Nexpose Ticketing System Integration Guide for JIRA

Contents

Last Revised: November 19, 2015

RAPID7

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 JIRA platform, said tickets can be assigned to work teams, prioritized and resolved.

The JIRA integration performs scans of sites or tags, depending on the configuration, and then creates tickets either for each machine or vulnerability, depending on the ticketing mode selected. On 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 JIRA, 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 JIRA service.

Installation and Configuration

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

Partner Product Configuration

Once all dependencies have been installed, the configuration files need to be edited with the details of the target Nexpose and JIRA clients. To insert the details, 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.

Within JIRA, a valid username and password pair should be created, along with a JIRA project. All tickets opened will be under the “Task” type.

The sites or tags which are to be scanned, 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 jira.config file. The JIRA config file is a YAML document with the following options:

Setting	Description	Sample Value
helper_name	This is the helper class name. This should not be changed.	JiraHelper
jira_url	The URL of the users JIRA page. Replace 'url' with the users base URL.	https://url/rest/api/2/issue/
username	The username of the JIRA account.	JBloggs
password	The password of the JIRA account.	pWord
project	This is the Key of the project in JIRA under which the tickets will be generated.	ProjectKey

Initial Run

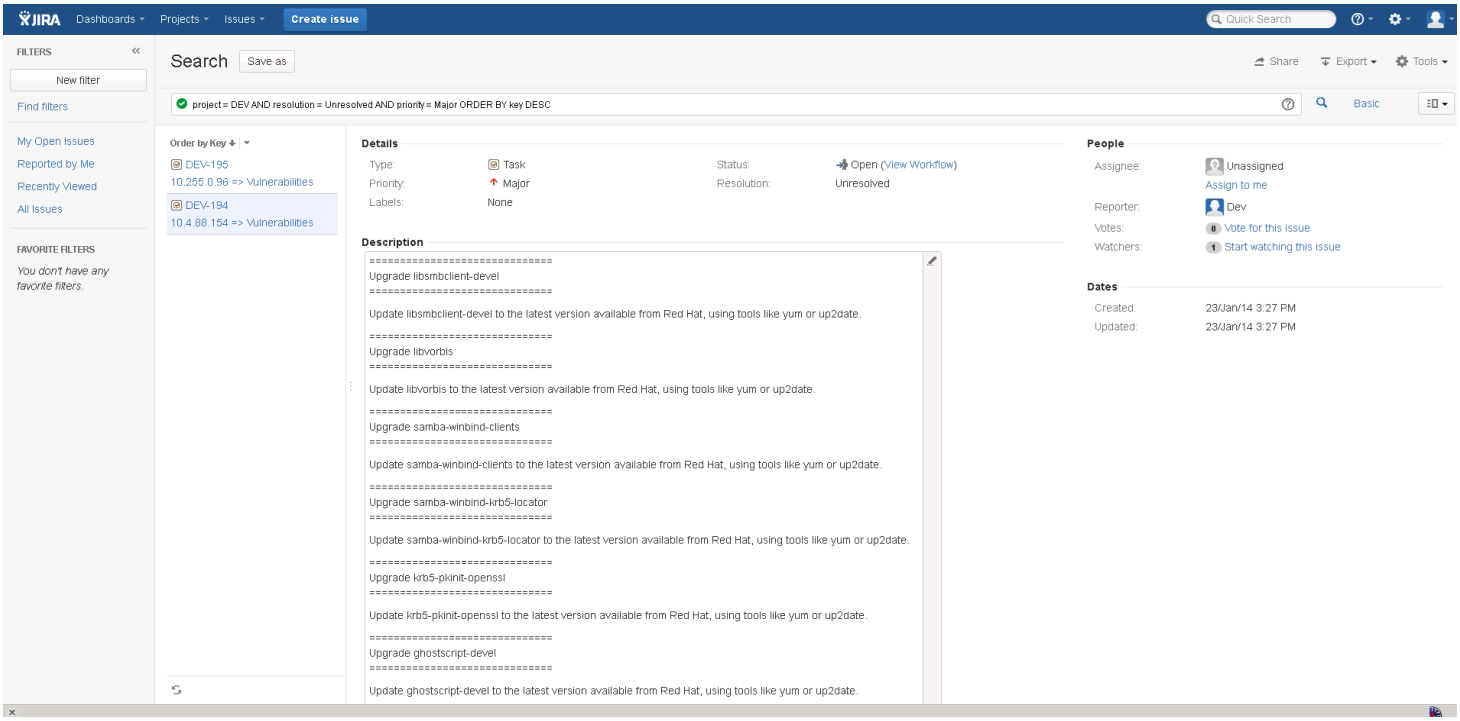
Assuming you've properly configured the Nexpose and JIRA parameters, issue the command:

- ruby nexpose_jira

from a command / bash shell within the 'bin' folder of the Gem. 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 **/var/lib/nexpose_ticketing/log** directory.

