SEISMIC

2019 Intern Hackathon

Data obfuscation Challenge

Abstract

Data is at the center of all we do here at Seismic and our ability to analyze our customer's data is the key to our success. Our customers entrust us with their most sensitive data and here at Seismic we make securing that data one of our top priorities. Data is used in all areas of our product to verify accuracy, test performance, identify and fix bugs, and to demonstrate the power of our platform to prospective customers. Before that data can be used for any of these purposes it is often necessary to obfuscate that data and remove any sensitive personally identifiable information (PII).

Challenge

Your goal is to implement a data obfuscator capable of producing a sanitized version of customer data. Obfuscation, is a deterministic, irreversible (unlike encryption), transform of data. Your application should read the source dataset, identify sensitive data, and output an obfuscated copy of that data. The following requirements should be met:

1. **Legible** - Obfuscated elements must maintain the formatting of their source. An obfuscated name, for example, should still be a valid and recognizable name and not an unreadable integer or array of bytes. Keep in mind that obfuscated data may be used for customer demonstrations and telling a story about user "7d86bb11248a4a9f6c1a2d5c3494" would be tedious.

2. **Data integrity** - If there are 10 users with the name "John" in the source dataset, then there should be exactly 10 users in the obfuscated dataset with the same name as well. Keep in mind that simply randomizing all the data would destroy the characteristics of the dataset that make it interesting. You job is to maintain as much of the original data as possible while making it impossible to identify the individual to which the data belongs.

3. **Deterministic** - Obfuscation should be repeatable. Executing your obfuscation with the same set of parameters should result in the same output. This is an important feature of any solution as a truly random implementation would be impossible to debug or validate.

Project Details

Participants are welcome to use any programming language or combination of languages to solve the challenge. A valid solution will accept two parameters as inputs; The first is the data source file and the second is the name of the file where obfuscated data will be written.

Completing the challenge in the allotted time will be challenging but the following checklist should help guide your implementation and ensure that everyone completes enough of the task to allow us to evaluate his/her skills:

1. Enumerate the datasets – Before anything can be done, you must be able to read data into your solution.
2. Identify data fields that contain PII and that must be obfuscated.
3. Implement and test an obfuscation algorithm.
4. Repeat step 3 until you have obfuscated all sensitive data.

Personally Identifiable Information (PII)

PII is data that can be used to identify the individual to which the data belongs, either alone or when combined with other fields. Social Security Number is an example of PII that identifies a user. A user's Name is not necessarily PII as names are not unique, but when combined with other fields such as Address and date of birth it would be possible to identify an individual.

Your first task will be to identify the fields or parts of fields that need to be obfuscated while maintaining as much of the original data as possible. For example, a phone number is PII but the area code by itself is not and may contain useful demographic information that should be preserved. For this reason, it is only necessary to obfuscate the last 7 digits of a phone number while preserving the area code.

Be prepared to discuss your decisions to preserver or obfuscate each field.

Data

Datasets of varying size and complexity will be provided to participants via email and will be available in both XML and JSON formats. Datasets range in size from 1 to 500 records. INT-datasets are more complex, international, datasets containing characters from Russian, Chinese, Japanese, Thai, German, French and English charsets.

Data is provided in the following formats:

Json


```
[
  {
    "Id": "09093e0f-d2c7-4a8c-a245-c7960f78811b",
    "Name": "Joseph M. Younkin",
    "Address": "1941 Pin Oak Drive",
    "City": "Davenport",
    "State": "IA",
    "Zip": "52803",
    "SSN": "484-70-4050",
    "Location": "41.573415, -90.470042",
    "Phone": "563-333-3230",
    "CountryCode": "1",
    "DOB": "February 16, 1992",
    "Email": "JosephMYounkin@teleworm.us",
    "UserName": "Endind",
    "Website": "sikaban.com",
    "Company": "Flagg Bros. Shoes",
    "Occupation": "Image designer",
    "HeightCM": "168",
    "WeightKG": "104.0",
    "BloodType": "O+",
    "Vehicle": "2001 Toyota Verossa",
    "Color": "Blue"
  }
]
```


Xml


```
<Users>
  <object>
    <Id>09093e0f-d2c7-4a8c-a245-c7960f78811b</Id>
    <Name>Joseph M. Younkin</Name>
    <Address>1941 Pin Oak Drive</Address>
    <City>Davenport</City>
    <State>IA</State>
    <Zip>52803</Zip>
    <SSN>484-70-4050</SSN>
    <Location>41.573415, -90.470042</Location>
    <Phone>563-333-3230</Phone>
    <CountryCode>1</CountryCode>
    <DOB>February 16, 1992</DOB>
    <Email>JosephMYounkin@teleworm.us</Email>
    <UserName>Endind</UserName>
    <Website>sikaban.com</Website>
    <Company>Flagg Bros. Shoes</Company>
    <Occupation>Image designer</Occupation>
    <HeightCM>168</HeightCM>
    <WeightKG>104.0</WeightKG>
    <BloodType>O+</BloodType>
    <Vehicle>2001 Toyota Verossa</Vehicle>
    <Color>Blue</Color>
  </object>
</Users>
```


Useful Resources

Seismic Engineers – We will be on hand to answer any questions you may have and to help you produce your best work

The Internet – participants can use the internet as a technical resource. Seismic has guest access on their public network

Name Generator - https://www.fakenamegenerator.com/order.php

JSON Parser - https://www.newtonsoft.com/json