options for the protocol connection string, run:

```
taniumfiletransfer_64 protocol <protocol>
```

S3 protocol file transfer parameters and example

Parameter	Description
connectTimeout	The amount of time to spend attempting to establish a connection. (Default: 30 seconds).
keyfile	The path to an S3 secret access key, stored in a text file to use for authentication.
region	An explicitly defined S3 region.
disableSSL	When set to 'true', SSL encryption is disabled. When set to 'false' (the default), SSL encryption is enabled. Note: Use SSL with Amazon AWS S3 services.
s3ForcePathStyle	Forces API calls to use path-style URLs (bucket name is part of the URL path) for accessing buckets when set to true. When false, the API can use path-style URLs or virtual-hosted-style URL's (bucket name is in the hostname). This setting may be necessary when attempting to send to non-AWS S3 services or when the bucket name cannot be used as part of hostname.

Example: `s3://SOMEKEY@my-compatible-s3-service.com:9000/mybucket/some/dir#keyfile=secretaccesskey.txt&s3ForcePathStyle=true&region=my-region`

Access key ID: `SOMEKEY`

endpoint: `https://my-compatible-s3-service.com:9000`

disableSSL: `false`

region: `my-region`

s3ForcePathStyle: `true`

bucket: `mybucket`

path: `some/dir`

keyfile: `secretaccesskey.txt`

connectTimeout: `30 seconds` (default)

SCP protocol file transfer parameters and example

Parameter	Description
connectTimeout	The amount of time to spend attempting to establish a connection. (Default: 30 seconds).
keyfile	The path to an SSH private keyfile to use for authentication.
knownHostsFile	Path to an SSH known hosts file. Default is `known_host`. Generate a `known_hosts` file using a command such as: `taniumfiletransfer_32.exe ssh-keyscan` *`ipaddress`* `> known_hosts`. If you use a non-standard port, the output in the `known_hosts` file needs to be edited to reference that port. Lines starting with a double-hash ## are the ones that need to be edited. For example, if you are using port 222, the line `## Host: 192.168.0.113:22 (192.168.0.113:22)...` needs to be updated to `## Host: 192.168.0.113:222 (192.168.0.113:222)....` If the `known_hosts` file is not edited to reference the correct port, Live Response encounters a failure. **Note:** The known-hosts file must be ASCII encoded.
insecureDisableHostKeyVerification	Setting this to `true` disables verification of the identity SSH hosts. The default is `false`. Disable Host Key Verification is not currently available with Live Response in Threat Response and should only be used when debugging connections.

Example: `scp://bob@my.hostname/dir#connectTimeout=5s&keyfile=id_rsa&insecureDisableHostKeyVerification=true`

username: `bob`

password: `<not supplied>`

hostname: `my.hostname`

port: `22 (default)`

path: `dir`

keyfile: `id_rsa`

host key verification: `DISABLED`

connectTimeout: `5 seconds`

SFTP protocol file transfer parameters and example

Parameter	Description
connectTimeout	The amount of time to spend attempting to establish a connection. (Default: 30 seconds).
keyfile	The path to an SSH private keyfile to use for authentication.
knownHostsFile	Path to an SSH known hosts file. Default is `known_host`. **Note:** The known-hosts file must be ASCII encoded.
insecureDisableHostKeyVerification	Setting this to `true` disables verification of the identity SSH hosts. The default is `false`. Disable Host Key Verification is not currently available with Live Response in Threat Response and should only be used when debugging connections.

Example:

```
sftp://bob@my.hostname/dir#connectTimeout=5s&keyfile=id_
rsa&insecureDisableHostKeyVerification=true
```

username: `bob`

password: `<not supplied>`

hostname: `my.hostname`

port: `22` (default)

path: `dir`

keyfile: `id_rsa`

host key verification: `DISABLED`

connectTimeout: `5 seconds`

SMB protocol file transfer example

Example: `smb://my.smb.host/some/dir`

hostname: `my.smb.host`

path: `some/dir`

Reference: Collector configuration

You can customize the data and files that are collected when you deploy the Live Response package. It might be helpful to have multiple versions of the configuration file for specific types of endpoints, such as endpoints that would have antivirus or quarantine files.

Live response includes the following example configuration files:

- `Standard_Collection.json`: Use for default data.
- `Extended_Collection.json`: Use to collect the same data as `Standard_Collection.json`, plus more file based artifacts, such as the kernel, the Master File Table, USN Journal, event logs, registry hive files, and so on.
- `Memory_Collection.json`: Use for memory acquisition.
- `Custom_Collection.json`: Use as an example if you are adding your own PowerShell scripts to Live Response.

