lfn-task-3

April 8, 2024

1 Method 1: Using NLP techniques

Anonymizing PII and creating csv file as output

[1]:

```
import csv
import re

# Regular expression to match log entries
log_pattern = re.compile(
    r'(?P<ip>\d+\.\d+\.\d+\.\d+) - - \[(?P<datetime>[^\]]+)\] "(?
 ↪P<method>[A-Z]+) (?P<path>[^ ]+) (?P<version>HTTP/\d\.\d)" (?P<status>\d{3})␣
 ↪(?P<size>\d+) "(?P<referrer>[^"]*)" "(?P<useragent>[^"]+)" (?
 ↪P<responsetime>\d+)'
)

#replacing PII with <ANONYMIZING>
def anonymize_data():
    return "<ANONYMIZED>"

# Iterating through each data entry nd checking for PII
def parse_and_anonymize_line(line):
    match = log_pattern.match(line)
    if match:

        return {
            'ip': anonymize_data(),
            'datetime': match.group('datetime'),  # Keeping datetime as it's␣
 ↪usually not PII
            'method': match.group('method'),
            'path': anonymize_data(),  # Paths can sometimes contain PII as it␣
 ↪can contain username
            'version': match.group('version'),
            'status': match.group('status'),
            'size': match.group('size'),
            'referrer': anonymize_data(),  # Referrer URLs can contain␣
 ↪sensitive information
            'useragent': anonymize_data(),  # User agents can be considered as␣
 ↪PII
```

```
            'responsetime': match.group('responsetime')
        }
    return None

# Process log file and write anonymized data to CSV
def process_log_file(input_file_path, output_file_path):
    with open(input_file_path, 'r') as infile, open(output_file_path, 'w',␣
 ↪newline='') as outfile:
        fieldnames = ['ip', 'datetime', 'method', 'path', 'version', 'status',␣
 ↪'size', 'referrer', 'useragent', 'responsetime']
        writer = csv.DictWriter(outfile, fieldnames=fieldnames)
        writer.writeheader()

        for line in infile:
            parsed_line = parse_and_anonymize_line(line.strip())
            if parsed_line:
                writer.writerow(parsed_line)

input_file_path = '/content/input_logfiles.log'
output_file_path = '/content/anonymized_logsnew.csv'

# Processing
process_log_file(input_file_path, output_file_path)
```

Anonymizing PII and creating csv file as output

[7]:

```
import re

# Regular expression to match log entries
log_pattern = re.compile(
    r'(?P<ip>\d+\.\d+\.\d+\.\d+) - - \[(?P<datetime>[^\]]+)\] "(?
 ↪P<method>[A-Z]+) (?P<path>[^ ]+) (?P<version>HTTP/\d\.\d)" (?P<status>\d{3})␣
 ↪(?P<size>\d+) "(?P<referrer>[^"]*)" "(?P<useragent>[^"]+)" (?
 ↪P<responsetime>\d+)'
)

#replacing PII with <ANONYMIZING>
def anonymize_data():
    return "<ANONYMIZED>"

# Iterating through each data entry nd checking for PII
def parse_and_anonymize_line(line):
    match = log_pattern.match(line)
    if match:
        anonymized_line = f'{anonymize_data()} - - [{match.group("datetime")}]␣
 ↪"{match.group("method")} {anonymize_data()} {match.group("version")}" {match.
 ↪group("status")} {match.group("size")} "{anonymize_data()}"␣
 ↪"{anonymize_data()}" {match.group("responsetime")}'
        return anonymized_line
    return None

# Process the log file and write anonymized data to a new log file
def process_log_file(input_file_path, output_file_path):
    with open(input_file_path, 'r') as infile, open(output_file_path, 'w') as␣
 ↪outfile:
        for line in infile:
            anonymized_line = parse_and_anonymize_line(line.strip())
            if anonymized_line:
                outfile.write(anonymized_line + '\n')


input_file_path = 'input_logfiles.log'
output_file_path = 'anonymized_logs.log'


process_log_file(input_file_path, output_file_path)
```

