Nexpose Ticketing System Integration Guide for Remedy ITSM

Contents

RAPID7

Last Revised: November 19, 2015

Solution Summary

The goal of incident management is to restore normal service operation as quickly as possible following an incident, while minimizing impact to business operations and ensuring quality is maintained. The integration with Nexpose allows customers to create incident tickets based on vulnerabilities found across their systems. With this information in the Remedy platform, said tickets can be assigned to work teams, prioritized and resolved.

The Remedy Nexpose Ticket Service Integration creates reports based upon a scan of user selected sites or tags, depending on the configuration, and then creates tickets either for each machine and / or vulnerability, depending on the ticketing mode selected. For subsequent scans, the existing tickets are updated (and potentially closed) and new tickets are created, based on the delta from the previous scan.

This service can operate in Default, Vulnerability and IP mode. Tickets are not updated in Default mode.

The helper prepares tickets for sending to Remedy, formatting them depending on the selected ticketing mode. It is then responsible for sending new tickets, updating existing tickets and closing the correct tickets from the Remedy service instance.

Installation and Configuration

Please see the Nexpose Ticketing Configuration Guide for how to install and configure the nexpose_ticketing Gem.

Once all dependencies have been installed, the configuration files need to be edited with the details of the target Nexpose and Remedy clients. To configure, open the configuration files under the config folder found in the Gem installation:

- Windows: C:\Ruby<version>\lib\ruby\gems\<version>\gems\nexpose_ticketing\lib\nexpose_ticketing\config
- Linux: /var/lib/gems/<version>/gems/nexpose_ticketing/lib/nexpose_ticketing/config

Your installation folder may differ; please refer to the Ruby documentation for the specific location.

The sites or tags which are to have tickets generated, as well as the ticketing mode, are defined in ticket_service.config. An example setup of this file can be found in the main Nexpose Ticketing Configuration Guide.

The log-in details are specified within the remedy.config file. Open the file using any text editor and note the following options:

Setting	Description	Sample Value
product	Name of the ticketing service.	Remedy*
vendor	Name of the ticketing software developer.	bmc*
helper_name	This is the helper class name.	RemedyHelper*
create_soap_endpoint	URL to the Remedy creation SOAP interface endpoint.	http://url/arsys/services/ARService?server=bmc-remedy-w&webService=HPD_IncidentInterface_Create_WS
query_modify_soap_endpoint	URL to the Remedy modification SOAP interface endpoint.	http://url/arsys/services/ARService?server=bmc-remedy-w&webService=HPD_IncidentInterface_WS
username	Username of a user within Remedy with rights to create, update and close incidents.	username
password	Password of above username.	password
authentication	Authentication code for the specified account (if the Remedy instance is set up to require it).	mysecretcode
first_name	First name of the above username. Remedy requires you include this data with web service requests.	John
last_name	Last name of the above username. Remedy requires you include this data with web service requests.	Doe
open_timeout	Timeout (in seconds) for opening a SOAP connection.	30
read_timeout	Timeout (in seconds) for reading a SOAP response.	30

* These values should not be changed.

Note: Finding Web Service Endpoints

In many cases, the sample endpoints above (substituting the actual URL for http://url) will work with no issue. However, here are the steps to verify the endpoints are correct:

1. Navigate to the list of web services on the Remedy instance at https://<midtierServer>/arsys/WSDL/protected/list
2. Click on the HPD_IncidentInterface_Create_WS link
3. Scroll to the bottom of the XML document and note the XML node <soap:address... />
4. Copy the location attribute for create_soap_endpoint

Replicate the above steps, substituting HPD_IncidentInterface_WS for the web service link and query_modify_soap_endpoint for the location attribute.

Initial Run

Assuming you've properly configured the Nexpose and Remedy parameters, execute the following command within the 'bin' folder to run the service:

- ruby nexpose_ticketing remedy

Every time this command is executed, the service will query Nexpose and obtain any new vulnerability information and open tickets accordingly.

To view log information, wait for the service to complete execution and then review the logs located in the **/lib/nexpose_ticketing/log** directory**.**

As part of a continuous ticketing program, it is recommended to run the command daily via a Cron job or Windows task.

