Spirion Extensions

Spirion Automation: Useful Patterns for Supplementing the Sensitive Data Engine

Table of Contents

Spirion Automation: Useful Patterns for Supplementing the Sensitive Data Engine

Introduction

Spirion's Sensitive Date Engine leverages AnyFind technology coupled with a powerful conditional engine in order to customize results to suit customer's needs. While very effective for finding many common types of PII, AnyFinds don't typically return organization specific IDs, intellectual property, or patterns/codes. To cover this gap, Spirion has implemented a Perl Compatible Regular Expression (PCRE) library.

PCRE is a powerful method of returning pattern matches in text streams. Resources for creating PCRE library compatible regexes are common on the web.

This extension contains several patterns Spirion Technical Staff have created and used over the years. Earlier versions of these patterns have shipped in Sensitive Data Manager, but do not currently ship with Sensitive Data Platform. While they are not as accurate as Spirion's AnyFinds, we hope you find them useful, and look forward to receiving feedback for improvement.

For more information on using Regular Expressions with Spirion, see: Create Custom Types using Regular Expressions (RegEx) – Spirion Support Portal

Released July 2023, © 2023 Spirion LLC.

Requirements

Before working on the steps outlined in this document, please confirm the following:
- The Spirion console (SDM or SDP) is up-to-date and accessible.

Important Notes

Software Versions

SDP console version 22.Q3.1.226.1, SDM console version 11.8.2, and Spirion agent version 12.5 or later are compatible with the steps described in this document.

Related Files

2

These instructions are accompanied by a zip archive called Spirion_CS_Regx–Useful–Patterns.zip

Included Patterns

Pattern: Western Currency

While not terribly useful on its own, this can be combined in Sensitive Data Engine with the CCN or Bank Account AnyFinds, helping to identify statements. It attempts to find various western currency amounts using simple matching. This pattern finds the following patterns with one of the following currency symbols required $ £ € and accommodates for different comma and period separators and decimal points.

Examples:

```
$ 0.99
```

```
$ .09
```

```
£123
```

```
€123,456
```

```
$123,456.99
```

Please note this reference on formatting:
https://www.languageediting.com/format-numbers-eu-vs-us/

Pattern: IPv4

Use this pattern to find IPv4 addresses in network documents etc. It matches tightly and loosely formatted IP Addresses. Only legal IP addresses are allowed, however there is no additional validation built into the pattern itself.

Pattern: IPv6

Use this pattern to find IPv6 addresses in network documents etc. Only legal IP addresses are allowed, however there is no additional validation built into the pattern itself.

Pattern: MAC Address

3

Use this pattern to find MAC addresses in network documents etc. Only addresses are allowed, however there is no additional validation built into the pattern itself.

Pattern: Machine Readable Passport Number

Our Passport AnyFind finds US Passports. This pattern finds the Machine-readable code at the bottom of all passports and is tuned to work with the vagaries that occur with OCR.

Pattern: International Telephone

Our Telephone AnyFind finds US telephone numbers. This pattern finds a variety of international telephone numbers as long as it has a + symbol and [country-code] such as:

Example: +44 (0) 1483 926 095

Although there is no real standard on including "+" vs "0" when writing down a phone number, using plus in our case SIGNIFICANTLY reduces noise and false positives. It is possible to write a variation expression that uses a preceding "0" instead, but would require context validation nearby as a trade-off like t, tel, m, mob, c, cell, etc

Note – The Spirion Telephone Number AnyFind validates the US +1 country code, so when using both the AnyFind and this pattern, US numbers with the +1 country code will result in duplicate matches.

Pattern: Dates

Why would you use this: The Date of Birth AnyFind locates birthdates, however, it is useful to find date patterns with no validation that are easily incorporated into Sensitive Data Definitions. This pattern does a good job of finding the following patterns with commas, spaces, slashes, and slashes. Backreferences are used to make sure the separators are consistent within the date match to reduce false positives

```
M DD YYYY such as 1-18-1981

MM DD YYYY such as 01/18/1981

YYYY AAA DD such as 1981 Jan 01

AAA DD YYYY such as January 1, 1981

AAA DD(st,rd,th) YYYY such as January 1st, 1981 (ordinal date formatting
support is limited)
```

4

Pattern: Birth Dates with simple validation

Why would you use this: The Date of Birth AnyFind locates birthdates, however text validation is tight. This pattern utilizes the above Date pattern and adds simple built-in text validation to eliminate the need to create an SDD for use cases where validation text is within 100 characters before the date match.

Validation words are birth/dob/born and can be easily added to or changed.

Pattern: Expiration Dates with simple validation

Why would you use this: AnyFind locates birthdates, however there are use cases to locate dates of issue or expiry such as identification cards. This pattern utilizes the above Date pattern and adds simple built-in text validation to eliminate the need to create an SDD for use cases where validation text is within 100 characters before the date matches.

Validation words are expir/issue and can be easily added to or changed.

- **%SearchUserName%**: This returns the name of the user who performed the search on the endpoint, e.g. "Administrator".
- **%TaskId%**: This returns the task id under which the search was run. The %TaskId% uniquely identifies a specific task stored in the database. A TaskId can be viewed in a report using the *Searches-Task Identifier* column.
- **%TotalDataTypes%**: This returns a numerical value representing the total number of data types found in the search. e.g. if SSN's and CCN's were found during the search, the value returned would be 2.

Import Instructions

To import the attached Sensitive Data Definitions into Sensitive Data Manager:

1. Download the attached Spirion_CS_Pegx-Useful-Patterns.zip archive file for the Sensitive Data Definition you want to import.
2. Unzip the archive file to extract the specific .xml file.
3. Open your Sensitive Data Manager Console and navigate to the **Admin** tab.
4. On the side menu, select **Sensitive Data Types**.

Service Tasks

Discovery Teams

Sensitive Data Types

External Authentication
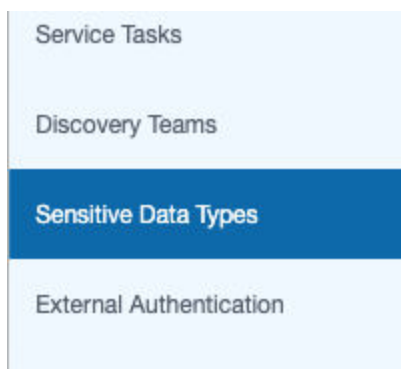
5. On the Ribbon that appears, click **Import**.

Import

6. Browse and select the .xml file for the definition you want to import. The Sensitive Data definition will load.

To import these patterns into SDP, please refer to the following documentation page: Working with Global Data Types (spirion.com)