2 Method 2 : Using Presidio- a pre-existing model to anonymize the PII

[3]:

```
!pip install presidio-anonymizer
```

```
Collecting presidio-anonymizer
  Downloading presidio_anonymizer-2.2.354-py3-none-any.whl (31 kB)
Collecting pycryptodome>=3.10.1 (from presidio-anonymizer)
  Downloading
pycryptodome-3.20.0-cp35-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(2.1 MB)
                           2.1/2.1 MB
10.3 MB/s eta 0:00:00
Installing collected packages: pycryptodome, presidio-anonymizer
Successfully installed presidio-anonymizer-2.2.354 pycryptodome-3.20.0
```

[4]:

```
!pip install presidio-analyzer presidio-anonymizer
```

```
Collecting presidio-analyzer
  Downloading presidio_analyzer-2.2.354-py3-none-any.whl (92 kB)
                           92.2/92.2 kB
2.2 MB/s eta 0:00:00
Requirement already satisfied: presidio-anonymizer in
```

```
/usr/local/lib/python3.10/dist-packages (2.2.354)
Requirement already satisfied: spacy<4.0.0,>=3.4.4 in
/usr/local/lib/python3.10/dist-packages (from presidio-analyzer) (3.7.4)
Requirement already satisfied: regex in /usr/local/lib/python3.10/dist-packages
(from presidio-analyzer) (2023.12.25)
Collecting tldextract (from presidio-analyzer)
  Downloading tldextract-5.1.2-py3-none-any.whl (97 kB)
                            97.6/97.6 kB
5.5 MB/s eta 0:00:00
Requirement already satisfied: pyyaml in /usr/local/lib/python3.10/dist-
packages (from presidio-analyzer) (6.0.1)
Collecting phonenumbers<9.0.0,>=8.12 (from presidio-analyzer)
  Downloading phonenumbers-8.13.34-py2.py3-none-any.whl (2.6 MB)
                            2.6/2.6 MB
10.7 MB/s eta 0:00:00
Requirement already satisfied: pycryptodome>=3.10.1 in
/usr/local/lib/python3.10/dist-packages (from presidio-anonymizer) (3.20.0)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (1.0.10)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (2.0.8)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (3.0.9)
Requirement already satisfied: thinc<8.3.0,>=8.2.2 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (8.2.3)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (1.1.2)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (2.4.8)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (2.0.10)
Requirement already satisfied: weasel<0.4.0,>=0.1.0 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (0.3.4)
Requirement already satisfied: typer<0.10.0,>=0.3.0 in
```

```
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (0.9.4)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (6.4.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (4.66.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (2.31.0)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (2.6.4)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages
(from spacy<4.0.0,>=3.4.4->presidio-analyzer) (3.1.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-
packages (from spacy<4.0.0,>=3.4.4->presidio-analyzer) (67.7.2)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (24.0)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in
/usr/local/lib/python3.10/dist-packages (from spacy<4.0.0,>=3.4.4->presidio-
analyzer) (3.3.0)
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.10/dist-
packages (from spacy<4.0.0,>=3.4.4->presidio-analyzer) (1.25.2)
Requirement already satisfied: idna in /usr/local/lib/python3.10/dist-packages
(from tldextract->presidio-analyzer) (3.6)
Collecting requests-file>=1.4 (from tldextract->presidio-analyzer)
  Downloading requests_file-2.0.0-py2.py3-none-any.whl (4.2 kB)
Requirement already satisfied: filelock>=3.0.8 in
/usr/local/lib/python3.10/dist-packages (from tldextract->presidio-analyzer)
(3.13.3)
Requirement already satisfied: annotated-types>=0.4.0 in
/usr/local/lib/python3.10/dist-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy<4.0.0,>=3.4.4->presidio-analyzer)
(0.6.0)
Requirement already satisfied: pydantic-core==2.16.3 in
/usr/local/lib/python3.10/dist-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy<4.0.0,>=3.4.4->presidio-analyzer)
(2.16.3)
Requirement already satisfied: typing-extensions>=4.6.1 in
/usr/local/lib/python3.10/dist-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy<4.0.0,>=3.4.4->presidio-analyzer)
(4.10.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from
requests<3.0.0,>=2.13.0->spacy<4.0.0,>=3.4.4->presidio-analyzer) (3.3.2)
```

```
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from
requests<3.0.0,>=2.13.0->spacy<4.0.0,>=3.4.4->presidio-analyzer) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from
requests<3.0.0,>=2.13.0->spacy<4.0.0,>=3.4.4->presidio-analyzer) (2024.2.2)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in
/usr/local/lib/python3.10/dist-packages (from
thinc<8.3.0,>=8.2.2->spacy<4.0.0,>=3.4.4->presidio-analyzer) (0.7.11)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in
/usr/local/lib/python3.10/dist-packages (from
thinc<8.3.0,>=8.2.2->spacy<4.0.0,>=3.4.4->presidio-analyzer) (0.1.4)
Requirement already satisfied: click<9.0.0,>=7.1.1 in
/usr/local/lib/python3.10/dist-packages (from
typer<0.10.0,>=0.3.0->spacy<4.0.0,>=3.4.4->presidio-analyzer) (8.1.7)
Requirement already satisfied: cloudpathlib<0.17.0,>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from
weasel<0.4.0,>=0.1.0->spacy<4.0.0,>=3.4.4->presidio-analyzer) (0.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from
jinja2->spacy<4.0.0,>=3.4.4->presidio-analyzer) (2.1.5)
Installing collected packages: phonenumbers, requests-file, tldextract,
presidio-analyzer
Successfully installed phonenumbers-8.13.34 presidio-analyzer-2.2.354 requests-
file-2.0.0 tldextract-5.1.2
```