Global Settings

You can configure the base settings for the Live Response configuration files. A module or file collection setting can override these base settings by including the setting name and value in the appropriate section.

```
"options":{
               "disk_info": true,
               "copy": true,
               "depth": 0,
               "max_num_files": -1,
               "raw": false,
               "raw_fallback": false,
               "hashes": ["md5","sha256"],
            "log_level": "info"
    }
```

disk_info

Specifies whether to collect time stamp information from the MFT. (Default: `true`)

`true` Collect `Standard_Information` time stamps from the MFT. If raw file collection occurs, `File_Name` attribute time stamps are also collected from the MFT. Enable raw file collection by setting one of the following properties: `raw: "true"` or `raw_fallback: "true".`
`false`: No time stamp information is collected.

copy

Specifies whether to copy files to the destination. The default global option is `true`, with overrides set to `false` for the process details, module details, and driver details modules.

`true`: Copy all files to the destination as part of the Live Response process. This set of files includes everything related to processes, loaded modules (dlls), driver files, and so on. Copying files adds significant time, bandwidth and storage space requirements to the process. In general, target file collection to files of particular interest.
`false`: Files are not copied.

depth

Specifies the number of subdirectories in which the regular expression is evaluated. For an unlimited number, set to `-1`.

max_num_files

Specifies a maximum number of files to collect. For an unlimited number, set to `-1`.

raw

Specifies whether to use Windows API mode, or to parse the master file table (MFT). API mode is faster than parsing the MFT.
`false`: (default) Use API mode.
`true`: Parse the master file table.

raw_fallback

Specifies whether to parse the MFT if API calls are unsuccessful. Set to `true` to enable.

hashes

Specifies the type of hash to calculate for the files.
Valid values: md5, sha256. SHA1

log_level

Reserved for future use to control logging to the Live Response log file.

Scripts

You can configure your own PowerShell script (ps1 file) to run as part of the Live Response. For example, the following configuration enables a `collect-test-script.ps1` file to run. You must also upload the script file to the package, for example: `collect-test-script.ps1`.

```
"scripts":[{
        {
                "name": "collect-test-script",
                "filename": "collect-test-script.ps1",
                "safe_args": ["-i", "input_file.txt", "-o", "output_file.txt"],
                "enabled": true,
                "order": "01"
        }
]
```

name

Specifies the name of the script.

filename

Specifies the name of the ps1 file. The script file must be uploaded to the Live Response package.

safe_args

Specifies a list of parameters and example values for the script.

enabled

Specifies whether the script is run when the Live Response package is deployed.

order

Controls the order in which scripts run. This order is commonly referred to as the "order of volatility" in digital forensics and incident response fields. It is often advisable to collect that data which is most likely to change before collecting data that changes less frequently for example, collecting running process details, which may change more frequently than configuration files stored on disk.

Modules

The modules section contains the data collection functions included with Live Response. An example of a module definition follows:

```
"modules":[
       {
               "name": "ProcessDetails",
               "enabled": true,
               "copy": false,
               "order": "02"
       },
]
```

name

Specifies the name of the module.

enabled

Specifies whether the module is enabled during the deployment of the Live Response package.

order

Controls the order that data is collected. This order is commonly referred to as the "order of volatility" in digital forensics and incident response fields. Collect data that is most likely to change before collecting data that changes less frequently. For example, collect running process details first, then configuration files stored on disk.

DEFAULT DATA MODULES

The following data is captured by default, and is configurable in the `Standard_Collection.json` file:

- Process details
- Module details
- Driver details
- Prefetch
- Amcache
- Shim cache
- Scheduled tasks
- Recent files
- Network connections
- Process handle details
- Autoruns details
- Hosts file

EXTENDED DATA MODULES

The following data is configurable in the `Extended_Collection.json` file:

- Process details
- Module details
- Driver details
- Prefetch
- Amcache
- Shim cache
- Scheduled tasks
- Recent files
- Network connections
- Process handle details
- Autoruns details
- Hosts file
- Standard and Master Boot Record
- Master File Table
- USN Journal, Kernel
- Registry Hives
- User Profiles
- Event Logs
- Prefetch files
- Chrome user data
- Trace database (if present)
- Index Database (if present)

A description of the files that are collected by each module is provided in the Threat Response documentation. See File Collector Sets for more information.

