



Slack Integration



Trademarks

RSA, the RSA Logo and EMC are either registered trademarks or trademarks of EMC Corporation in the United States and/or other countries. All other trademarks used herein are the property of their respective owners. For a list of EMC trademarks, go to www.emc.com/legal/emc-corporation-trademarks.htm.

License Agreement

This software and the associated documentation are proprietary and confidential to EMC, are furnished under license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. This software and the documentation, and any copies thereof, may not be provided or otherwise made available to any other person.

No title to or ownership of the software or documentation or any intellectual property rights thereto is hereby transferred. Any unauthorized use or reproduction of this software and the documentation may be subject to civil and/or criminal liability. This software is subject to change without notice and should not be construed as a commitment by EMC.

Third-Party Licenses

This product may include software developed by parties other than RSA.

Note on Encryption Technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing or exporting this product.

Distribution

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license. EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Contents

RSA NetWitness Suite Integration with Slack	4
Prerequisites	4
Integration Steps	4
Create Script Output Action	5
Define Script Notification Server	7
Define a Script Template	8
Update ESA Rule Output Action	9

RSA NetWitness Suite Integration with Slack

As threats evolve, it is important for organizations to keep pace. As part of this trend, many organizations are moving to Slack for team communications and to help drive a more efficient operational workflow. You can use RSA NetWitness Suite to help drive some of the changes as well. You can configure ESA to send alerts from NetWitness Slack using the "run script" capability.

Prerequisites

You need the following items before you begin the integration process:

- Slack Account
- Webhook URL
- Channel and Username

Integration Steps


The remainder of this document discusses the steps for integrating the RSA NetWitness Suite ESA Slack.

1. [Create Script Output Action](#)
2. [Define Script Notification Server](#)
3. [Define a Script Template](#)
4. [Update ESA Rule Output Action](#)

Create Script Output Action

This section describes how to create a script output action.

Login to the RSA NetWitness Suite server UI to complete the following procedure for creating a Script Output Action.

1. In the **Security Analytics** menu, select **Administration > System > Global Notifications**.
2. Ensure the **Output** tab is selected.
3. Click  and select **Script** from the drop-down menu to add a script.
4. Enter a name and description for the script.
5. Copy and paste the following text into the **Script** area:

```
#!/bin/bash
webhook_url="WEB HOOK URL"
channel="CHANNEL NAME"
username="USERNAME"

text=$*
escapedText=$(echo $text | sed 's/"\/"/g' | sed "s/'/\'/g" )
json="{\"channel\": \"\${channel}\", \"username\": \"\${username}\", \"icon_emoji\": \"ghost\",
\"attachments\": [{\"color\": \"danger\", \"text\": \"\${escapedText}\"}]}"
/usr/bin/curl -s -d "payload=$json" "$webhook_url"
```

Your screen should look similar to the following:

Define Script Notification

Enable

Name *

Description

Script * ?


```
webhook_url="WEB HOOK URL"
channel="CHANNEL"
username="USERNAME"

text=$*
escapedText=$(echo $text | sed 's/"^"/g' | sed "s/'^'/g" )
json="{\"channel\": \"$channel\", \"username\": \"$username\",
\"icon_emoji\": \"ghost\", \"attachments\": [{\"color\": \"danger\", \"text\":
\"$escapedText\"}]}"
/usr/bin/curl -s -d "payload=$json" "$webhook_url"
```

