Detection and Classification of Personally Identifiable Information in Images Using Artificial Intelligence

Owais Shaikh[1]
[1] Security Researcher, RedHunt Labs Private Limited, Dehradun, India
Email: owais@0x4f.in

***Abstract*—Personally Identifiable Information (PII) is any content that is sensitive that needs to be treated as secure and private. When data pieces such as a person's name, address, Social Security number, phone number, email address, and so on may be used to identify a specific individual, they are deemed PII. As organizations grow, so does their volume of data. This makes identifying and protecting such sensitive resources at a scale quite complex. In this project, we demonstrate where and how PII can be discovered and how we developed a working prototype of a tool that can easily detect PII images using advanced artificial intelligence (AI) techniques such Optical Character Recognition (OCR) and image classification using Convolutional Neural Networks (CNN).**

***Index Terms*—personally identifiable information, attack surfaces, directory traversal, image classification, machine learning and optical character recognition.**

I. INTRODUCTION

Today, Personally Identifiable Information (PII) faces a wide variety of threats [1]. With the increasing number of security breaches, protecting valuable data such as Personal Identifiable Information (PII) must be the top priority of all organizations. Breaches involving PII are hazardous to both individuals and organizations. Individual harms may include identity theft, embarrassment, or blackmail. Organizational harms may include a loss of public trust, legal liability, or remediation costs. Breach of PII could also lead to regulatory and legal issues. In order to secure PII from leakage and exposure, organizations need to ensure proper handling and disposal of such assets. The first step in accomplishing this is to identify the exposure of such assets.

A. Problem Statement

In our pursuit for an extensible, free-to-use system for scanning and identifying PII for our own company, we stumbled upon many solutions, such as Amazon Macie [2] and ManageEngine Data Security Plus [3], each of them with their own quirks. Macie, for example, only supports S3 buckets and only integrates with Amazon's own services. Data Security Plus doesn't support S3 and is paid. We couldn't find an extensible open-source tool that could identify potentially exposed PII and satisfy our internal needs, so we at Red Hunt Labs developed Octopii and open-sourced it to generate momentum for it in the open-source community.

B. PII and its Exposure

Personally Identifiable Information (PII) is any content that is sensitive that needs to be treated as secure and private. Examples might include a rule that includes a behavior that sets a particular cookie or attaches PDFs of individual statements. Personally Identifiable Information can be used to identify or trace back an

individual. When data pieces such as a person's name, address, Social Security number, phone number, email address, and so on may be used to identify a specific individual, they are deemed PII. As organizations grow in size, so are their volumes of data linked with them. This makes identifying and protecting such sensitive resources at a scale becomes quite complex.

Protecting PII is solely the responsibility of the organization that's handling the data. Several incidents have emerged in recent years. Organizations failed to implement appropriate security standards, putting customers' confidential data at risk. In more than half of the cases, the PII was exposed through badly configured Amazon S3 buckets. Strict rules have been enacted against organizations that put their customers' data at risk. [4]

II. SOLUTION

Taking all the above facts into consideration, we at RedHunt Labs developed Octopii. Octopii is an AI-powered Personal Identifiable Information (PII) scanner that uses Tesseract's Optical Character Recognition (OCR) and Keras' Convolutional Neural Networks (CNN) models to detect various forms of Government IDs, passports, debit cards, driver's licenses, photos, signatures, etc. Let's take a closer look at how Octopii works and why it's essential to look out for exposed PII throughout your assets. Octopii's source code is publicly available [5] and can be downloaded for free from GitHub at https://github.com/redhuntlabs/Octopii

A. How Octopii works

The main objective of Octopii is to identify PII documents which can be either the original electronic copy of the document or the documents scanned and uploaded by the people. If the image is manually captured and uploaded, there are difficulties in determining the document type. As with a manually captured document, the orientation would be crooked, the image might not be well-lit,
the characters might not be readable owing to document degradation, it might be cropped, and so on. As a result, it becomes difficult to classify an image according to the type of document.
The image classification accuracy highly depends on how good the model is trained. Octopii uses an open-source library for image classification called Keras and does the following to classify an image:

B. *Importing and cleaning images*

Octopii possesses the ability to scan for exposed PII from HTML-based open-directory listings, Amazon S3 buckets or a local path on the web server. Depending on the type of path specified, Octopii imports the images via OpenCV and Python Imaging Library (PIL). To circumvent difficulties caused by manual scanning, the images are cleaned, deskewed and rotated for scanning. When a directory or an URL is provided, the tool will recursively traverse through each of the directories and fetch the images. Whether a file is an image or not is decided by OpenCV being able to read it. Currently, JPEG, PNG and GIF files are confirmed to be working.

C. *Image classification*

The neural network needs a trained data model of some kind to work with to provide accurate results. This is where machine learning (ML) comes into the picture. We generated a simple data model using Google's Teachable Machine, which uses a simple, 53-layer deep Mobile Net model. [6] This data model is generated from PII we blurred, and some standard-issue templates we generated after research.

