Integration Guide

ServiceNow <-> Nexpose Integration

Vulnerability -> Ticket workflow

Partner Name:	ServiceNow
Website:	http://www.servicenow.com
Product Name:	ServiceNow Incident
Version:	Darwin
Import Type:	Automated via API

Last Revised: November 10, 2013

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 ticket incidents based on vulnerabilities found across their systems. With this information in their ServiceNow platform, said tickets can be assigned to work teams, prioritized and resolved.

Partner Product Configuration

The JSON Web Service Module needs to be activated within ServiceNow before deployment of the Integration script. As of November 2013, the following steps are required to activate it:

- Log into your ServiceNow instance using administrator credentials.
- Navigate to **System Definition > System Plugins.**
    - Right-click the JSON plugin name on the list and select **Activate/Upgrade**
- Click **Activate**

As always, check ServiceNow documentation for further information.

Introduction

This document will guide you through all the steps necessary to configure the ServiceNow script to successfully import Nexpose vulnerability data into ServiceNow incident ticketing system.

Before you begin

The script 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 most appropriate for your need.

Once installed, the following Ruby Gems must also be installed:

- nexpose (http://rubygems.org/gems/nexpose)
- json (http://rubygems.org/gems/json)
- nokogiri. (http://rubygems.org/gems/nokogiri)

In some systems, dependencies might need to be installed prior to the installation of each Gem. Please refer to their appropriate documentation for instructions.

Configuring the script

Once all dependencies have been installed, the script should now be configured. To configure, open the script with a text editor (Notepad, Notepad++, etc)

- Configure ServiceNow settings:

```
# SEVICENOW CONFIGURATION
# To use this script the JSON web service module must be activated at
#   System Definition -> System Plugins -> Search for JSON -> Activate/Upgrade

# ServiceNow username
@@user = 'admin'

# ServiceNow password
@@password = 'admin'

# ServiceNow URL
# Example: @@serviceNowURL = "https://yourinstance.service-now.com/incident.do?JSON"
@@serviceNowURL = "https://yourinstance.service-now.com/incident.do?JSON"
```

  - Define the username, password, and URL for your particular ServiceNow instance; we recommend creating a user with permissions to create tickets solely for this purpose.

- Configure verbose SSL settings:

```
# Verbose connection mode? If yes, SSL connections will output to STDERR
# Example @@verbose_mode = 'Y'
@@verbose_mode = 'N'
```

  - If set to 'Y' (Yes) all connection information between the script and the ServiceNow instance will output to $STDERR. Leave 'N' as default.

- Configure Nexpose settings:

```
# NEXPOSE CONFIGURATION

# Nexpose username
# Example: @@nxuser = 'nxadmin'
@@nxuser = 'nxadmin'

# Nexpose password
# Example: @@nxpass = 'nxadmin'
@@nxpass = 'nxadmin'

# Nexpose console IP address
# Example: @@console = '127.0.0.1'
@@console = '127.0.0.1'
```

  - A valid username, password and console IP address. We recommend creating a user with permissions to create reports on the sites/asset groups necessary.

```
# Site IDs separated by commas, leave empty for no site.
# Example: @@sites = [01, 02, 06]
@@sites = [01]

# Asset group IDs separated by commas, leave empty for no site. A site OR asset group must be defined.
# Example: @@asset_groups = [02, 05, 08]
@@asset_groups = []
```

- A collection of sites or asset groups to be reported on. At least ONE site or ONE Asset group must be defined. Separate multiple values with commas and a space.

```
# Severity floor level.
# Severity is a integer (non-decimal) number between 0 and 10 inclusive with 8 > being critical level.
# Setting a level also includes the rest of the levels up to 10.
# Example: @@severity_level = 8 also includes level 9 and 10.
@@severity_level = 8
```

- A minimum severity level on a scale from 0 to 10 (inclusive). The higher the number the more critical the vulnerability to report on. Leave an 8 for critical only.

  WARNING, a low number on this setting could translate into a big number of tickets generated, edit with caution.

- Configure Logging settings:

```
# LOGGING.
# This script includes a logger, all output will be sent to the file service_now.log in the directo
# where this script is run.
require 'Logger'
$LOG = Logger.new('service_now.log', 'monthly')

# Valid log levels: Logger::DEBUG Logger::INFO Logger::WARN Logger::ERROR Logger:FATAL
$LOG.level = Logger::INFO
```

- A logger is included with the script that outputs by default all INFO events to the file service_now.log in the directory where the script is run. To configure this setting, feel free to change the logging level or the name of the file to be output in this setting.

