A Report on the Internship Programme At Sonata Software,Chennai

Name:SHANGARI C

College: Chennai Institute of Technology

Intern between 06-May-2024 to 31-May-2024

CONTENT

Introduction to Document AI

Document AI, also known as Document Artificial Intelligence, is a technology that leverages advanced AI techniques to process, analyze, and manage documents from Google Cloud Platform. It combines various AI components like Natural Language Processing (NLP), Optical Character Recognition (OCR), and Machine Learning (ML) to interpret and extract meaningful data from both structured and unstructured documents.

Explanation of How Document AI Works

Document AI works through a series of steps that involve the ingestion, analysis, and extraction of data from documents:

1. Document Ingestion:

- The process begins with the ingestion of documents, which can be in various formats such as PDFs, images, scanned copies, or digital text files.
- OCR technology is often used at this stage to convert images of text into machine-readable text.

2. Preprocessing:

- Preprocessing involves cleaning and preparing the text data for analysis. This may include noise reduction, layout analysis, and text normalization.

3. Text Analysis:

- NLP techniques are applied to understand and analyze the textual content. This includes tokenization, part-of-speech tagging, named entity recognition, and syntactic parsing.

4. Data Extraction:

- Specific information is extracted based on predefined templates or machine learning models. This could involve extracting key-value pairs, identifying entities, or summarizing the content.

5. Post Processing:

- The extracted data is then validated, organized, and stored in a structured format for further use or analysis.

Creating a application using Document AI

I developed a simple web UI using Flask to interact with Google Cloud's Document AI. The initial step was setting up the development environment. I installed Flask, a lightweight web framework for Python, and configured the Google Cloud SDK to authenticate and interact with Google Cloud Platform (GCP) services. This involved running the `gcloud init` command to set up the GCP project and install necessary components. Ensuring the development environment was properly configured was crucial for smooth subsequent development steps.

home.py

```
from flask import Flask, render_template, request
import os
from api import api
import filetype


app = Flask(__name__)

@app.route('/')
def upload_file():
    return render_template('ui.html')

@app.route('/upload', methods=['POST'])
def upload():
    file = request.files['file']
    if file.filename == '':
        return 'No selected file'

    if file:
        # Save the file to the templates directory
        file_path = os.path.join('templates', file.filename)
        file.save(file_path)
        kind = filetype.guess(file_path)
        if kind is None:
            print('Cannot guess file type!')
        MIME=(' %s' % kind.mime)
        print(MIME)
        MIME = kind.mime.strip()  # Remove leading and trailing
spaces
        print(MIME)
        result = api(file_path, MIME)
        print(result)
    return render_template('result.html', text=result)

if __name__ == '__main__':
    app.run(debug=True)
```

The next phase involved creating the Flask application itself. I started by structuring the basic Flask application, which involved defining routes and setting up the web server. This setup enabled secure authentication for all API requests made to Document AI services.

With the basic Flask app in place, I proceeded to connect it with Document AI. I initialized the Document AI client using the Google Cloud client library for Python. This step requires specifying the project ID, location, and processor ID, which uniquely identifies the Document AI processor configured in GCP. Proper initialization ensured that the application could communicate with the Document AI API to process documents.