Troubleshooting

The most common errors when running the script are:

- Configuration errors (such as incorrect indentation or user details).
- Specifying an incorrect or invalid SOAP endpoints.
- Specifying a Nexpose user without permission to create reports.
- Not specifying a site or tag.
- Specifying both a tag and a site. The tag will take priority and the site will not be scanned.

If not enabled, enable logging and view the log files after re-running the service. These files will contain any error information and will also contain information statements about what vulnerability data was found in Nexpose and transmitted to Remedy.

Some additional troubleshooting notes:

- Many Remedy installations are on non-standard ports (where the defaults are 80 and 443). Web service connectivity could run into issues if you do not specify the correct port for the two endpoints.
- If the web services connect successfully but cannot create incidents, there may be an issue with automatic assignment of incidents. In the current version of Nexpose-to-Remedy integration, automatic group assignment of incidents must be enabled or incident creation will fail.
  Configure group assignment using the Application Administration Console from within Remedy.
- Newer versions of Remedy may have slightly updated WSDLs. In the case that the WSDL on the Remedy instance differs from the one located in /lib/nexpose_ticketing/config/remedy_wsdl, please contact Rapid7 for assistance.

If the service is still experiencing a problem after troubleshooting, please send an email to support@rapid7.com with the rapid7_remedy_version.log and ticket_service.log attached and a description of the issue.

Helper Method Overview

There are several methods implemented in the Remedy_Helper common to all helpers, for creating, updating and closing tickets, as well as several Remedy specific methods. Below is an outline of the methods, along with an explanation of when they are called and how they work:

generate_new_ticket

- Description: Method to generate a new Savon-based ticket object for sending to Remedy. Used as a base generic ticket when preparing tickets to send, it can take extra fields as parameters, allowing for customisation and sending extra data. Inserts the standard ticket data, leaving the summary and note fields empty for completion in a subsequent method (when preparing tickets to send)
- Used By: This method is used by the prepare_tickets method to create a new ticket object to hold data about a ticket.

get_client

- Description: A method to create and return a Savon client object for sending tickets to Remedy. This takes the wsdl file describing the correct network service and a specific endpoint as parameters for connecting to the service instance and uses the user information stored in the remedy.config file.
- Used By: This method is called by the create_tickets, update_tickets, close_tickets and query_for_tickets methods, to create the savon client needed to send tickets or information to the correct endpoint on the Remedy instance. This client will then be subsequently called with the data to be sent.

send_tickets

- Description: This method sends a list of tickets (in SOAP format) to Remedy individually (each ticket in the list as a separate web service call) using a client created beforehand. An array of prepared tickets to send is provided to the method. The method then calls the Savon client, passing in the chosen service value (submit or modify) and the current ticket as the message.
- Used By: This method is used by the create_tickets, update_tickets and close_tickets methods to send created tickets to the Remedy instance.

create_tickets

- Description: This method sends a list of prepared new tickets to the Remedy instance. First it creates the Savon client for sending the tickets by calling the get_client method, passing in the wsdl create endpoint and the SOAP create endpoint. It then calls the send_tickets method, with the newly created client, the submit service and the ticket array as parameters to send the tickets to Remedy.
- Used By: This method is called from the all_site_report, full_site_report and delta_site_new_scan methods in ticket_service.rb to send new tickets to the Remedy service after they have been discovered and prepared.

update_tickets

- Description: This method sends a list of prepared tickets to the Remedy instance to update existing tickets. If no tickets are passed into the method, then there are no tickets to update. When updating tickets, the method first creates the Savon client for sending tickets by calling the get_client method, passing in the wsdl endpoint and the SOAP modify endpoint. It then calls the send_tickets method, with the newly created client, the submit service and the ticket array as parameters to send the tickets to Remedy.
- Used By: This method is called from the delta_site_new_scan method in ticket_service.rb when in IP and vulnerability mode to update existing tickets in Remedy.