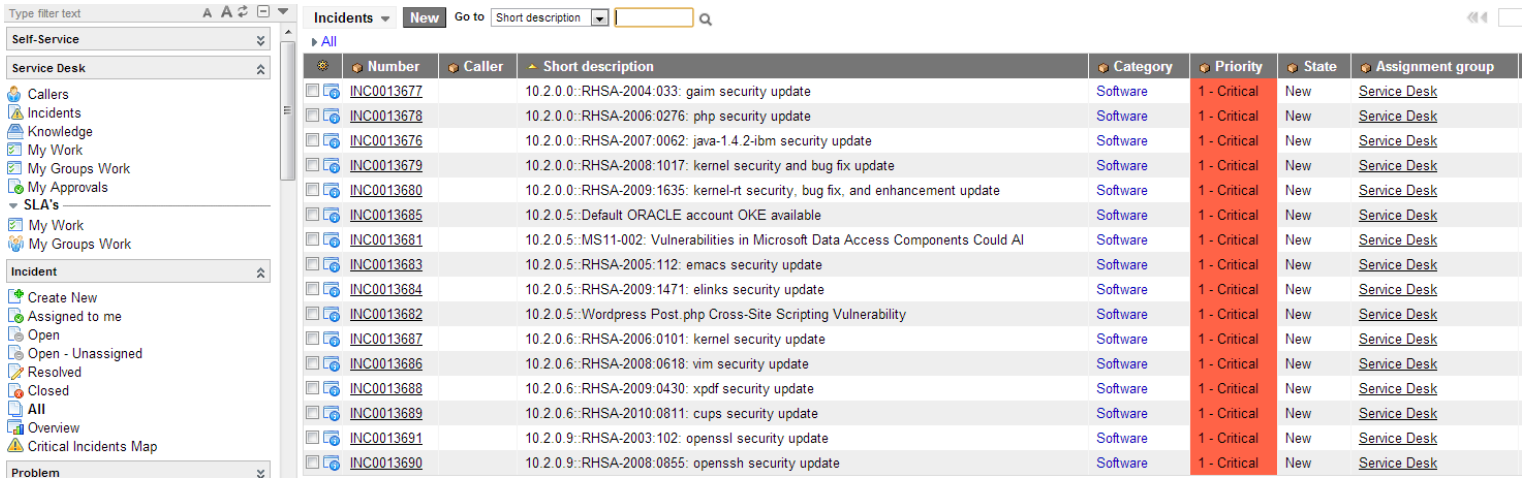
- Run the script for the first time.

  - The script can be run using the command from the command line:

    > ruby service_now.rb

  - While running, all logging information will be output to the service_now.log file.

  - Once finished, tickets should've been created in the ServiceNow instance:



Number	Caller	Short description	Category	Priority	State	Assignment group
INC0013677		10.2.0.0::RHSA-2004:033: gaim security update	Software	1 - Critical	New	Service Desk
INC0013678		10.2.0.0::RHSA-2006:0276: php security update	Software	1 - Critical	New	Service Desk
INC0013676		10.2.0.0::RHSA-2007:0062: java-1.4.2-ibm security update	Software	1 - Critical	New	Service Desk
INC0013679		10.2.0.0::RHSA-2008:1017: kernel security and bug fix update	Software	1 - Critical	New	Service Desk
INC0013680		10.2.0.0::RHSA-2009:1635: kernel-rt security, bug fix, and enhancement update	Software	1 - Critical	New	Service Desk
INC0013685		10.2.0.5::Default ORACLE account OKE available	Software	1 - Critical	New	Service Desk
INC0013681		10.2.0.5::MS11-002: Vulnerabilities in Microsoft Data Access Components Could Al	Software	1 - Critical	New	Service Desk
INC0013683		10.2.0.5::RHSA-2005:112: emacs security update	Software	1 - Critical	New	Service Desk
INC0013684		10.2.0.5::RHSA-2009:1471: elinks security update	Software	1 - Critical	New	Service Desk
INC0013682		10.2.0.5::Wordpress Post.php Cross-Site Scripting Vulnerability	Software	1 - Critical	New	Service Desk
INC0013687		10.2.0.6::RHSA-2006:0101: kernel security update	Software	1 - Critical	New	Service Desk
INC0013686		10.2.0.6::RHSA-2008:0618: vim security update	Software	1 - Critical	New	Service Desk
INC0013688		10.2.0.6::RHSA-2009:0430: xpdf security update	Software	1 - Critical	New	Service Desk
INC0013689		10.2.0.6::RHSA-2010:0811: cups security update	Software	1 - Critical	New	Service Desk
INC0013691		10.2.0.9::RHSA-2003:102: openssl security update	Software	1 - Critical	New	Service Desk
INC0013690		10.2.0.9::RHSA-2008:0855: openssh security update	Software	1 - Critical	New	Service Desk

Recommendations

We recommend running the script after scans have been completed and previous tickets have been closed. If a ticket has not been closed (and the vulnerability still exists) the script will create two tickets with the same information. In any case, further modifications can be made in regards of the number of sites / asset groups to be reported on. In some scenarios, having multiple copies of the script with different sites/asset groups running in a (cron) job at different times could be the best option.

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 setting the logging mechanism to ERROR and retrying, in most cases the issue will present itself during execution of a specific step (for example, a connection error could mean the script cannot connect to Nexpose/ServiceNow).

If everything still fails, please send an email to integrations-support@rapid7.com with the service_now.log attached and a description of the issue.