This is how Octopii is able to "think" about how likely an image is to be a certain kind of image – because we bias its opinions on purpose via the model, therefore letting the machine tell us what an asset could be. It does not actually understand what these asset types are; it simply assigns an index number to it, which we cross with the labels file that we manually specify to understand the network's output.

D. *Optical Character Recognition (OCR)*

Later a verification method is performed in order to determine the accuracy of the image classified in the above method.OCR in Octopii is powered by Tesseract, an open-source OCR engine. Tesseract – like our image classifier – uses a neural network subsystem that is optimized solely to recognize lines of text.

Octopii simply calls Tesseract [7] on a copy of the image we feed into the image classifier to look for strings of text. We manually specify the search strings that Octopii uses, which contains unique words that only a certain type of PII may have (for example, a banking document may have the word “Passbook” which a driver’s license doesn’t have). This functionality can also be enhanced with regular expressions, which can not only improve accuracy but can also help users understand how rapidly and easily identity theft can be automated.

E. *Scoring*

A directory is looped over and searched for images. These images are scanned for unique features via the image classifier (done by comparing it to a trained model), along with OCR for finding substrings within the image. Octopii assigns a confidence score to these outcomes as shown in Table I:

TABLE I. ALL POSSIBLE CONFIDENCE SCORE RESULTS IN OCTOPII

Score	Scenario	Reliable
90 - 100	Best case	Yes
50 - 90	Average case	Yes
0 - 50	Worst case	Sometimes
Incorrect classification	False positive	No

• Best case (score >=90): The image is sent into the image classifier algorithm to be scanned for features such as an ISO/IEC 7810 card specification [8], colors, location of text, photos, holograms etc. If it is successfully classified as a type of PII, OCR is performed on it looking for particular words and strings as a final check. When both of these are confirmed, the result from Octopii is extremely reliable.
• Average case (score >=50): The image is partially/incorrectly identified by the image classifier algorithm, but an OCR check finds contradicting substrings and reclassifies it.
• Worst case (score >=0): The image is only identified by the image classifier algorithm but an OCR scan returns no results.
• Incorrect classification: False positives due to a very small model or OCR list may incorrectly classify PIIs, giving inaccurate results.

As a final verification method, images are scanned for certain strings to verify the accuracy of the model. The accuracy of the scan can be determined via the confidence scores in output. If all the mentioned conditions are met, a score of 100.0 is returned. These strings are verified against a manually edited JSON file containing a more specific list of strings to look for, along with their country and regular expressions for data fields such as identification numbers for extraction if needed.

F. *Training a model*

To train a new image classifier model, data can also be fed into a Keras, and the new and improved h5 model file can be used as input for Octopii. The easiest way to do this is via Google’s Teachable Machine. [9] The model file that Teachable Machine generates is based on a MobileNet network. We supplied 50 images of various types of PII such as passports, drivers’ licenses, government identity cards etc from different countries, with appropriately named classes, 500 epochs and a learning rate of 0.01. Teachable Machine then passes these images through its training algorithm and over and over again until the epoch limit is reached. [9] To verify accuracy, a user can hold up their ID card to a webcam and try to have it guessed by its label.

G. *Operation*

Since Octopii is a python-based command-line tool, you need to have your python environment setup correctly. It is necessary that your system should have all the required dependencies for Octopii to function properly. You can install the required modules all at once using the requirements.txt file. Octopii currently supports scanning local file system scanning, Amazon S3 buckets and open directory listings via their URLs. For users to test out the tool and to get a better understanding, we created a dummy environment with PII images hosted in it, and ran the following command to start an Octopii scan: *python3 octopii.py <URL>*

```
python3 octopii.py https://pii-carbonconsole.fra1.digitaloceanspaces.com/

[
    {
        "asset_type": "Passport",
        "country_of_origin": "International",
        "confidence": 100,
        "file_name": "dummy-passport-india.jpg",
        "extension": "jpg",
        "path": "https://pii-carbonconsole.fra1.digitaloceanspaces.com/dummy-passport-india.jpg"
    },
    {
        "asset_type": "Driver License",
        "country_of_origin": "International",
        "confidence": 100,
        "file_name": "dummy-drivers-license-nebraska-us.jpg",
        "extension": "jpg",
        "path": "https://pii-carbonconsole.fra1.digitaloceanspaces.com/dummy-drivers-license-nebraska-us.jpg"
    }
]
```

Figure 1. A sample output from the Octopii tool

As shown in Fig. 1 above, we set up a dummy Amazon Simple Storage Service (S3) instance on a DigitalOcean server. We configured the endpoint to purposely show a directory with images of several dummy PII, including Permanent Account Number (PAN) cards and Aadhaar cards from India and passports from around the world. This setup was a replication of an exposed S3 we found during our regular research for a client.

