Integration Guide

RAPID7 Nexpose Ticketing Gem Developer Guide

Website:	https://github.com/rapid7/nexpose_ticketing
Support:	integrations_support@rapid7.com
Version:	0.0.1
Type:	Ruby

Last Revised: February 2nd, 2014

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. Using the Nexpose Ticketing Gem will allow users to implement their own integration endpoint with a ticketing system.

Gem Configuration

Introduction

This document will guide you through all the technical steps necessary to configure and develop a Ticketing integration using the Nexpose Ticketing Gem service.

Before you begin

The integration was created using Ruby, as such, a Ruby interpreter must be installed on the system where it's going to run. The following link shows the different options for installing Ruby in several platforms:

https://www.ruby-lang.org/en/downloads/

Please install the codebase that best suits your environment.

Once installed, the next step would be to check out the source code from GitHub, at:

https://github.com/rapid7/nexpose_ticketing

Directory Structure

The source code has the following structure:

```
\---nexpose_ticketing

    |  .gitignore                                  gitignore File.

    |  nexpose_ticketing.gemspec                   Gemspec the ticketing system.

    |  README.md                                   README File.

    |

    +---bin

    |     nexpose_jira                             JIRA Implementation.

    |

    +---lib

    |   |  nexpose_ticketing.rb                    Main Interface with the service.

    |   |
    |   \---nexpose_ticketing
    |       |   queries.rb                              SQL Queries.
    |       |   ticket_repository.rb                    Report Generator.
    |       |   ticket_service.rb                       Ticketing Service.
    |       |
    |       +---config
    |       |       jira.config                         JIRA Configuration File.
    |       |       ticket_service.config               Ticket Service Config.
    |       |
    |       \---helpers
    |               jira_helper.rb                      JIRA Helper Implementation.
    |
    \---test                                            Test Files below.
        |   data_report.csv
        |   empty_report.csv
        |   nexpose_ticketing_spec.rb
        |   sample_historical_scan_data.csv
        |   ticket_repository_spec.rb
        |   two_vulns_report.csv
        |
        \---helpers
                jira_dummy_one_ip_ticket.txt
                jira_dummy_two_default_tickets.txt
                jira_helper_spec.rb
```

Service Workflow.

Any new implementation should have three important files:

- **Helper File:** This file should implement the methods to transform a CSV Report into the appropriate format. For example, the JIRA service takes in a JSON object so a CSV -> JSON Translation must be performed.

- **Config File:** The configuration file defines anything implementation specific such as username, password, URL/IP needed by the Helper to create a ticket. **NOTE**: The Field marked as 'helper_name' is mandatory and should have the name of the Helper File class. This is used dynamically at service startup.

- **Executable File:** This file should read the Config File and call the main NexposeTicketing main class with the configuration options.

JIRA Example.

In the case of the JIRA Implementation the three files are explained bellow:

- **Helper File jira_helper.rb:**

  This file implements three main methods:

  1. Initialize. This is the constructor that'll take the implementation options and the service options.

  2. create_ticket(tickets). This method will take in an array of tickets previously created and "send" them to the ticketing system. In the case of JIRA, the transport is through a HTTPS POST Method.

  3. prepare_tickets(vulnerability_list). This method will delegate to two other methods depending on the configuration file a vulnerability_list (CSV Report) called from the service.

     a. prepare_tickets_default(vulnerability_list). This method should parse the CSV report to the format that the ticketing system uses. In the case of JIRA this format is in JSON.

        The 'default' method means that there should be a ticket per IP and Vulnerability. For example, if on a scan of IP 1 we find vulnerability A, B, C. Three tickets are created: IP 1 -> A, IP 1 -> B, IP 1 -> C.

     b. prepare_tickets_by_ip (vulnerability_list). Likewise, this method should parse the CSV report to the format that the ticketing system uses. The different with default is that in this case we're grouping by IP. Using the previous example there would only be one ticket created: IP 1 grouping A, B, C Into one ticket.

     Please refer the jira_helper.rb file for an example of the implementation.

- **Configuration File jira.config.**

The configuration file defines the entire information specific with the Helper class. The jira.config file has:

```
# (M)) Helper class name.

:helper_name: JiraHelper

# Optional parameters, these are implementation specific.

:jira_url: https://url/rest/api/2/issue/

:username: jirausername

:password: jirapassword

:project: projectname
```

The **mandatory** option is the class name for the helper under helper_name. This name is used to load dynamically the developed helper. The other fields are implementation specific, in the case of the JIRA example, we require a URL, username, password and project. Remember, these variables are only used by the Helper file you develop so they are very implementation specific.

- **Executable file: nexpose_jira.**

The executable file is a simple file that reads the Configuration File, parses it into a Hash of variables and then initializes the Ticket service with the data calling the method start: NexposeTicketing.start (jira_options)

Please refer to the Executable file nexpose_jira under the bin folder for an example.

Packaging it all up.

Where the files are saved is very important; once the gem is installed, any development done should be saved to the specific folders:

- **Executable file:** The executable file should be saved to the bin folder.
- **Helper file:** All helper files reside under the /lib/helper directory and the service expects to load the helper file from there.
- **Configuration file:** The configuration file should reside under /config folder.

All these folders can be found usually under the following paths:

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

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

Running for the first time

Assuming you've properly configured the Nexpose and your helper parameters, type in the executable file created from a command/bash shell. Every time this command is run the service will query Nexpose and obtain any new vulnerability information and open tickets accordingly.

What if something goes wrong?

The most common errors when running the script are configuration based, users without permission to create tickets or generate reports with Nexpose, incorrect passwords or not specifying a site or asset group when configuring the script.

We recommend reading the log file under the log folder in the Gem.

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