**Nwachukwu Edumanichukwu**

**Answers Only**

**CYBR 445 - Advanced Incident Detection and Response  
Module 4 Lab – Data Loss Prevention**

In this fourth lab, we will examine the technologies that are used to discover and protect sensitive data. While there were two data loss protection systems that were open source, they have not been maintained and nearly every functional data loss prevention system is commercial, and very complicated to setup. The tools that underlie the systems are easy to understand and work with. We will examine these tools in the lab and then build our own simple data loss protection script that will use OCR and regular expression to examine a picture for sensitive data.

**You will be required to submit the following graded items as part of this lab:**

* Answer all questions listed in **BOLD**
* Provide screenshots when asked

Accessing the Lab

This lab is hosted in the universities IS Lab and requires special instructions to access it. If you are not familiar with accessing the IS Lab, please see the document in this course that walks you through accessing the Cybersecurity Desktop. You can access the Cybersecurity Desktop through the Web or using VMWare’s Horizon client. You should use the native Horizon client when possible as it provides better performance. The web client can be accessed at https://workspace.bellevue.edu. Make sure you log into this interface with your Bellevue student ID and password.

Part 1 - Regular-Expressions for Sensitive Data

In this first part of the lab, we will look at both regular expressions and additional algorithms such as the LUHN check for credit cards.

1. Start by opening and browser and surfing to <https://playground.nightfall.ai/data#pii>. Click on the PII/PCI and scroll to the second table that has Name, Email, Phone, Location, IP Address, and MAC address.

Graphical user interface, text, application, email

Description automatically generated

1. Open a second browser tab and browse to <https://regex101.com>. Copy the data from the entire second table on nightfall.ai to the Test String box on regex101.

Text

Description automatically generated

1. We are going to use regular expressions and named captures groups to automatically pull sensitive information from this data. Examine and copy the below regular expression into the Regular Expression textbox on regex101.com. Notice that this expression highlights and marks the email addresses, phone numbers, IP addresses, and MAC addresses in the test data we copied onto regex101. Make sure that you remove any whitespace characters that immediately following the regular expression if you are copying the regex from this document into regex101. The line should end with a +).

(?<email\_address>\w+@[\w.]+)\s+(?<phone\_number\({0,1}\d{0,1}-{0,1}\d{3}\){0,1}[\s.-]{0,1}\d{3}[\s.-]{0,1}\d{4}).\*\s+(?<ip\_address>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\s+(?<mac\_address>[a-f0-9]+:[a-f0-9]+:[a-f0-9]+:[a-f0-9]+:[a-f0-9]+:[a-f0-9]+)

1. Examine the match addresses and notice that groups are color coded and marked according to the regular expression and type of information.

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

1. Clear the regular expression and test string fields on regex101. Copy the information from the first table on nightfall.ai into the test string field on regex101.

Text

Description automatically generated

1. Now it’s your turn to write a DLP compatible regular expression. Use the Regular Expression text box and the example given above, along with any regular expressions guides or cheat sheets of your choosing to automatically select the two types of information, Name and Credit Card Number. The Test String and Match Information should look similar to the following:

Table

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Paste the following below:

* The plain text of your regular Expression
* A screenshot of the color-coded test string
* A screenshot of the Match Information pane on the right side of the screen

1. For some sensitive types of data, there are additional checks that can be done to separate random data from sensitive data. Credit cards, for example, have an algorithm called the Luhn Algorithm that can be used to validate that a CC number that matches a regular expression is also valid. Go to <https://www.dcode.fr/luhn-algorithm> and type in a simple fake credit card number such as 1111-2222-3333-4444. Note how the Luhn check fails.

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Go to the website <https://vccgenerator.org>. On the right side of the screen where it shows CC Generator, fill out the following information.
   1. Brand: Mastercard or VISA
   2. Country: United States
   3. CVV/CVV2: Random
   4. PIN: On
   5. Date: Random|Random
   6. Money: $500-$1000
   7. Quantity: 5
   8. Format: Text

Click Generate and then View Results. The Card Details will appear in the right Card Details pane. Copy a card number into the Number(s) to Check on the dcode.fr website and run the Luhn Check again. The results should be valid. Filtering out CC numbers by regex and then Luhn validations are a great way to eliminate false positives.

Graphical user interface, text, website

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

**Paste a screen shot of your generated CC numbers and the dcode website showing a valid Luhn check below.**

Note: You should never use real sensitive data for testing a DLP system as DLP systems will usually inspect things like outgoing e-mail, web traffic, network traffic, local file copies to removable media, and data transferred to applications like chat programs. Always make sure you are using fake test data that is designed to match regular expressions, Luhn algorithms, or any other checks but be unusable by an attacker if captured.