Files

The Files section specifies which files to collect from the endpoints, along with the associated metadata.

```
"files":[
        {

                "name": "MFT",
                "path": "%systemdrive%",
                "regex": "(\\$MFT$)",
                "hashes": ["md5","sha1","sha256"],
                "enabled": true,
                "order": "01",
                "raw": true

        }
]
```

FILE PROPERTIES

name

Specifies a name that describes the group of files.

path

Specifies a file path on the endpoint.

regex

Specifies a regular expression that is run on the specified directory path. All files that match are gathered.

hashes

Specifies a list of hashes to collect for each file, can include md5, sha1, sha256.

enabled

Description

order

Controls order that files are gathered. Collect data that is likely to change before collecting data that changes less frequently.

depth

Specifies the number of subdirectories in which the regular expression is evaluated.

Regular expressions and environment variables

Paths and file patterns support regular expression syntax.

The File Pattern regular expression is applied to the file name only.

The following table provides some example patterns to show how Live Response uses both regular expressions and environment variables on Windows, Linux, and macOS endpoints.

Example Live Response task	Operating system	Path	File pattern	Explanation
Collect host file	Windows	`%systemdrive%\windows\system32\drivers\etc`	`^hosts$`	Windows applies regular expressions to file name.
	Linux/macOS	`/etc`	`hosts$`	In this example, `hosts` matches.
Collect Bash History of every user	Windows	Not Applicable	Not Applicable	Not Applicable
	Linux/macOS	`$HOME`	`/.bash_history$`	A file name that matches `.bash_history`.
Collect a file names findme.txt from platform root	Windows	`C:\`	`^findme.txt$`	The filename starts with `findme.txt`
	Linux/macOS	`/`	`^findme.txt$`	The filename starts with `/findme.txt`

Any environment variables that you use resolve as described in the following table.

Environment variable	Supported operating system	Corresponding value
%appdata%	Windows	*C:*\Users\username\appdata\roaming
%homepath%	Windows	\Users\username
%localappdata%	Windows	*C:*\Users\username\appdata\local
%psmodulepath%	Windows	*C:*\Users\username \documents\windowspowershell\modules
%temp%	Windows	*C:*\Users\username\appdata\local\temp
%tmp%	Windows	*C:*\Users\username\appdata\local\temp
%userprofile%	Windows	*C:*\Users\username
%taniumdir%	Windows	The Tanium Client directory. Defaults are: \Program Files\Tanium\Tanium Client\ (32-bit OS) \Program Files (x86)\Tanium\Tanium Client\ (64-bit OS)
$TANIUMDIR	Linux, Mac	The Tanium Client directory. Defaults are: /Library/Tanium/TaniumClient/ (macOS) /opt/Tanium/TaniumClient/ (Linux)
$HOME	Linux, Mac	All user home directories that do not have a home directory set to blacklisted shells, and match shells that are listed in /etc/shells file. If there is no /etc/shells file, all shells are allowed.

Environment variables that are local to the endpoint are supported. For example, if %SYSTEMROOT% is set on an endpoint to expand to C:\WINDOWS, you can use such a variable on a path.

Isolating endpoints

Tanium Quarantine 3.1.1.0012

With Tanium™ Quarantine you can isolate a Windows, Linux, or Mac endpoint that shows evidence of compromise or other suspicious activity. Use Quarantine to apply, remove, and test for quarantine.

When an endpoint is quarantined, only approved traffic is allowed on the quarantined endpoint. By default, this traffic is allowed only:

- Between the Tanium Client on the quarantined endpoint and Tanium Server over port 17472.
- For essential traffic that is necessary to obtain and resolve IP addresses (DHCP/DNS).

Quarantine includes a safety feature that automatically reverses a quarantine policy that was applied by the tool. After a quarantine policy is applied, the effect of the policy is logged. If the endpoint is able to communicate with Tanium Server, Quarantine logs the successful application of the policy. If a policy prevents the endpoint from communicating with Tanium Server, Quarantine backs out the policy and saves logs in the action folder.

Before you begin

> **IMPORTANT:** Test the quarantine policy in a lab environment before deploying the policy. Do not apply a policy until its behavior is known and predictable. Incorrectly configured policies can block access to the Tanium Server.

- Install the Tanium Quarantine solution. For more information, see Install Quarantine on page 22.

  > **Note:** You must first install the Incident Response solution before installing Quarantine.

