IBM Security

IBM

Resilient SOAR Platform

Resilient Machine Learning Function

User Guide

V1.0

Date: February 2020

Resilient SOAR Platform
Resilient Machine Learning Function
User Guide

Platform Version	Publication	Notes
1.0	February 2020	Initial publication.

Table of Contents

Overview

Resilient Functions simplify development of integrations by wrapping each activity into an individual workflow component. These components can be easily installed, then used and combined in Resilient workflows. The Resilient platform sends data to the function component that performs an activity then returns the results to the workflow. The results can be acted upon by scripts, rules, and workflow decision points to dynamically orchestrate the security incident response activities.

This is a user guide for the Resilient Machine Learning Function (fn_resilient_ml). It is a companion guide to the *Resilient Machine Learning Function Reference Guide*.

Resilient Machine Learning function supports Nature Language Processing (NLP). NLP is used to digest textual data of incidents and provide advanced predictions of incident relationships. This tool is integrated with Resilient platform and customized for Resilient users.

This integration provides the followings:

- A command line tool to build an NLP model
- A search function that ranks incidents according to similarity

This package also includes an example workflow that demonstrates how to call the function, two rules that starts the workflow, and a custom data table the function uses to write the results.

Installation and Configuration

Before downloading the Resilient Machine Learning Function, perform the following steps to make sure that your Resilient environment meets the prerequisites:

1. Make sure your Resilient platform is version 32 or later.
2. Make sure you have a dedicated Resilient account to use for your integration. If it is a user account, it must have the permissions to view and modify administrator and customization settings, and read and update incidents. If using an API key account, it must have the permissions to read incidents, and edit incident fields and notes.
3. Make sure you have installed and configured a Resilient integration server, as described in the Resilient SOAR Platform Integration Server Guide.
4. Make sure that the integration server is running Resilient Circuits version 32 or later, Python version 3.6 or later, and "pip".
5. Run the following command to install pandas on the integration server:

   ```
   sudo pip install pandas-0.23.4-cp27-cp27m-linux_x86_64.whl
   ```

   This step is necessary because there is no wheel for pandas on Redhat Enterprise 7 available from pip, and Redhat Enterprise 7 does not have gcc/g++ installed, so pip cannot build it from source code. A wheel was built locally, using the source code of https://files.pythonhosted.org/packages/e9/ad/5e92ba493eff96055a23b0a1323a9a803af71ec859ae3243ced86fcbd0a4/pandas-0.23.4.tar.gz. This is the same source code as the "pip install pandas" downloads.

6. Download and install the Resilient Machine Learning Function package as described in the Resilient SOAR Platform Integration Server Guide.
7. Open the app.config file and edit the following settings in the [fn_machine_ml] section. Note that the first one is required. The other two already have default values.

   ```
   model_path=path to the folder you are going to save your model files
   num_top_similar_incidents=5
   num_features=50
   ```

   The settings in [fn_machine_ml] section define how the function locates the saved machine learning model used for prediction.