Part 2 - Google Image Webservices and DLP Demo

In this lab we will take a look at demos for two enterprise services, DLP based on plain text, and OCR services. These libraries, with network taps/sniffing tools, regular expressions, machine learning, hashing, and algorithms like the Luhn check are often combined to make a full DLP system.

1. Start by navigating to <https://cloud.google.com/dlp/demo/#!/>. You will see text in the upper panel, and a list of sensitive data discovered in the text on the bottom panel. Inspect the bottom panel and answer the questions below:

Table

Description automatically generated with medium confidence

**List the different types of sensitive data detected in the default text below.**

**Phone number, Email address, credit card number, domain name, US Driver’s License number, US Health Care NPI, URL, Person’s name, Ireland Driver’s License number, Australia Driver’s License number, Germany Identity Card Number and Document type/R&D/System Log**

**Are there any results that surprise you? Why or why not?**

**No. There are several ways and methods to identify individuals.**

**Can all of this data be detected using regular expressions?**

**Yes, it can be detected using regex.**

1. Open a new browser tab or window and navigate to <https://playground.nightfall.ai/data#pii>. Copy all of the data on the page from the first table to the last JSON data into the upper panel/text box on the Google DLP demo. Answer the following questions:

A screenshot of a computer

Description automatically generated with medium confidence

**How well did the Google DLP demo do at detecting sensitive data?**

**Google was able to detect and separate names and credit card.**

**List the sensitive data Google DLP detected:**

**Names and credit card numbers.**

**List any sensitive data that was missed below:  
There was nothing else left to detect.**

1. While detecting sensitive data in text is a great use case, there are other mode complicated use cases. One example is detecting sensitive data in images, video, audio, etc. There are now libraries and machine learning services that can inspect these file types and translate into plain text so that we can inspect for sensitive data. In a new browser window navigate to <https://cloud.google.com/vision/docs/drag-and-drop>. Click Try the API/Try It. You should see a box where ~~you~~ can drag and drop an image.

Graphical user interface, text, application, email

Description automatically generated

1. Navigate back to <https://playground.nightfall.ai/data#pii>. Use the Windows Snipping tool to take a screenshot of the second table having Name, Email, Phone, Location, IP Address, and MAC address using the rectangular snip mode. Save the snip as an image.

Graphical user interface, text, application, email

Description automatically generated

1. Upload the picture to the Google Vision API demo page. Wait for the page to process the image and click on Text. You should see the text in the picture on the right panel. Scroll those the view the text detected.

Graphical user interface, application

Description automatically generated

**How well did the Google Vision API do at detecting the text?**

**Google Vision API did an excellent job detecting text.**

Part 3 - Building a Simple OCR Data Loss Prevention Script

In this final lab we will utilize the libraries necessary to pull sensitive data from a picture and match for social security numbers. We will do this first on command line and then using Python. We will use Python, Tesseract (including the Python libraries), regular expressions, and grep.

1. From the IS Lab virtual machine desktop, open Firefox. Navigate to <https://10.98.100.11:4200>. Login in use the username analyst# and password An@lyst#!! where # is your student/analyst number.

Graphical user interface, text, application, email

Description automatically generated

1. You should be in your home directory in the terminal. Verify this by typing ls and ensuring the dlp\_test.png file is in your directory. If you do not see it, type cd ~.
2. Examine the image below. This this the dlp\_test.png image that is on your desktop. You will not be able to view this image directly over the terminal but we will be able to see the data contained in it once it extracted.

Table

Description automatically generated

1. Close the image by clicking the X in the top right-hand corner. Click in the terminal window. We will start by using tesseract on the command line to extract text via OCR. Run the following command: tesseract dlp\_test.png stdout -l eng.

Text

Description automatically generated

1. Now let’s extract the SSN from the image on the command line. Type the following command: tesseract dlp\_test.png stdout -l eng | grep -oP ‘^\d{3}-\d{2}-\d{4}’.

**Take a screenshot fo the command window and paste the results below.**

1. Now we will perform a similar function using Python and the Tesseract and Pillow libraries. Type the following commands:
   1. python
   2. import pytesseract
   3. from PIL import Image
   4. import re
   5. img = Image.open(‘dlp\_test.png’)
   6. text = pytesseract.image\_to\_string(img)
   7. text
   8. pattern = ‘\d{3}-\d{2}-\d{4}’
   9. re.findall(pattern,text)

**Paste a screenshot of the results below.**

1. Exit the terminal by click the x in the top right of the screen. Click Close Terminal. Exit Linux by clicking the power symbol on the top right of the screen/taskbar. Click Power Off/Log Out and then Log Out.