- You must have a Content Administrator account for Tanium Console. For more information, see Tanium Core Platform User Guide: Managing Roles.
- Identify the traffic that is required when an endpoint is under quarantine.
- You must have a lab machine on your target platform (Windows, Linux, or Mac) on which you can test the quarantine policies. You must be able to physically access the machine or to access it using RDP (Windows) or SSH (Linux, Mac).

- You must have access to the endpoint that you want to quarantine through a sensor or saved question in the Tanium Console.

Endpoint operating system requirements

Supported Windows versions

- Windows 7
- Windows 8.1
- Windows 10
- Windows Server 2003
- Windows Server 2008
- Windows Server 2012
- Windows Server 2019
- Windows Server 2012R2
- Windows 2008R2

Supported Linux OS versions

- RedHat/CentOS 5 IPTables on SYSV
- RedHat/CentOS 6 IPTables on SYSV
- RedHat/CentOS 7 Firewalld on Systemd
- Ubuntu 12,14 UFW on Upstart
- Ubuntu 15 UFW on Upstart/Systemd

Supported Mac OS versions

- macOS 10.15 Catalina
- macOS 10.14 Mojave
- macOS 10.13 High Sierra
- macOS 10.12 Sierra
- OS X 10.11 El Capitan
- OS X 10.10 Yosemite
- OS X 10.9 Mavericks

OSX 10.8 - Mountain Lion and earlier releases are based on ipfirewall (IPFW) and are not supported.

Configure Windows endpoints

The **Apply Windows IPsec Quarantine** package uses Windows IPsec policies to quarantine the endpoint. You can also add custom rules and options, see Create custom quarantine rules on page 70 for more information.

> **Note:** You cannot use Windows IPsec Quarantine on networks where a domain IPsec policy is already enforced.

Check that the IPsec Policy Agent service is running on the endpoints

Optionally, you can verify that the IPsec Policy Agent is listed as a running service in Windows.

1. In Tanium™ Interact, ask the question: `Get Service Details containing "PolicyAgent" from all machines with Service Details containing "PolicyAgent"`
2. In the table that gets returned, check the results in the following columns.
   - **Service Status**: Running or Stopped
   - **Service Startup Mode**: Manual or Automatic
3. If necessary, drill down into the results to determine which endpoints do not have the IPsec Policy agent running.

(Windows XP only) Deploy quarantine tools

The Quarantine Tools Pack includes a Microsoft policy that IPsec Quarantine uses to quarantine endpoints that are running Microsoft Windows XP. The application of IPsec policy is native to versions of Microsoft Windows later than Microsoft Windows XP and they do not require the tool pack.

To find endpoints that require the quarantine tools pack:

1. From the Tanium Console, open the Quarantine dashboard.
2. Click **Needs Quarantine Tools Pack (XP only)**, and select the Windows XP-based endpoints that require the tool pack.
3. Select **Deploy Action**. The package wizard opens.
4. Select **Distribute Quarantine Tools**. The tool pack is deployed to the selected endpoints.

Configure Linux endpoints

The **Apply Linux IPTables Quarantine** package quarantines endpoints that are running Linux-based operating systems that support the use of the **iptables** module.

Verify that endpoints are not using Network Manager

Linux IPTables Quarantine checks to ensure that the **iptables** module is installed and disables the use of the **Network Manager** module on endpoints that are targeted for quarantine.

You can check for Linux-based endpoints that are running Network Manager by using the **Linux Network Manager** sensor to determine if Network Manager is enabled. In Interact, type `network manager` to find the sensor. This sensor has no parameters.

Configure Mac endpoints

The **Apply Mac PF Quarantine** package quarantines endpoints that are running Mac OS X operating systems that support the use of Packet Filter (PF) rules. This package creates packet filter rules that isolate endpoints by eliminating communication with network resources. Packet Filter (PF) software must be installed on endpoints that are targeted for quarantine.

Test quarantine on lab endpoints

By default, the quarantine on the lab endpoint blocks all communication except the Tanium Server. You can configure custom rules to define allowed traffic direction, allowed IP addresses, ports, and protocols. For more information about how to create and deploy custom rules, see Create custom quarantine rules on page 70.

> **IMPORTANT:** Do not quarantine without testing the rules configuration in the lab.