[6]:

```python
from presidio_analyzer import AnalyzerEngine
from presidio_anonymizer import AnonymizerEngine
from presidio_anonymizer.entities import OperatorConfig
import csv

# Initializing Presidio Analyzer and Anonymizer
analyzer = AnalyzerEngine()
anonymizer = AnonymizerEngine()

# Function to anonymize a single line of text
def anonymize_text(text):
    analysis_results = analyzer.analyze(text=text, language='en')
    anonymized_results = anonymizer.anonymize(
        text=text,
        analyzer_results=analysis_results,
        operators={"DEFAULT": OperatorConfig("replace", {"new_value":␣
 ↪"<ANONYMIZED>"}), "IP_ADDRESS": OperatorConfig("replace", {"new_value":␣
 ↪"<IP-ANONYMIZED>"})}
    )
    return anonymized_results.text
```

```python
# Read the log file, anonymize content, and write to a new CSV file
def process_log_file(input_file_name, output_file_name):
    with open(input_file_name, 'r') as infile:
        lines = infile.readlines()

    with open(output_file_name, 'w', newline='') as outfile:
        writer = csv.writer(outfile)

        writer.writerow(['Anonymized Log Entry'])

        for line in lines:
            anonymized_line = anonymize_text(line.strip())
            writer.writerow([anonymized_line])


input_file_path = '/content/input_logfiles.log'
output_file_path = '/content/anonymized_logs_presidio.csv'

process_log_file(input_file_path, output_file_path)
```

```
WARNING:presidio-analyzer:configuration file /usr/local/lib/python3.10/dist-
packages/conf/default.yaml not found.  Using default config: {'nlp_engine_name':
'spacy', 'models': [{'lang_code': 'en', 'model_name': 'en_core_web_lg'}]}.
WARNING:presidio-analyzer:configuration file is missing
'ner_model_configuration'. Using default
WARNING:presidio-analyzer:model_to_presidio_entity_mapping is missing from
configuration, using default
WARNING:presidio-analyzer:low_score_entity_names is missing from configuration,
using default
WARNING:presidio-analyzer:labels_to_ignore is missing from configuration, using
default
WARNING:presidio-analyzer:Entity FAC is not mapped to a Presidio entity, but
keeping anyway. Add to `NerModelConfiguration.labels_to_ignore` to remove.
```

[ ]: