Splunk® Platform Upgrade Readiness App
Use the Splunk Platform Upgrade Readiness App 2.2.0

Generated: 1/31/2020 8:12 am
# Table of Contents

**Get started**
- About the Splunk Platform Upgrade Readiness App.................................1
- Install the Splunk Platform Upgrade Readiness App..............................2
- Manage permissions for the Splunk Platform Upgrade Readiness App....4

**Check your upgrade readiness**
- Scan a Splunk platform instance with the Splunk Platform Upgrade Readiness App.................................................................6
- REST API endpoint reference for the Splunk Platform Upgrade Readiness App..................................................................................10

**Release notes**
- Release notes for the Splunk Platform Upgrade Readiness App...........23
- Share data in the Splunk Platform Upgrade Readiness App................24
- Credits for the Splunk Platform Upgrade Readiness App....................25
Get started

About the Splunk Platform Upgrade Readiness App

Use the Splunk Platform Upgrade Readiness App to prepare your Splunk platform instance for upgrade to the next version. The Upgrade Readiness App scans selected Splunk apps installed on a Splunk Enterprise version 7.1, 7.2, 7.3, or 8.0 instance for features and code that will not work in some versions of the Splunk platform.

The scan results describe how to take action to ensure the apps and customizations you are responsible for work as expected after you upgrade to a new version of the Splunk platform. For example, the scan results indicate if any apps and customizations rely on Python 2. This is important because support for Python 2 is being phased out. The Splunk Enterprise version 8.0 release includes both the Python 2.7 and Python 3.7 runtimes to help customers transition from Python 2 to Python 3. Splunk will remove the Python 2.7 runtime altogether in a future release.

For more information about the Splunk Enterprise Python 3 release, see Python 3 migration with the Splunk platform.

The Splunk Platform Upgrade Readiness App is tool for assisting with your transition between Python 2 and Python 3. Even if your deployment passes all checks, you still need to independently test your private and customized apps and add-ons for incompatibilities.

What the scan checks for

The Upgrade Readiness App checks for:

- Use of Advanced XML
- Use of SplunkWeb legacy mode
- The existence of files named test.py
- Dependency on the Swig or M2Crypto modules
- Use of custom Mako templates
- Use of custom CherryPy endpoints
- Python scripts

For more information about these checks and what to do with the results, see Scan your instance with the Splunk Platform Upgrade Readiness App.
What apps are included in the scan

When you install the Splunk Platform Upgrade Readiness App on a Splunk platform instance, it scans all apps and add-ons in the `$SPLUNK_HOME/etc/apps` folder, with some exceptions. The app scans the private apps that you have created for your own organization as well as apps that you have installed from Splunkbase, whether they are third-party apps or Splunk Supported apps. The scan includes disabled apps.

Exceptions

Even if you have them installed, the Upgrade Readiness App excludes the following apps from the scan:

- Python for Scientific Computing
- Search & Reporting
- Splunk App for Infrastructure
- Splunk App for PCI Compliance
- Splunk App for VMWare
- Splunk Business Flow
- Splunk Cloud Gateway
- Splunk Enterprise Security
- Splunk Industrial Asset Intelligence
- Splunk IT Service Intelligence

See also

To get started, see Install the Splunk Platform Upgrade Readiness App.

For instructions on how to use the app, see Scan your instance with the Splunk Platform Upgrade Readiness App.

For information about features, fixed issues, and known issues in this app, see the Release notes for the Splunk Platform Upgrade Readiness App.

Install the Splunk Platform Upgrade Readiness App

Install the Splunk Platform Upgrade Readiness App on all instances of your Splunk platform deployment that run apps or on a single test Splunk platform instance with all your apps installed on it.
Requirements

To install the Upgrade Readiness App, you must have installed Splunk Enterprise or Splunk Cloud version 7.1, 7.2, 7.3, or 8.0 with apps or add-ons. You must also have KV store enabled.

Performance considerations

To limit the performance impact of running scans, install the Upgrade Readiness App on a single test Splunk platform instance. Install all of your apps and add-ons on this test instance, then install the Upgrade Readiness App and run scans there.

Memory usage and scan time are proportional to the cumulative size of the Python files across all of your apps. For example, a Splunk Enterprise 7.3.0 instance with about 9,000 Python files across 15 apps might take about 75-100 minutes to complete a scan. The user session timeout limit is 60 minutes by default, so change your limit before conducting large scans. For more information, see Configure user sessions timeouts.

Install the app on an unclustered Splunk Enterprise instance

Prerequisite

You must have the admin_all_objects capability to install the app.

Steps

Follow these steps to install an app on an unclustered search head, forwarder, or single-instance Splunk Enterprise deployment:

1. If you have installed a previous version of the Upgrade Readiness App, delete that version and clear your cache before you install the Upgrade Readiness App version 2.0.
2. Download the Upgrade Readiness App from Splunkbase.
3. From the Splunk Web home screen, click the gear icon next to Apps.
4. Click Install app from file.
5. Locate the downloaded file and click Upload.

The app icon for the Upgrade Readiness App is not visible until you restart your instance.
Next, see Manage permissions for the Splunk Platform Upgrade Readiness App.

**Install the app to a search head cluster**

To deploy the app to the search head cluster members, use the deployer. See Use the deployer to distribute apps and configuration updates in *Distributed Search* in the Splunk Enterprise documentation.

Next, see Manage permissions for the Splunk Platform Upgrade Readiness App.

**Install the app in Splunk Cloud**

You can also install the Upgrade Readiness App with Splunk Cloud 7.1, 7.2, 7.3, or 8.0:

- If you have a single-instance deployment, use self-service install.
- If you have a distributed deployment, you cannot use self-service install to install the Upgrade Readiness App. Submit a case using the Splunk Support Portal to request this app on your Splunk Cloud instance.

Next, see Manage permissions for the Splunk Platform Upgrade Readiness App.

**Manage permissions for the Splunk Platform Upgrade Readiness App**

After you install the Splunk Platform Upgrade Readiness App on your Splunk Platform instance, grant permissions to the roles that are allowed to run scans.

**Prerequisite**

You must be a Splunk administrator or have the `admin_all_objects` capability in order to edit app permissions.

1. From the Splunk Web home page, click the gear icon next to Apps in the apps sidebar to open the Apps page.
2. In the row for the Splunk Platform Upgrade Readiness App, click Permissions.
3. Under App permissions, click the checkbox under Write for all roles that you want to give the ability to run scans.
4. Click Save.
Users assigned to roles who have write access to the Splunk Platform Upgrade Readiness App can run scans. Each user's scan returns results for the apps to which they have at least read permissions. See Scan your instance with the Splunk Platform Upgrade Readiness App.
Check your upgrade readiness

Scan a Splunk platform instance with the Splunk Platform Upgrade Readiness App

Prerequisites

To run scans, you need write permissions to the Splunk Platform Upgrade Readiness App. See Manage permissions for the Splunk Platform Upgrade Readiness App.

Ensure you have taken performance factors into consideration. See Install the Splunk Platform Upgrade Readiness App.

Steps

To scan your Splunk platform instance with the Splunk Platform Upgrade Readiness App in Splunk Web, do the following:

1. Launch the Splunk Platform Upgrade Readiness App.
2. Click Run New Scan.
3. Select which apps to scan. You can scan private apps, Splunkbase apps that you extended with your own customizations, all apps, or a custom selection of apps. Private apps include any apps not found on Splunkbase, such as apps that you created.
4. Review the results of the scan for each app.
5. (Optional) Export your scan results by clicking Export Results. You can choose to export in either JSON or CSV format.

You can also run scans using the REST API. See REST API endpoint reference for the Splunk Platform Upgrade Readiness App.

Some Splunk apps are too large to scan. If you cannot scan a Splunk app, follow the app’s documentation for updates on Python 3 readiness.

Act on scan results

Review and act on the scan results for all your private apps. Private apps are apps that are private to your organization and not available on Splunkbase.
Resolve blocking issues in your private apps and your customized Splunkbase apps, if any, as these will block successful upgrade to the Splunk Enterprise Python 3 release. Check all other file paths flagged by the app and determine what actions to take to make them upgrade ready.

If one or more checks in an app is marked "SKIPPED", this means that the Upgrade Readiness App wasn't able to complete the check due to the way the app is packaged. You can repackage the app using the Splunk Packaging Toolkit and run the scan again, or check for the upgrade compatibility issues manually. See Overview of the Splunk Packaging Toolkit on the Splunk developer portal for more information.

Apps that are compatible with Splunk Enterprise 8.0 and Python 3 are marked "PASSED", and you don't need to take any further action. If you have a version of an app that is not compatible with Splunk Enterprise 8.0 and Python 3, but a compatible version is available, the scan results direct you to download that version from Splunkbase to pass the check.

**Remove Advanced XML**

Advanced XML has been deprecated since Splunk platform version 6.3.0. If the scan identifies use of Advanced XML in your app, locate the file path containing the Advanced XML and rewrite the file to use Simple XML. For information about building dashboards and other visualizations using Simple XML, see Editing Simple XML in the Splunk Enterprise *Dashboards and Visualizations* manual.

**Disable Splunk Web legacy mode**

Splunk Web legacy mode has been deprecated since Splunk platform version 6.4.0. Set `appServerPorts` to a non-zero value in `web.conf`, because a zero value no longer triggers Splunk Web to operate in legacy mode in the Splunk Enterprise Python 3 release. For more information about this setting, see `web.conf` in the Splunk Enterprise *Admin Manual*.

**Rename files named test.py**

In Python 3, `test.py` is a reserved file name. Rename any files named `test.py` to a non-reserved file name to avoid file name conflicts.

**Remove dependency on the M2Crypto and Swig libraries**

The M2Crypto and Swig packages and their dependencies have been removed from Splunk Enterprise version 8.0 and later. Incorporate the libraries into your
apps as needed, or remove all dependencies on these libraries.

**Update Mako templates**

In the Splunk Enterprise Python 3 release, some app server components of Splunk Web will use the Python 3.7 interpreter. Update all Mako templates to be Python 3 compatible so they will work on Splunk Web in the new release. If you want the app to continue to work on previous Splunk platform releases, make the syntax dual compatible with Python 2 and 3. If your organization requires you to remove Python 2 syntax completely by a certain schedule, you can rewrite the Mako templates to use dual-compatible syntax as an intermediate step during your upgrade process, then rewrite them to use Python 3 syntax at a later time.

**Update CherryPy endpoints**

In the Splunk Enterprise Python 3 release, some app server components of Splunk Web will use the Python 3.7 interpreter. Update all CherryPy endpoints to be Python 3 compatible so they will work on Splunk Web in the new release. If you want the app to continue to work on previous releases, make the syntax dual compatible with Python 2 and 3. If your organization requires you to remove Python 2 syntax completely by a certain schedule, you can rewrite them to dual-compatible syntax as an intermediate step during your upgrade process, then rewrite them to Python 3 syntax at a later time.

**Update all other Python files**

The Splunk Enterprise Python 3 release will include interpreters for both Python 2 and 3. In that release, for all Python code not dependent on Splunk Web or the app server, Splunk platform administrators can choose to use either interpreter by default and on a script-by-script basis. In a later release, Splunk plans to remove the Python 2 interpreter.

As a best practice, update Python files in your apps to be compatible with both Python 2 and 3 so that they are backwards compatible with previous releases, compatible with either Python interpreter in the Splunk Enterprise Python 3 release, and compatible with future releases when the Splunk platform removes the Python 2 interpreter. If your organization requires you to remove Python 2 syntax completely by a certain schedule, rewrite them using Python 3 syntax.

To make Python scripts dual compatible, use a Python 2/3 compatibility utility such as six, provided by the Python Software Foundation. For more about the six library, see https://pypi.org/project/six/.
If the upgrade readiness app identifies one or more Python scripts that are used in scripted alerts, convert those scripted alerts to custom alert actions. Scripted alerts are deprecated and might not work in future releases of the Splunk platform. See Convert a script alert action to a custom alert action in *Developing Views and Apps for Splunk Web* in the Splunk Enterprise documentation, but note that the script example shown for the custom alert action has not yet been revised to demonstrate dual-compatible Python syntax.

**What to do with scan results for Splunk-supported and third-party apps**

The Splunk Platform Upgrade Readiness App scan results include Splunk-supported apps and third-party apps supported by partners and developers. App owners are responsible for updating their apps and releasing new versions that meet these upgrade readiness requirements in order to support the Splunk Enterprise Python 3 release.

If you've extended an app or customized anything locally, review the results for any custom file paths that you've added to that app and take action on those to prepare for upgrade. Otherwise, you can wait for the app owner to make updates.

For third-party apps, you can contact the developer directly using their contact details on Splunkbase to learn more about their upgrade plans. If the developer does not plan to update the app, you can make the updates yourself using the guidance in the scan checks.

**Dismiss file paths**

The Upgrade Readiness App allows you to dismiss file paths from the scan results. Dismiss files that you have verified are already Python 3 compatible so that you can narrow the list to files that still need your attention.

When you dismiss a file path, it removes it from all future scan results.

If you dismiss a path by accident and want to reinstate it in future scans, delete the Splunk Platform Upgrade Readiness App from your Splunk platform instance and then reinstall it.
REST API endpoint reference for the Splunk Platform Upgrade Readiness App

Use the REST API to run scans programmatically on your Splunk platform instance.

Authentication and Authorization
To use this REST API, you must have write permissions to the Splunk Platform Upgrade Readiness App.

/services/spura_app_list?type={scan_type}

https://<host>:<mPort>/services/spura_app_list?type={scan_type}
Lists the apps for a user by the scan type

Usage details
To use this endpoint, you must have permission to see the apps on your instance.

GET
Fetches the list of apps for a user by the scan type.

Request parameters
None.

Returned values

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;name&quot;: &lt;folder_name_of_apps&gt;,</td>
</tr>
<tr>
<td></td>
<td>&quot;label&quot;: &lt;user_friendly_name_of_app&gt;,</td>
</tr>
<tr>
<td></td>
<td>&quot;type&quot;: &lt;type_of_app&gt; (Splunkbase App, SplunkSupported App, Private App),</td>
</tr>
<tr>
<td></td>
<td>&quot;link&quot;: &lt;link_of_app_on_splunkbase&gt;</td>
</tr>
<tr>
<td>400, 403, 404,</td>
<td>{</td>
</tr>
<tr>
<td>or 500</td>
<td>&quot;status&quot;: false,</td>
</tr>
<tr>
<td></td>
<td>&quot;message&quot;: &lt;failure_cause&gt;</td>
</tr>
</tbody>
</table>

Example request and response
Request

curl -k -u username:password
https://localhost:8089/services/spura_app_list?type=splunkbase -X GET

Response

{
   "name": "Splunk_TA_kafka",
   "type": "Splunk Supported App",
   "label": "Splunk Add-on for Kafka",
   "link": "https://splunkbase.splunk.com/app/2935/"
}

/services/spura_scan_deployment

https://<host>:<mPort>/services/spura_scan_deployment

Triggers a new scan of the apps that you have at least Read access to on the Splunk platform instance.

Usage details
To use this endpoint, a scan must not be running.

POST

Starts a scan and writes progress in KV store

Request parameters
None.

Payload

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>apps</td>
<td>String</td>
<td><strong>Required.</strong> List of apps to scan. Obtain this information from the /spura_app_list/ endpoint.</td>
</tr>
<tr>
<td>apps.name</td>
<td>String</td>
<td><strong>Required.</strong> Name of the app (folder_name).</td>
</tr>
<tr>
<td>apps.label</td>
<td>String</td>
<td><strong>Required.</strong> User-friendly name of the app.</td>
</tr>
<tr>
<td>app.type</td>
<td>String</td>
<td><strong>Required.</strong> Type of app. Acceptable values: Splunkbase App</td>
</tr>
<tr>
<td>app.link</td>
<td>String</td>
<td>Link to the app on Splunkbase.</td>
</tr>
</tbody>
</table>

Returned values
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>`{ “status”: true, “message”: &lt;success_message&gt; }</td>
</tr>
<tr>
<td>400, 403, 404, or 500</td>
<td>`{ “status”: false, “message”: &lt;failure_cause&gt; }</td>
</tr>
</tbody>
</table>

**Example request and response**

**Request**

curl -k -u username:password  

**Response**

```json
{
  "status": true,
  "message": "Scan called"
}
```

/services/spura_read_progress

https://<host>:<mPort>/services/spura_read_progress  
Reads current progress for an ongoing or completed scan.

**Usage details**

To use this endpoint, you must be running a scan or have completed a scan.

**GET**

Reads current progress of an ongoing scan on the instance from the KV store or fetches the previous scan results for the user.

**Request parameters**

None.

**Returned values**
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| 200  | {
|      |   "status": <status_of_scan>,
|      |   "message": <message_related_to_status>,
|      |   "progress": <progress_percentage_value>,
|      |   "results": { <dict_of_results_for scan> }
|      | }
| 400, 403, 404, or 500 | {
|      |   "status": false,
|      |   "message": <failure_cause>
|      | }

Example request and response

Request

curl -k -u username:password
https://localhost:8089/services/spura_read_progress  -X GET

Response when scan is in progress

{
   "message": "1 apps out of 8 scanned. Scanning App: Splunk Add-on Example",
   "progress": 12,
   "results": {},
   "status": "IN_PROGRESS"
}

Response when scan is complete

{
   "message": "Deployment scanned successfully for user: admin",
   "progress": 100,
   "results": {
      "apps": [],
      "checks": [
         {
            "description": "Check for Advanced XML <module> elements. The Module system was deprecated in Splunk 6.3 as part of the advanced XML deprecation. See: http://docs.splunk.com/Documentation/Splunk/latest/Module",
            "messages": [],
            "result": "PASSED",
         }
      ]
   }
}
"name": "Advanced XML",
"required_action": "No required action."
},
{
"description": "Detect the existence of CherryPy endpoints. With the upcoming Splunk Python 3 migration, Splunkweb will run entirely on Python 3. As custom CherryPy endpoints (AKA custom web controls) run on Splunkweb, Python code in custom CherryPy endpoints must be upgraded to be compatible with Python 3."
,"messages": []
,"result": "PASSED",
"name": "Custom CherryPy endpoints",
"required_action": "No required action."
},
{
"description": "Check for the existence of Mako template. Mako template should be compatible with Python 3 in the upcoming Splunk Python 3 migration."
,"messages": [
{
"result": "warning",
"message_line": null,
"message": "With the upcoming Splunk Python 3 migration, Splunk Web will run entirely on Python 3. As custom Mako templates run on Splunk Web, Python code in Mako templates must be upgraded to be compatible with Python 3, and failing to upgrade it may cause breakage. Please ensure your Mako template is compatible with Python 3. If you've finished your upgrade, please disregard this message. File: appserver/static/js/templates/Dialogs/Error.html",
"message_filename": "appserver/static/js/templates/Dialogs/Error.html"
},
{
"result": "warning",
"message_line": null,
"message": "With the upcoming Splunk Python 3 migration, Splunk Web will run entirely on Python 3. As custom Mako templates run on Splunk Web, Python code in Mako templates must be upgraded to be compatible with Python 3, and failing to upgrade it may cause breakage. Please ensure your Mako template is compatible with Python 3. If you've finished your upgrade, please disregard this message. File: appserver/static/js/templates/Models/LoadingMsg.html",
"message_filename": "appserver/static/js/templates/Models/LoadingMsg.html"
},
{
"result": "warning",
"message_line": null,
"message": "With the upcoming Splunk Python 3 migration, Splunk Web will run entirely on Python 3. As custom Mako templates run on Splunk Web, Python code in Mako templates must be upgraded to be compatible with Python 3, and failing to upgrade it may cause breakage. Please ensure your Mako template is compatible with Python 3. If you've finished your upgrade, please disregard this message. File: appserver/static/js/templates/Models/LoadingMsg.html",
"message_filename": "appserver/static/js/templates/Models/LoadingMsg.html"
}]}
cause breakage. Please ensure your Mako template is compatible with Python 3. If you've finished your upgrade, please disregard this message. File: appserver/templates/base.html",
  "message_filename": "appserver/templates/base.html"
}
  }
  "result": "BLOCKER",
  "name": "Custom Mako templates",
  "required_action": "Check to ensure that Mako templates are upgraded to be compatible with Python 3."
},
  {
    "description": "Check removed M2Crypto usages.",
    "messages": [],
    "result": "PASSED",
    "name": "M2Crypto module",
    "required_action": "No required action."
  },
  {
    "description": "Check reserved filename test.py.",
    "messages": [],
    "result": "PASSED",
    "name": "test.py",
    "required_action": "No required action."
  },
  {
    "description": "Check that the 'appServerPorts' property in the web.conf does not contain
port 0.",
    "messages": [],
    "result": "PASSED",
    "name": "Splunkweb legacy mode",
    "required_action": "No required action."
  },
  {
    "description": "Check for existence of Python scripts. All Python scripts must be updated due to the upcoming Splunk Python 3 migration.",
    "messages": [
      {
        "result": "warning",
        "message_line": "N/A",
        "message": "The file at path /opt/splunk/etc/apps/TA-example/bin/update_state.py needs to be checked for Splunk Python 3 migration",
        "message_filename": "/opt/splunk/etc/apps/TA-example/bin/update_state.py"
      },
      {
        "result": "WARNING",
        "name": "All Python scripts",
        "required_action": "Update these Python scripts to
be dual-compatible with Python 2 and 3.

/services/spura_cancel_scan

https://<host>:<mPort>/services/spura_cancel_scan
Cancels an ongoing scan for the user on the instance.

Usage details
To use this endpoint, you must be running a scan.

DELETE

Sets canceled flag for the ongoing scan which restricts the scan from writing further progress in the KV store and allows the thread to return.

Request parameters
None.

Returned values
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;status&quot;: COMPLETE,</td>
</tr>
<tr>
<td></td>
<td>&quot;message&quot;: &lt;message_for_cancellation&gt;,</td>
</tr>
<tr>
<td></td>
<td>&quot;progress&quot;: 100</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>400, 403, 404, or 500</td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;status&quot;: false,</td>
</tr>
<tr>
<td></td>
<td>&quot;message&quot;: &lt;failure_cause&gt;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>

Example request and response

Request

```
curl -k -u username:password
https://localhost:8089/services/spura_cancel_scan  -X DELETE
```
Response

```
{
   "status": "COMPLETE",
   "message": "Scan for user: admin on host: localhost cancelled successfully",
   "progress": 100
}
```

/services/spura_export_report?id={scan_id}&format={file_format}

https://host:<mPort>/spura_export_report?id={scan_id}&format={file_format}
Fetches a JSON or CSV report for a given scan_id.

Usage details
In order to use this endpoint, a previous scan result must be available, and you must know its scan_id.

GET

Fetches the JSON or CSV report for a previous scan, given a scan_id.

Request parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>
### Required

The scan id of a completed scan run by this user.

### Format

**String**

**Required.** The format of the report.

### Returned values

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| 200  | { 
|     |   "apps": [ <list_of_apps> ], 
|     |   "summary": { 
|     |     <dict_of_summary_of_scan> }, 
|     |   "scan_id": <id_of_the_scan_report> 
| 400, 403, 404, or 500 | { 
|      |   "status": false, 
|      |   "message": <failure_cause> |

### Example request and JSON response

#### Request

```bash
curl -k -u username:password https://localhost:8089/services/spura_export_report?id=admin_1565094900&format=JSON -X GET
```

#### Response

```json
{
   "apps": [ 
    {
        "checks": [ 
            
            "description": "Check for Advanced XML <module> elements. The Module system was deprecated in Splunk 6.3 as part of the advanced XML deprecation. See:\nhttp://docs.splunk.com/Documentation/Splunk/latest/Module",
            "messages": [],
            "result": "PASSED",
            "name": "Advanced XML",
            "required_action": "No required action."
        

        

}| 
```
endpoints. With the upcoming Splunk Python 3 migration, Splunkweb will run entirely on Python 3. As custom CherryPy endpoints (AKA custom web controls) run on Splunkweb, Python code in custom CherryPy endpoints must be upgraded to be compatible with Python 3."

"messages": [],
"result": "PASSED",
"name": "Custom CherryPy endpoints",
"required_action": "No required action."
},
{
"description": "Check for the existence of Mako template. Mako template should be compatible with Python 3 in the upcoming Splunk Python 3 migration."
"messages": [
{
"result": "warning",
"message_line": null,
"message": "With the upcoming Splunk Python 3 migration, Splunk Web will run entirely on Python 3. As custom Mako templates run on Splunk Web, Python code in Mako templates must be upgraded to be compatible with Python 3, and failing to upgrade it may cause breakage. Please ensure your Mako template is compatible with Python 3. If you've finished your upgrade, please disregard this message. File: appserver/static/js/templates/Dialogs/Error.html",
"message_filename": "appserver/static/js/templates/Dialogs/Error.html"
},
{
"result": "warning",
"message_line": null,
"message": "With the upcoming Splunk Python 3 migration, Splunk Web will run entirely on Python 3. As custom Mako templates run on Splunk Web, Python code in Mako templates must be upgraded to be compatible with Python 3, and failing to upgrade it may cause breakage. Please ensure your Mako template is compatible with Python 3. If you've finished your upgrade, please disregard this message. File: appserver/static/js/templates/Models/LoadingMsg.html",
"message_filename": "appserver/static/js/templates/Models/LoadingMsg.html"
},
{
"result": "warning",
"message_line": null,
"message": "With the upcoming Splunk Python 3 migration, Splunk Web will run entirely on Python 3. As custom Mako templates run on Splunk Web, Python code in Mako templates must be upgraded to be compatible with Python 3, and failing to upgrade it may cause breakage. Please ensure your Mako template is compatible with Python 3. If you've finished your upgrade, please disregard this message. File: appserver/templates/base.html",
"message_filename": "appserver/templates/base.html"
"result": "BLOCKER",
"name": "Custom Mako templates",
"required_action": "Check to ensure that Mako templates are upgraded to be compatible with Python 3."
},
{
  "description": "Check removed M2Crypto usages.",
  "messages": [],
  "result": "PASSED",
  "name": "M2Crypto module",
  "required_action": "No required action."
},
{
  "description": "Check reserved filename test.py.",
  "messages": [],
  "result": "PASSED",
  "name": "test.py",
  "required_action": "No required action."
},
{
  "description": "Check that the 'appServerPorts' property in the web.conf does not contain \n port 0.",
  "messages": [],
  "result": "PASSED",
  "name": "Splunkweb legacy mode",
  "required_action": "No required action."
},
{
  "description": "Check for existence of Python scripts. All Python scripts must be updated due to the upcoming Splunk Python 3 migration.",
  "messages": [
    {
      "result": "warning",
      "message_line": "N/A",
      "message": "The file at path /opt/splunk/etc/apps/TA-example/bin/update_state.py needs to be checked for Splunk Python 3 migration",
      "message_filename": "/opt/splunk/etc/apps/TA-example/bin/update_state.py"
    },
  ],
  "result": "WARNING",
  "name": "All Python scripts",
  "required_action": "Update these Python scripts to be dual-compatible with Python 2 and 3."
}
],
"summary": {
  "type": "Splunkbase App",}
dismiss a file path from the scan report and skip it from future scans.

Usage details
To use this endpoint, a previous scan must be available for this user.

POST

Writes an entry for a specified file path in the KV store for the user and update the JSON report by removing the file path entry.

Request parameters
None.

Payload

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>app</td>
<td>String</td>
<td>Required. Folder name of the app containing the file path you want to dismiss.</td>
</tr>
</tbody>
</table>
**Name** | **Type** | **Description**
---|---|---
*check* | **String** | **Required.** The name of the check containing the file path you want to dismiss. **Acceptable values:** Custom Mako templates | Custom CherryPy endpoints | Python scripts
*file_path* | **String** | **Required.** The file path you want to dismiss.

**Returned values**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td></td>
</tr>
</tbody>
</table>

```
{
   "status": true,
   "message": <success_message>
}
```

| 400, 403, 404, or 500 | 

```
{
   "status": false,
   "message": <failure_cause>
}
```

**Example request and response**

**Request**

curl -k -u username:password
https://localhost:8089/services/spura_dismiss_file -H 'Content-Type: application/json' -d '{"app": "TA-example", "check": "All Python scripts", "file_path": "/opt/splunk/etc/apps/TA-example/bin/ta_example/__init__.py"}' -X POST

**Response**

```
{
   "message": "File: /opt/splunk/etc/apps/TA-example/bin/ta_example/__init__.py for check: All Python scripts for app: TA-example successfully registered for dismissing for user: admin on host: cds.local. The fresh scan results would skip this file.",
   "success": true
}
```
Release notes

Release notes for the Splunk Platform Upgrade Readiness App

The Splunk Platform Upgrade Readiness App version 2.2.0 was released on January 27, 2020. This version is compatible with Splunk Enterprise and Splunk Cloud versions 7.1, 7.2, 7.3, and 8.0.

Known issues

This version of the Splunk Platform Upgrade Readiness App has the following known issues and workarounds.

<table>
<thead>
<tr>
<th>Publication date</th>
<th>Defect number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020-01-13</td>
<td>SPL-181633</td>
<td>Installing the Splunk Cloud Gateway App and then running a scan causes the Upgrade Readiness App to hang. If you delete the Splunk Cloud Gateway App, the Upgrade Readiness App resumes working, and you can run scans.</td>
</tr>
<tr>
<td>2019-12-17</td>
<td></td>
<td>If you are upgrading from Splunk Platform Upgrade Readiness App v1.0.0 to this version, delete the old version of the app and clear your browser cache before installing this version. If you upgrade the app and do not follow this process, the scan might skip some apps.</td>
</tr>
</tbody>
</table>

Fixed issues

<table>
<thead>
<tr>
<th>Date resolved</th>
<th>Defect number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-12-09</td>
<td>SPL-178174</td>
<td>The scanning process might terminate abruptly on Windows systems.</td>
</tr>
<tr>
<td>2019-11-25</td>
<td>SPL-177786</td>
<td>Scans can hang if scan time exceeds user session timeout limit. Hung scans appear to still be running. Select Cancel Scan to end the scan, and then break up your list of apps into multiple scans of smaller batches. Alternatively, increase your session timeout</td>
</tr>
<tr>
<td>Date resolved</td>
<td>Defect number</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>duration beyond the 60 minute default before running long scans. See Configure user sessions timeouts.</td>
</tr>
<tr>
<td>2019-11-20</td>
<td>SPL-178115</td>
<td>When all apps pass the scan, sometimes the app does not show the results.</td>
</tr>
<tr>
<td>2019-11-20</td>
<td>SPL-178424</td>
<td>Sometimes the scan fails if it encounters a permissions issue with an app, rather than skipping the app.</td>
</tr>
<tr>
<td>2019-11-20</td>
<td>SPL-176823</td>
<td>Scan might skip files with syntax errors.</td>
</tr>
<tr>
<td>2019-10-16</td>
<td>SPL-176909</td>
<td>If you run the Upgrade Readiness App locally many times in a row, it might begin skipping checks.</td>
</tr>
<tr>
<td>2019-10-14</td>
<td>SPL-176424</td>
<td>CherryPy endpoint check results display web.conf instead of the Python file. Find the Python file in the Python scripts check.</td>
</tr>
<tr>
<td>2019-10-09</td>
<td>SPL-174672</td>
<td>Readiness App displays unfriendly error message when user without permissions attempts a scan.</td>
</tr>
<tr>
<td>2019-10-01</td>
<td>SPL-174819</td>
<td>Scan results for check 1 differs for different operating systems.</td>
</tr>
<tr>
<td>2019-10-01</td>
<td>SPL-174559</td>
<td>Results from check 1 and check 2 are displaying relative paths instead of full paths.</td>
</tr>
</tbody>
</table>

**Share data in the Splunk Platform Upgrade Readiness App**

When the Splunk Platform Upgrade Readiness App is deployed on Splunk Enterprise, the Splunk platform sends anonymized usage data to Splunk Inc. ("Splunk") to help improve the Splunk Platform Upgrade Readiness App in future releases. For information about how to opt in or out and how the data is collected, stored, and governed, see Share data in Splunk Enterprise.

**What data is collected**

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>

24
Finds your version of the Upgrade Readiness App, the type of scans that you run, and the amount of time in minutes that your scans take.

Also, gathers metadata on the types of apps that you scan, such as if your apps are from Splunkbase or not, and the results of the app scans.

```
{
  appVersion: 2.1.0
  scanType: All/Private/Splunkbase/Custom
  scanTypeModified: True/False
  executionTime: 183
  Dashboards_Examples_App: { [-]
    source: Splunkbase/Private
    appStatus:
      Passed/Warning/Blocker/Skipped
    advancedXMLStatus:
      Passed/Blocker/Skipped
    advancedXMLNumber: 10
    cherryPyStatus:
      Passed/Blocker/Skipped
    cherryPyNumber: 2
    MakoXMLStatus:
      Passed/Blocker/Skipped
    MakoNumber: 7
    PythonScriptStatus:
      Passed/Warning/Skipped
    PythonScriptNumber: 57
  }
```

Credits for the Splunk Platform Upgrade Readiness App

The Splunk Platform Upgrade Readiness App contains libraries that were written by others, and are being redistributed as part of the Splunk Platform Upgrade Readiness App under their respective open source licenses. We want to thank the contributors to these projects.

For complete third-party software information for Splunk Platform Upgrade Readiness Apps, download this PDF file:

- Splunk Platform Upgrade Readiness App Third-party software credits