As part of a continuous ticketing program, the command may be run 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 a JIRA user without sufficient permissions to create tickets.
- 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.

To assist with troubleshooting, log files are contained within the **/var/lib/nexpose_ticketing/log** folder.

If everything still fails, please send an email to integrations_support@rapid7.com with the ticket_helper.log and ticket_service.log attached and a description of the issue

Helper Method Overview

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

get_jira_key

- **Description**: This method is used to fetch the unique JIRA key for an existing ticket in a project. Using the provided query (containing project ID, nexpose ID and closure_step_id) it uses a HTTP GET to request and return the correct ID
- **Used By**: This method is used by the prepare_close_tickets and prepare_update_tickets to find the corresponding JIRA key for existing tickets.

get_jira_transaction_details

- **Description**: This method fetches the JIRA ticket transition details for the given JIRA ticket key. This is used to discover valid transitions between different ticket states in JIRA. It will try to match the provided response to the desired transition as specified in the JIRA configuration file.
- **Used By**: This method is used by the close_tickets method to allow completed tickets to be correctly transitioned to the user closed state.

prepare_create_tickets

- **Description**: This method is used 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. This is based upon 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 tickets for sending to the JIRA service. It is also called by the prepare_update_tickets method to prepare all new or existing tickets for sending to JIRA.

create_tickets

- **Description**: This method is used to send new tickets to the JIRA project. An array of prepared tickets to create is provided to the method. It then creates a new HTTP POST request for each ticket, using the data from the jira.config file, before sending to the user specified URL.
- **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 JIRA service after they have been discovered and prepared.

prepare_tickets

- **Description**: This method prepares a list of vulnerabilities from Nexpose, transforming them into ticket format for the JIRA service. Using the matching fields variable, selected in the prepare_create_tickets method, 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 group of rows is encountered, the current object is converted into JSON and appended to the array of ticket

objects. The row is then used as the basis for the next ticket. This array of tickets is the returned to the parent method.

- **Used By**: This method is called by the prepare_create_tickets method to parse a CSV file of vulnerabilities into individual tickets based on the user chosen ticketing mode.

prepare_close_tickets

- **Description**: This method uses a provided CSV file to query JIRA for the corresponding JIRA key. For each row in the file the method calls get_jira_key, passing in a query containing the project number and the NXID of the issue. The returned key is placed in an array with its corresponding ticket before being returned.
- **Used By**: This method is called from the delta_site_new_scan method in ticket_service.rb to gather keys of resolved tickets for closing in the JIRA service. It is used in all 3 ticket generation modes.

close_tickets

- **Description**: This method is for sending ticket closure messages in JSON format to the JIRA service. A list of valid JIRA ticket keys are provided to the method, with each key representing a ticket completed before the last scan. For each ticket, the method calls get_jira_transition_details, requesting the steps to get from the tickets current state into the user selected 'closed' state. A HTTP request is then sent to JIRA to transition the selected ticket into the 'closed' state.
- **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.

prepare_update_tickets

- **Description**: This method prepares ticket updates from a provided CSV list of vulnerabilities. JIRA uses a ticket key to push updates, so first all the tickets are prepared using the prepare_create_tickets method, allowing new tickets to be sent along with the updates. For each created ticket the method then tries to get the corresponding JIRA key using get_jira_key. This key, either as a value or nil, is put into a key value pair along with the generated ticket. This pair is then appended to an array of tickets to send. The nil value allows the update_ticket method correctly send the whole ticket, rather than just updating the ticket description.
- **Used By**: This method is called from the delta_site_new_scan method in ticket_service.rb when in IP and vulnerability mode.

update_tickets

- **Description**: This method is for sending ticket updates in JSON format to the JIRA service. Each ticket in the provided list is a key value pair representing its JIRA key and the corresponding newly generated ticket description. If a ticket is to be updated, then the JIRA key is appended to the URL, and the request body contains an update message in JSON format containing the new description for the ticket. This is then pushed to the service. If the ticket does not yet exist (no corresponding JIRA key) then this is considered a new ticket. The request body instead contains the whole ticket, which is then sent to the service. These tickets are sent as individual HTTP POST requests, using the settings and user specified URL from the jira.config file.
- **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 JIRA.