Once the S3 URL is passed as an argument, Octopii traverses through the S3 file system's XML and tries to download images into RAM. It then performs its scans on the images and is able to give us an estimate on its findings as shown above. It can return asset type, the country that the PII originates from, how confident it is in its findings and file metadata including the file's location or URL.

III. Conclusions

New attack vectors and vulnerabilities keep originating quite often [10] and might affect one (or many) assets across your organisation. During such times, having a precise external asset inventory makes it easy to scan for systems affecting the newly published vulnerability. Companies like RedHunt Labs help organizations discover their untracked assets, data exposure and external attack surface using indigenous tools, with their focus being an all-in-one attack surface management solution.

Storing sensitive customer, employee or user data such as Government ID, photos etc in a safe environment is extremely important. A lot of small businesses fail to understand this and end up storing them in unsafe, insecure environments such as open directories or S3s without proper authorization, prioritizing ease-of-access over better security. Since this area of security is less scrutinized than others, the solutions to detect PII are few and far-apart. Octopii is RedHunt Labs' effort to try and improve on this frontier of open-source intelligence.

Acknowledgment

Since Octopii is an open-source project [11], we appreciate contributions from the community. Octopii relies heavily on machine learning, and there's always room for improvement when training models are used. The

image classification model can be improvised by providing more datasets to train on. Since PII is highly confidential information, obtaining a large dataset to train the model is highly limited. We envision Octopii supporting the classification of several types of international documents, which is only possible via user support and open-source contribution.

We'd like to thank users that have forked our GitHub repository and added integration with platforms other than the provided defaults. We'd also like to thank Google for their Teachable Machine platform, which made our model creation process quick and easy-to-implement, and the developers involved with the SciKit, BeautifulSoup, OpenCV and Tesseract projects.

REFERENCES

[1] R. N. Zaeem, S. Budalakoti, K. S. Barber, M. Rasheed and C. Bajaj, "Predicting and explaining identity risk, exposure and cost using the ecosystem of identity attributes," 2016 IEEE International Carnahan Conference on Security Technology (ICCST), 2016, pp. 1-8, doi: 10.1109/CCST.2016.7815701.

[2] Amazon.com, Inc., "Macie," aws.amazon.com, August 14, 2017. [Online]. Available: https://aws.amazon.com/about-aws/whats-new/2017/08/introducing-amazon-macie/ [Accessed May 15, 2022].

[3] Zoho Corporation, "DataSecurity Plus," manageengine.com, January 20, 2021. [Online]. Available: https://download.manageengine.com/data-security/datasecurity-plus-quick-start- guide.pdf/ [Accessed May 12, 2022].

[4] Office of Management and Budget, Government of the United States, "Safeguarding Against and Responding to the Breach of Personally Identifiable Information," whitehouse.gov, May 22, 2007. [Online]. Available: https://www.whitehouse.gov/wp-content/uploads/legacy_drupal_files/omb/memoranda/2007/m07-16.pdf/ [Accessed May 15, 2022].

[5] RedHunt Labs Private Limited, "Octopii," github.com, April 20, 2022. [Online]. Available: https://github.com/redhuntlabs/Octopii/ [Accessed May 18, 2022].

[6] Keras, "MobileNet, MobileNetV2, and MobileNetV3," keras.io, April 12, 2020. [Online]. Available: https://keras.io/api/applications/mobilenet/. [Accessed May 16, 2022].

[7] University of Notre Dame, "Optical Character Recognition with Tesseract: a Tutorial for Medievalists,", nd.edu, May 11, 2022. [Online]. Available: https://sites.nd.edu/manuscript-studies/ 2022/05/11/optical-character-recognition-with-tesseract-a-tutorial-for-medievalists/ [Accessed May 16, 2022].

[8] International Organization for Standardization and International Electrotechnical Commission, "Identification cards — Physical characteristics," 2019 ISO/IEC JTC 1/SC 17 Committee, 2019, 4th edition, "Card dimensions and tolerances," ics: 35.240.15.

[9] Kyle Phillips, "Teachable Machine 2.0 makes AI easier for everyone", blog.google, November 07, 2019. [Online]. Available: https://blog.google/technology/ai/teachable-machine/ [Accessed April 30, 2022].

[10] J. Pastor-Galindo, P. Nespoli, F. Gómez Mármol and G. Martínez Pérez, "The Not Yet Exploited Goldmine of OSINT: Opportunities, Open Challenges and Future Trends," in IEEE Access, vol. 8, pp. 10282-10304, 2020, doi: 10.1109/ACCESS.2020.2965257.

[11] D. Agustian, P. P. G. P. Pertama, P. N. Crisnapati and P. D. Novayanti, "Implementation of Machine Learning Using Google's Teachable Machine Based on Android," 2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS), 2021, pp. 1-7, doi: 10.1109/ICORIS52787.2021.9649528.