docapi.py

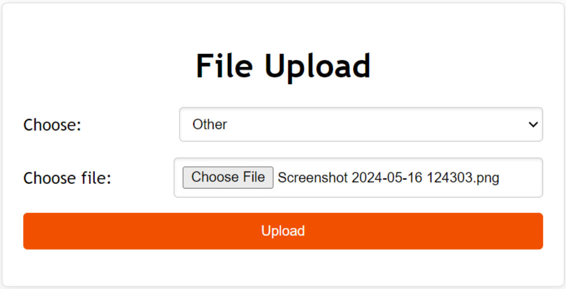
```
from google.api_core.client_options import ClientOptions
from google.cloud import documentai




def api(PATH,MIME):

    PROJECT_ID = "sonata-gcp-delivery"
    LOCATION = "us"  # Format is 'us' or 'eu'
    PROCESSOR_ID = "cc07d5761cdc74a1"  # Create processor in
Cloud Console

    # Refer to
https://cloud.google.com/document-ai/docs/file-types
    # for supported file types
    #MIME_TYPE = "application/pdf"
    MIME_TYPE=MIME
    FILE_PATH=PATH
    # Instantiates a client
    docai_client = documentai.DocumentProcessorServiceClient(

client_options=ClientOptions(api_endpoint=f"{LOCATION}-document
ai.googleapis.com")
    )

    # The full resource name of the processor, e.g.:
    #
projects/project-id/locations/location/processor/processor-id
    # You must create new processors in the Cloud Console first
    RESOURCE_NAME = docai_client.processor_path(PROJECT_ID,
LOCATION, PROCESSOR_ID)

    # Read the file into memory
    with open(FILE_PATH, "rb") as image:
        image_content = image.read()

    # Load Binary Data into Document AI RawDocument Object
    raw_document =
documentai.RawDocument(content=image_content,
mime_type=MIME_TYPE)

    # Configure the process request
    request = documentai.ProcessRequest(name=RESOURCE_NAME,
raw_document=raw_document)

    # Use the Document AI client to process the sample form
    result = docai_client.process_document(request=request)

    document_object = result.document
    print("Document processing complete.")
    print(document_object)
    text=(f"Text: {document_object.text}")
    return text
```

The core functionality of the application was implemented by creating an endpoint to process documents. This endpoint accepts file uploads via HTTP POST requests. When a file was uploaded, the application read its content and used the Document AI API to process it.

Running the Flask application locally was an essential part of the development process. By setting environment variables and running the `flask run` command,

Output

#image1

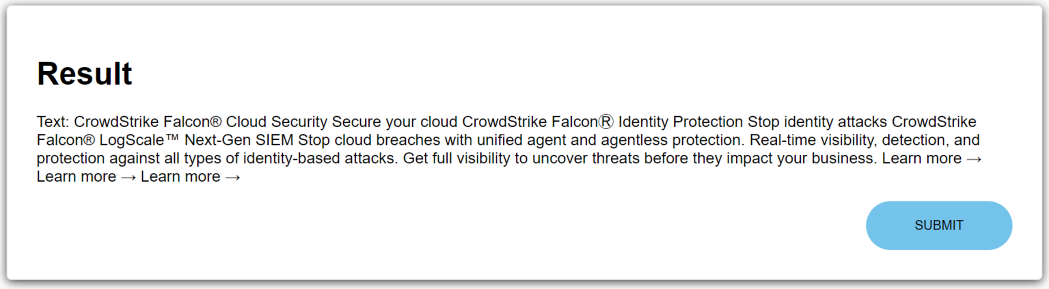


File Upload

Choose: Other

Choose file: Screenshot 2024-05-16 124303.png

#image2



Result

Text: CrowdStrike Falcon® Cloud Security Secure your cloud CrowdStrike FalconⓇ Identity Protection Stop identity attacks CrowdStrike Falcon® LogScale™ Next-Gen SIEM Stop cloud breaches with unified agent and agentless protection. Real-time visibility, detection, and protection against all types of identity-based attacks. Get full visibility to uncover threats before they impact your business. Learn more → Learn more → Learn more →

Preparing for deployment involved finalizing the application and ensuring all dependencies were listed in the requirements.txt file. Instead of using a typical cloud platform, I deployed the application on a Windows Server Virtual Machine (VM) provided by Sonata Company. This required configuring the VM to run the Flask application and ensuring it had all necessary software installed, such as Python and Flask. I set up the web server using a WSGI server like Gunicorn and configured it to serve the Flask app.

In summary, the project involved creating a Flask web application integrated with Google Cloud's Document AI. The development process included setting up the environment, authenticating with Google Cloud, implementing document processing functionality, and deploying the application to Google App Engine. Throughout this process, I gained valuable experience in using Flask, Google Cloud services, and Document AI, and learned to troubleshoot common development challenges. Future enhancements may include exploring additional Document AI features, improving the user interface, and implementing more robust error handling and validation mechanisms.