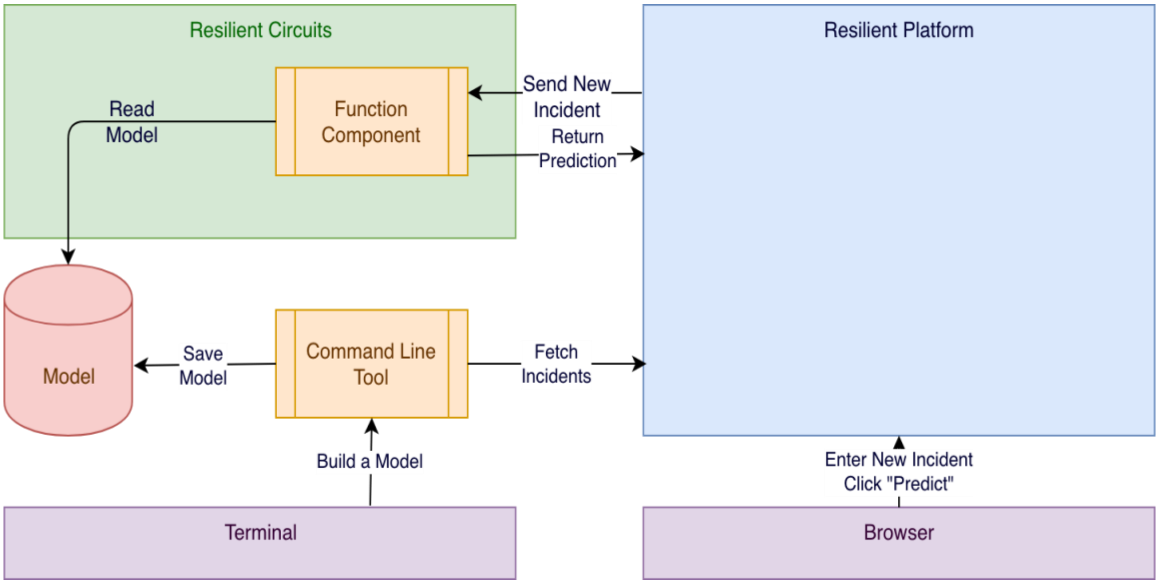
model_path	Absolute path to the folder that contains the saved NLP model.
num_top_simillar_incident	Number of similar incidents to return (ranked by similarity).
Num_features	Number of features for the word2vec model.

Please refer to the *Resilient Machine Learning Reference Guide* for more information about Resilient Machine Learning.

Function Descriptions

This integration package contains two components.

- A command line component to build an NLP model.
- A function component to compute and return incident similarity using the NLP model.



As shown in the above graph, users use the command line tool from a terminal to build an NLP model. The command line tool connects to the specified Resilient platform to fetch incidents and artifacts. These are the samples used to train an NLP model. The model is then saved into files.

The function component takes an incident ID, retrieves the incident using the ID, and loads the saved model from the folder specified in app.config. It can then use the model to compute similarities for this incident. This is triggered by a menu item in the incident page.

In summary, there are two processes, one to build an NLP model, the other to compute similarities using a model.

Build a Machine Learning Model

The command line tool is used to build an NLP model. It contains 2 subcommands.

build_nlp

This subcommand is used to build an NLP model. It downloads incidents and artifacts from a Resilient platform, and then saves them into two CSV files. Then it uses the data to build an NLP model. Those two CSV files are deleted once a model is built. The Resilient platform is specified in the app.config file. The subcommand has the following parameters:

-i	Optional. CSV file with incident data. Download incidents if this is absent.
-a	Optional. CSV file with artifact data. Download artifacts if this is absent.
-o	Optional. Model name. Default to be “resilient” if absent.

Example:

```
res-ml build_nlp
```

By default, an NLP model is saved into the following files in the folder where you run this command:

- resilient-w2v.txt
- resilient-sif.pkl

- resilient-pca.json
- resilient-vec.json

view

This subcommand is used to view basic information of one of the four files above.

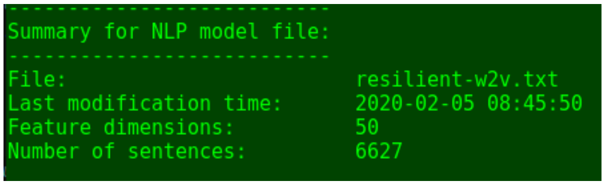
It takes one flag:

-i Required. File name

Example:

```
res-ml view -i resilient-w2v.txt
```

It shows the summary of the saved model, including last modification time, number of features (number of dimensions), and sample count.