Cancel Save

6. Click **Save** to close the dialog box and save the script.

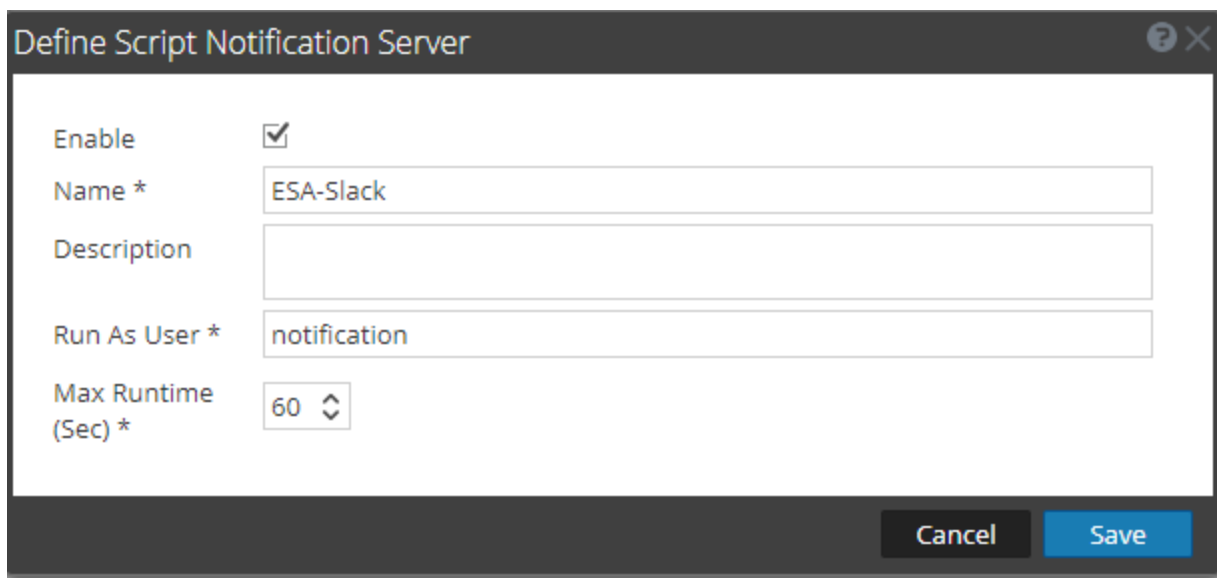
Define Script Notification Server

Next, you need to set Slack as a Global Notification Server.

1. In the **Security Analytics** menu, select **Administration > System Global Notifications**.
2. Click the **Servers** tab.
3. From the **+ ▾** drop-down menu, select **Script**.

The **Define Script Notification Server** dialog is displayed.

4. Enter a name for the notification server, and accept the default values.



The screenshot shows a dialog box titled "Define Script Notification Server". It contains the following fields and controls:

- Enable**: A checkbox that is checked.
- Name ***: A text input field containing "ESA-Slack".
- Description**: An empty text input field.
- Run As User ***: A text input field containing "notification".
- Max Runtime (Sec) ***: A spinner control set to "60".

At the bottom right of the dialog, there are two buttons: "Cancel" and "Save".

5. Click **Save**.

Define a Script Template

In Global Notifications, define a template for Slack.

1. In the **Security Analytics** menu, select **Administration > System**.
2. In the options panel, select **Global Notifications**.
3. Click the **Templates** tab.
4. Click **+** to define a new template.

The Define Template dialog box is displayed.

5. Fill in the fields in the Define Template dialog box as follows:

- Enter a name
- From the **Template Type** menu, select **Event Stream Analysis**
- Enter a description
- Copy and paste the following text into the **Template** feed:

```
<!-- The default template, renders all alert data as a json object --><#include
"macros.ftl"/>ESA Rule Name : ${moduleName} - Alert Severity : ${severity} - Alert Time
: ${time?datetime?iso_utc} - Event source : ${eventSourceId} - id : ${id}
```

Your screen should look similar to the following:

6. Click **Save**.

Update ESA Rule Output Action

Finally, configure a Rule output action.

1. In the **Security Analytics** menu, select **Alerts > Configure**.
2. In the **Rule Library**, select the rule for which you want to receive alerts in Slack.

The RSA Live ESA Rule tab is displayed.

3. Click **+** and select **Script** from the drop-down menu.
4. For the **Notification**, **Notification server** and **Template** fields, select the items you created earlier.

Output	Notification	Notification Server	Template
<input type="checkbox"/> SCRIPT	ESA-Alerts	ESA-Slack	ESA Slack Template

Output Suppression of every minutes

5. Save the rule.
6. Redeploy the updated rule to the ESA Service.

This completes the integration of RSA NetWitness Suite with Slack. Check your slack account for alerts from thr RSA ESA service.

The screenshot shows a Slack channel interface for 'test-esa-alerts'. A notification banner at the top states: 'Your team is getting close to the 10k message limit'. Below this, several messages are visible, each starting with a red vertical bar and containing alert details:

- 4517-aa97-9b3312303ceb
- ESA Rule Name : AlertUserDestination - Alert Severity : 9 - Alert Time : 2017-08-22T10:34:39Z - Event source : [redacted] - id : 2f81a6df-fb7b-42d5-a4d8-f0ac07ba0cf9
- incoming-webhook-esa APP 7:03 PM
- ESA Rule Name : AlertUserDestination - Alert Severity : 9 - Alert Time : 2017-08-22T11:24:56Z - Event source : [redacted] - id : 367079a6-13c5-462f-8874-003b639c6c8e
- ESA Rule Name : AlertUserDestination - Alert Severity : 9 - Alert Time : 2017-08-22T11:25:26Z - Event source : [redacted] - id : 846ed8d1-30fb-4c50-ae8-197d9748848b
- ESA Rule Name : AlertUserDestination - Alert Severity : 9 - Alert Time : 2017-08-22T11:26:47Z - Event source : [redacted] - id : 35391c76-d76b-4f5a-b6b3-970090fbe646