1. Target computers for quarantine.
   a. In Tanium Console, use the **Is Windows**,**Is Linux**, or **Is Mac** sensor to locate an endpoint to quarantine.
   b. Select the entry for **True**, and click **Drill Down**.
   c. On the saved questions page, select **Computer Name** and click **Load**.
      A Computer Names list displays the names of all computers that are running the selected OS.
   d. Select the lab endpoint as a target and click **Deploy Action**.
2. In the **Deployment Package** field, type the name of the quarantine package that you want to deploy:
   - `Apply Windows IPsec Quarantine`
   - `Apply Linux IPTables Quarantine`
   - `Apply Mac PF Quarantine`

3. (Optional) Define quarantine rules and options.
   For more information about quarantine rules, see Create custom quarantine rules on page 70.
   - If you already attached a `taniumquarantine.dat` file to the package you are deploying, you do not need to make any other configurations.
   - Otherwise, select **Override Config** to apply custom rules to the action.
   - If you are using the options and rules in this package deployment, select any options that you want to enable and enter your custom quarantine rules into the **Custom Quarantine Rules** field.

4. Click **Show Preview to Continue** to preview the targeting criteria for the action. Click **Deploy Action**.
5. Verify quarantine of the targeted lab endpoint.

   Confirm that the computer has no available means of communication to resources other than Tanium Server and any endpoints that you configured in custom quarantine rules.

   You can use RDP (Windows) SSH (Linux/Mac), the Ping network utility, or a similar means to confirm that communication is blocked. By default, the only traffic that the quarantine allows is between Tanium Client on the quarantined computer and Tanium Server over port 17472. If the computer is a server that must allow connections to name servers, verify that those connections are allowed to pass through.
6. Verify the visibility of the quarantined computer to Tanium Server.
   1. Target the lab computer with a question or sensor.
   2. Check the sensor results for the visibility of the quarantined computer.
   3. On the Quarantine dashboard, click **Isolated Machines**. A single computer is listed with a `Yes` on the **Quarantine: Isolated Machines** page.

Action folders are located under the Tanium Client installation folder on the endpoint, usually `<`*`Tanium Client`*`>\Downloads\Action_XXXX.log`.

Remove quarantine

Deploy the **Remove Windows IPsec Quarantine**, **Remove Mac PF Quarantine**, or **Remove Linux IPTables Quarantine** package to the endpoint to remove the quarantine from the computer. Use RDP (Windows), SSH (Mac/Linux), the Ping utility, or another method to confirm the removal of the quarantine and the normal communication of the test computer.

Create custom quarantine rules

Quarantine rules and options define allowed traffic direction, allowed IP addresses, ports, and protocols. All other traffic is blocked. These rules are in the same format for Windows, Linux and Mac. For custom quarantine rule syntax, see Reference: Custom rules and options on page 71.

If you do not define any quarantine rules, the default values are used, which gives the quarantined endpoint access only to the Tanium Server and permits DNS/DHCP traffic.

If you previously provided a Windows IPsec policy file in earlier versions of Quarantine, the IPsec policy overrides the custom quarantine rules.

> **IMPORTANT:** Test the quarantine policy in a lab environment before deploying the policy. Do not apply a policy until its behavior is known and predictable. Incorrectly configured policies can block access to the Tanium Server.

Options for deploying custom quarantine rules and options

You can define quarantine rules and options by either attaching a configuration file to the package, or by selecting options in the Tanium Console when you deploy a quarantine action.

Attach configuration file to package

You can attach a `taniumquarantine.dat` configuration file that defines quarantine rules and options to either a new package or the existing Quarantine packages. Then push that package out to the endpoints. For an example `taniumquarantine.dat` file, see Reference: Custom rules examples on page 73.

1. From the Main menu, go to **Content > Packages**.
2. You can either create a new package, or edit one of the existing Quarantine packages:
   - **Apply Windows IPsec Quarantine**
   - **Apply Mac PF Quarantine**
   - **Apply Linux IPTables Quarantine**.
3. Update the `taniumquarantine.dat` file.
   a. To download the current file, click **Download** ⬇.
   b. Remove the file that is currently in the package 🗑.
   c. Click **Add** to upload the updated `taniumquarantine.dat` file.

4. Click **Save** to save the updates to the package.

Select options in user interface when you deploy Quarantine actions

When you deploy the **Apply Windows IPsec Quarantine**, **Apply Mac PF Quarantine**, or **Apply Linux IPTables Quarantine** actions, you can define the quarantine rules and options as a part of that action. For more information, see Test quarantine on lab endpoints on page 68.