close_tickets

- Description: This method sends a list of prepared ticket closures to the Remedy instance. If not tickets are passed into the method, then there are no tickets to be closed. When updating the ticket status, the method first creates the Savon client for sending tickets by calling the get_client method, passing in the wsdl endpoint and the SOAP modify endpoint. It then calls the send_tickets method, with the newly created client, the submit service and the ticket array as parameters to send the tickets to Remedy.
- Used By: This method is called from the delta_site_new_scan method in ticket_service.rb to close existing tickets. It is used in all 3 ticket generation modes.

query_for_ticket

- Description: This method sends a query in SOAP format to Remedy to return a single ticket based upon the criteria, usually the NXID of the ticket. First the method creates the Savon client using the get_cleint method, passing in the wsdl endpoint and the SOAP modify endpoint. The client is then called, passing in the unique identifier and a query string requesting any tickets not in the closed state. The method then returns the ticket information in a hash, or nil if no results are found.
- Used By: This method is used by the prepare_tickets and prepare_close_tickets method to query the Remedy instance for an existing ticket matching the provided criteria.

prepare_create_tickets

- Description: This method is called to choose the correct 'matching fields' for the current ticketing mode before calling the prepare_tickets method. This value is used to group related vulnerability information together when creating tickets. The matching fields value is chosen based upon the current ticketing mode and how the information is to be grouped per ticket: Individual vulnerability to IP for Default mode; by Individual IP for IP mode; and by Vulnerability for Vulnerability mode.
- Used By: This method is called from the all_site_report, full_site_report and delta_site_new_scan methods in ticket_service.rb to prepare and format new tickets for sending to the Remedy service.

prepare_update_tickets

- Description: This method is called to choose the correct 'matching fields' for the current ticketing mode before calling the prepare_tickets method. This value is used to group related vulnerability information together when creating tickets. The matching fields value is chosen based upon the current ticketing mode and how the information is to be grouped per ticket: by Individual IP for IP mode; and by Vulnerability for Vulnerability mode.
- Used By: This method is called from the delta_site_new_scan method in ticket_service.rb when in IP and vulnerability mode.

prepare_tickets

- Description: This method is called to prepare a list of vulnerabilities, converting them into the correct ticket format for sending to the Remedy instance. This method can handle both new and existing tickets by calling the query_for_ticket method, passing in the NXID for the ticket. If the ticket already exists, then the method converts the retrieved incident into a new ticket object and will update the ticket object. If the ticket doesn't exist, the method will create a new ticket object instead. Using the matching fields variable, this method groups related rows from the CSV file together into a single ticket. For each row corresponding to a new ticket in the CSV file, a new ticket object will be created with the correct values and description. For every subsequent row that is part of the same ticket, the ticket description will be updated by the common_helper.rb class. When a new matching field value is encountered, the previous ticket is placed in the array of created tickets, before creating a new ticket for the current row. This array of tickets is then returned to the parent method.
- Used By: This method is called from the prepare_create_ticket and prepare_update_tickets methods, to prepare a list of tickets to send to Remedy.

ticket_from_queried_incident

- Description: This method converts a queried remedy incident (ticket) into a Savon formatted ticket for editing and resending to Remedy. This method takes the original ticket to update as the initial argument. When updating the ticket information, the new "Notes" are passed as the second argument, whereas to close or change the status of the ticket, the notes are left as nil and the new status value is passed in.
- Used By: This method is called from the extract_queried_incident method when updating a ticket and directly from the prepare_close_tickets method when preparing to send a ticket closure.

extract_queried_incident

- Description: This method extracts from a queried Remedy incident the relevant data required for an update to be made to said incident. It then creates a ticket with the extracted data.
- Used By: This method is called from the prepare_tickets method to populate the fields of a newly created ticket with required information, before inserting updated data into the notes field for resending to Remedy.

prepare_close_tickets

- Description: This method prepares a list of ticket closures from the CSV of vulnerabilities exported from Nexpose, to send to the Remedy instance. For each row of the CSV, the query_for_ticket method queries Remedy, passing in the NXID to get the existing ticket. Then, using the ticket_from_queried_incident method, the tickets status is changed to closed.
- Used By: This method is called from the delta_site_new_scan method in ticket_service.rb to change the status of closed tickets for sending to Remedy. It is used in all 3 ticket generation modes.