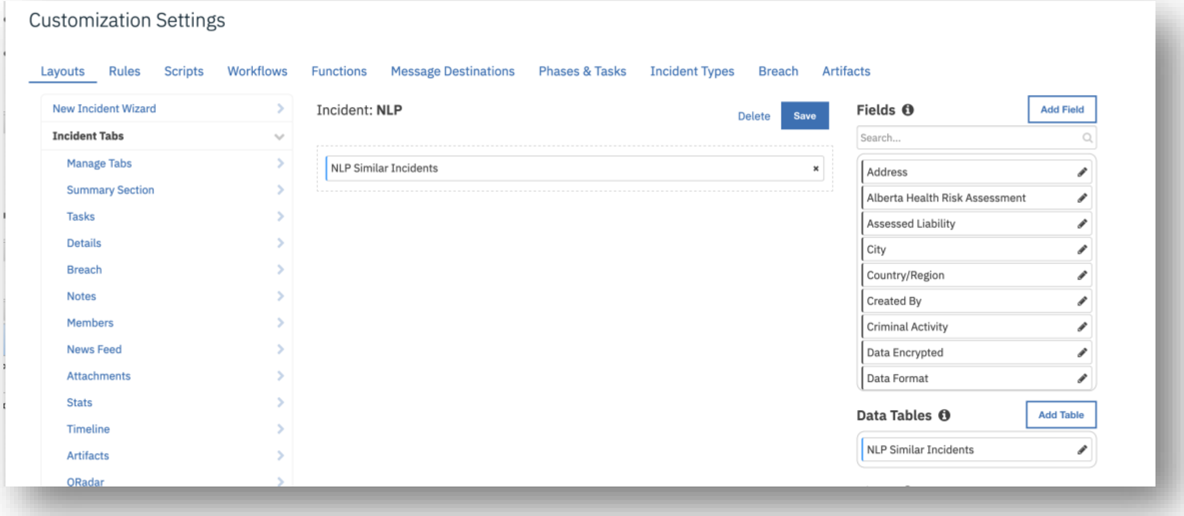


```
---------------------------
Summary for NLP model file:
---------------------------
File:                        resilient-w2v.txt
Last modification time:      2020-02-05 08:45:50
Feature dimensions:          50
Number of sentences:         6627
```

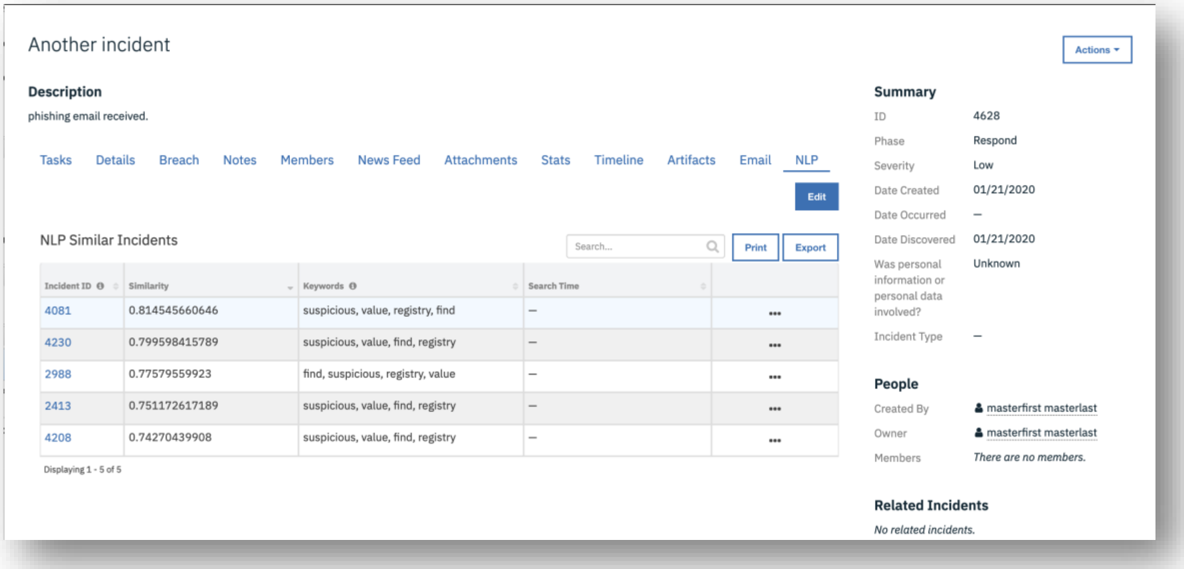
Use an NLP model to do a search

Once a machine model is built and saved, it can be used to find similar incidents by the function component. The function component locates the saved model to use by looking at "model_path" which is set in app.config. The four files listed previously for a saved NLP model need to be in this folder.

The sample workflow writes the result into a custom data table called "NLP Similar Incidents." To show the data table in an incident page, you need to add it to the incident layouts. For example, you can create a new tab called NLP and add the data table to the layout. For details on incident layouts, see the *Resilient SOAR Platform Playbook Designer Guide*.



The new NLP tab and the custom field is shown in the incident page.



Description

phishing email received.

NLP Similar Incidents

Incident ID	Similarity	Keywords	Search Time	
4081	0.814545660646	suspicious, value, registry, find	–	•••
4230	0.799598415789	suspicious, value, find, registry	–	•••
2988	0.77579559923	find, suspicious, registry, value	–	•••
2413	0.751172617189	suspicious, value, find, registry	–	•••
4208	0.74270439908	suspicious, value, find, registry	–	•••

Displaying 1 - 5 of 5

Summary

ID	4628
Phase	Respond
Severity	Low
Date Created	01/21/2020
Date Occurred	–
Date Discovered	01/21/2020
Was personal information or personal data involved?	Unknown
Incident Type	–

People

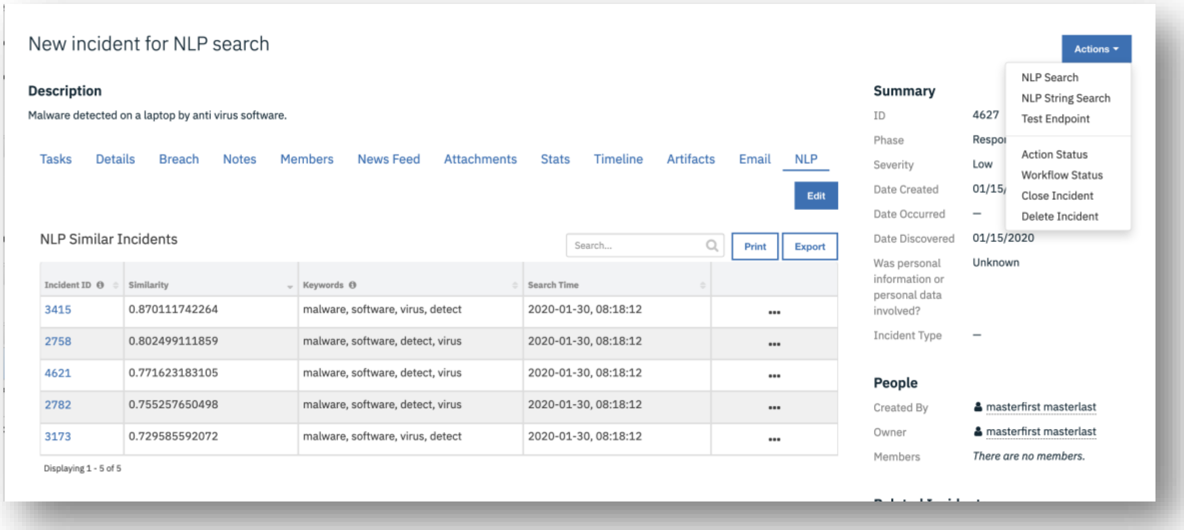
Created By	masterfirst masterlast
Owner	masterfirst masterlast
Members	*There are no members.*

Related Incidents

No related incidents.

Once you build an NLP model, you can use it to find similar incidents. Create a new incident and enter name and description. This information is used to do an NLP search.

From the incident Actions menu, select “NLP Search”.



New incident for NLP search

Description

Malware detected on a laptop by anti virus software.

NLP Similar Incidents

Incident ID	Similarity	Keywords	Search Time	
3415	0.870111742264	malware, software, virus, detect	2020-01-30, 08:18:12	•••
2758	0.802499111859	malware, software, detect, virus	2020-01-30, 08:18:12	•••
4621	0.771623183105	malware, software, detect, virus	2020-01-30, 08:18:12	•••
2782	0.755257650498	malware, software, detect, virus	2020-01-30, 08:18:12	•••
3173	0.729585592072	malware, software, virus, detect	2020-01-30, 08:18:12	•••

Displaying 1 - 5 of 5

Summary

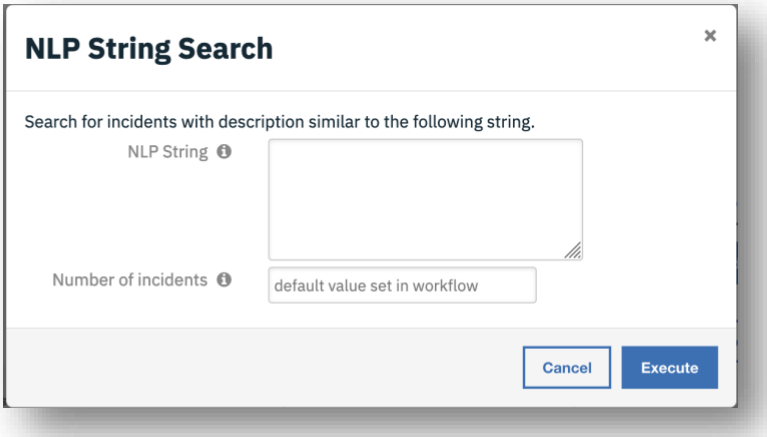
ID	4627
Phase	Respo
Severity	Low
Date Created	01/15/
Date Occurred	–
Date Discovered	01/15/2020
Was personal information or personal data involved?	Unknown
Incident Type	–

People

Created By	masterfirst masterlast
Owner	masterfirst masterlast
Members	*There are no members.*

Wait for the integration to finish its job. The results are shown. Note that you can click the “Similarity” column to force the table to sort according to similarity.

As you can see from the Actions menu, there is another menu item for “NLP String Search”. This is slightly different from the “NLP Search” above. For “NLP Search”, the function uses the name and description of the corresponding incident to do the NLP search. For “NLP String Search”, users can input a string for the NLP search. Thus, users can copy a portion of the description into the following text field and do an NLP search. This menu item can be useful if an incident contains some irrelevant information. Also shown in the image below, users can specify how many incidents to return.



NLP String Search

Search for incidents with description similar to the following string.

NLP String

Number of incidents: default value set in workflow

Rebuild an NLP model

When more incidents are created after an NLP model is built, users shall rebuild the model making use of the newly available data. The process is the same as building a new model. In general, users can rebuild the NLP model once a week.

Troubleshooting

There are several ways to verify the successful operation of a function.

- Resilient Action Status

  When viewing an incident, use the Actions menu to view Action Status. By default, pending and errors are displayed. Modify the filter for actions to also show Completed actions. Clicking on an action displays additional information on the progress made or what error occurred.

- Resilient Scripting Log

  A separate log file is available to review scripting errors. This is useful when issues occur in the pre-processing or post-processing scripts. The default location for this log file is: `/var/log/resilient-scripting/resilient-scripting.log`.

- Resilient Logs

  By default, Resilient logs are retained at `/usr/share/co3/logs`. The `client.log` may contain additional information regarding the execution of functions.

- Resilient-Circuits

  The log is controlled in the `.resilient/app.config` file under the section `[resilient]` and the property `logdir`. The default file name is `app.log`. Each function creates progress information. Failures show up as errors and may contain python trace statements.

For errors related to building a machine learning model, please refer to *Resilient Machine Learning Reference Guide* for more details.

Support

For support, visit https://ibm.com/mysupport.

Including relevant information from the log files will help us resolve your issue.