Reference: Custom rules and options

Custom rules format

The format for custom rules is not case sensitive. You can put each rule on a new line. Trailing white spaces are not supported. This format is used for both the configuration file and in the user interface.

```
Direction:Protocol:IPAddress:CIDR:Port
#Comment
```

Direction

Valid values: `IN` or `OUT`
Specifies whether incoming or outgoing traffic is allowed.

Protocol

Valid values: `ICMP`, `TCP`, `UDP`
If you specify ICMP, the ICMP protocol is allowed to communicate to and from the specified addresses. This limitation is because IPSec does not filter ICMP Type/Codes. The filtering is done by ADVFirewall.

IPAddress

Specifies any IPv4 address or you can use ANY for all.

CIDR

Valid values: `0-32` or undefined
Subnet masks in dotted decimal format are not permitted in the input file.
Undefined (blank) is same as 32 and uses the IP Address only.

Port

Valid values: `0-65535` or undefined
Leave undefined (blank) to permit all ports. Ranges are not currently supported, only individual ports or all ports can be defined.

> **Note:** When using the Custom Quarantine Rules parameter in the package, the total characters should be 1100 or less. If you need more characters, you can use a custom DAT file.

Quarantine options

You can configure quarantine options in a configuration file or in the deploy action user interface when you quarantine an endpoint.

Configuration file format

```
OPTION:OptionName:OptionValue
```

Options

Option Name (Deploy Action screen in Tanium Console)	Option Name (configuration file)	Description
Allow All DHCP	*AllDHCP*	Set to `true` to allow DHCP traffic to any server. Default: `true`
Allow All DNS	*AllDNS*	Set to `true` to allow DNS traffic to any destination. Default: `true`
N/A	*CurrentDNS*	Set to `true` to allow DNS traffic to only the Current DNS. Default: `false`
Allow All Tanium Servers	*TaniumServers*	Set to `true` to allow Tanium traffic to the Tanium Servers that are defined in your ServerList or Servers configuration on the Tanium Client. Default: `true`
Allow Alternate Tanium Servers	*ALTTS*	Specify the IP addresses. For example, use this option when you want to avoid using DNS during Quarantine. When removed from quarantine, the original Tanium Server is restored. Separate with a comma or leave empty for no alternates.

Option Name (Deploy Action screen in Tanium Console)	Option Name (configuration file)	Description
Validate Tanium Server Availability	*CheckTS*	Set to `true` to validate that the Tanium Server can be reached on the Tanium port. If this validation fails, back out the rules. Default: `true`
VPN Servers	*VPNSERVERS*	Specify the VPN servers to automatically create rules for with a comma-separated list. Adding servers creates rules for each host as follows: IP:50/51, UDP:500, 4500 TCP/UDP:443 Default: `NO VPNServers`
Notification Message	*Notify*	Specify a string message to notify the user that the system is being quarantined. The message limit is 255 characters. Certain characters are not allowed, such as ($), (!), (`), (‘),(*) and some characters require escapes, such as (\"). Work with your TAM to test any special characters before using in production. Default: `No notification`

Reference: Custom rules examples

Example for Custom Quarantine Rules field

```
IN:UDP:10.0.0.21:32:161
OUT:UDP:10.0.0.21:32:162
```

This example defines two rules:

- Allow SNMP queries (UDP Port 161) from another device at 10.0.0.21.
- Allow SNMP traps (UDP Port 162) to be sent to a device at 10.0.0.21.

> **Note:** This example demonstrates the use of parameter options in the package and not a `taniumquarantine.dat` file.

`taniumquarantine.dat` sample file

For DAT files, each entry must be on one line; you cannot use pipe (|) characters to combine lines. Trailing white spaces are not supported.

```
#Allow ICMP out to a specific IP Address
OUT:ICMP:192.168.10.15::0
#Allow ICMP in from a specific IP Address
IN:ICMP:192.168.20.10:32:0
#Allow TCP port 80 in from a class C subnet
IN:TCP:192.168.1.0:24:80
#Allow UDP port 161 in from a specific IP Address
IN:UDP:10.0.0.21:16:161
#Allow HTTPS (tcp 443) out to a specific class B subnet
OUT:TCP:192.168.0.0:16:443
OPTION:ALLDNS:TRUE
OPTION:CURRENTDNS:FALSE
OPTION:ALLDHCP:TRUE
OPTION:TANIUMSERVERS:TRUE
OPTION:CHECKTS:TRUE
OPTION:NOTIFY:This Device has been Quarantined
```