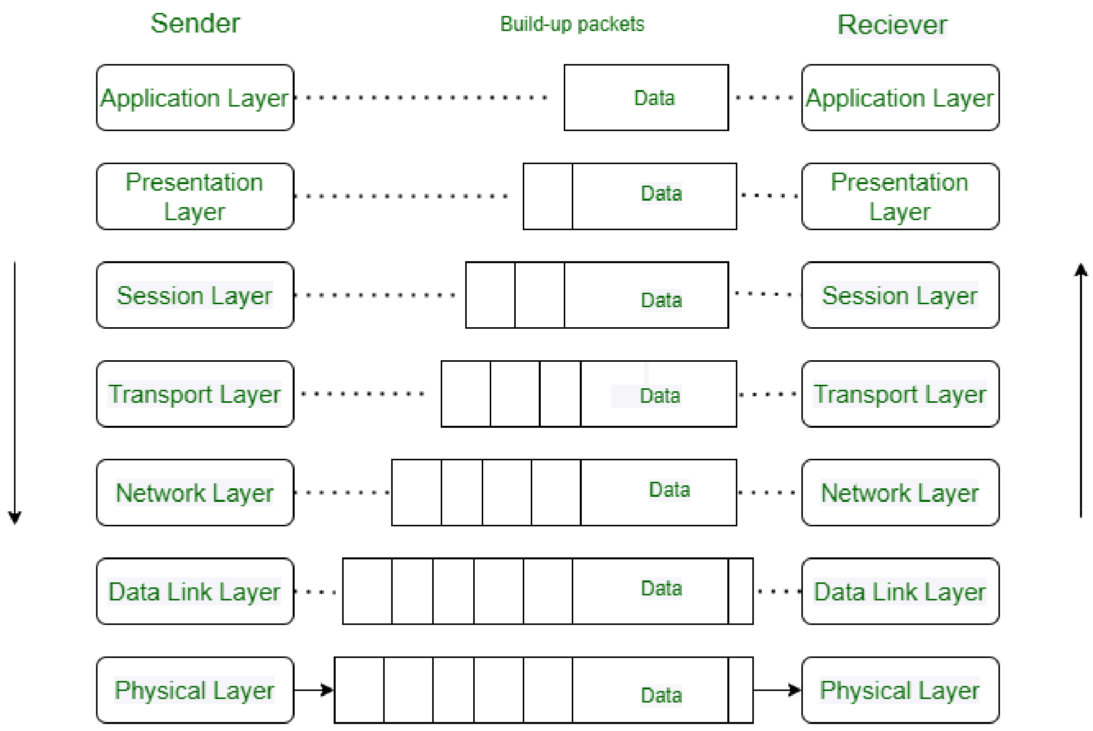
Introduction Class to IT infrastructure

During my internship, I attended a comprehensive class on IT infrastructure, focusing on the foundational elements that support modern computing environments. The session provided an in-depth understanding of the OSI (Open Systems Interconnection) model and different types of IP addresses. Additionally, practical exercises using Wireshark were conducted to inspect and analyze each layer of the OSI model.

OSI Model

The OSI model is a conceptual framework used to understand and standardize the functions of a telecommunication or computing system without regard to its underlying internal structure and technology. The model divides these functions into seven distinct layers, each serving specific roles and responsibilities in the process of data transmission across a network.

Layers of the OSI Model



Physical Layer Inspection: We observed the raw data being transmitted, including bit-level details.

Data Link Layer Inspection: Analyzed Ethernet frames, including MAC addresses and frame structure.

Network Layer Inspection: Examined IP packets, including source and destination IP addresses, and routing information.

Transport Layer Inspection: Inspected TCP/UDP segments, including port numbers and control flags.

Session, Presentation, and Application Layers: Analyzed higher-layer protocols and their data, such as HTTP requests and responses, SSL/TLS encryption details, and application data formats.

To reinforce our understanding, we used Wireshark to inspect and analyze network traffic across different layers of the OSI model. Wireshark allows for detailed examination of the data packets being transmitted over a network, providing insights into protocol operations and potential network issues.

IP Types

The class also covered different types of IP addresses used in networking:

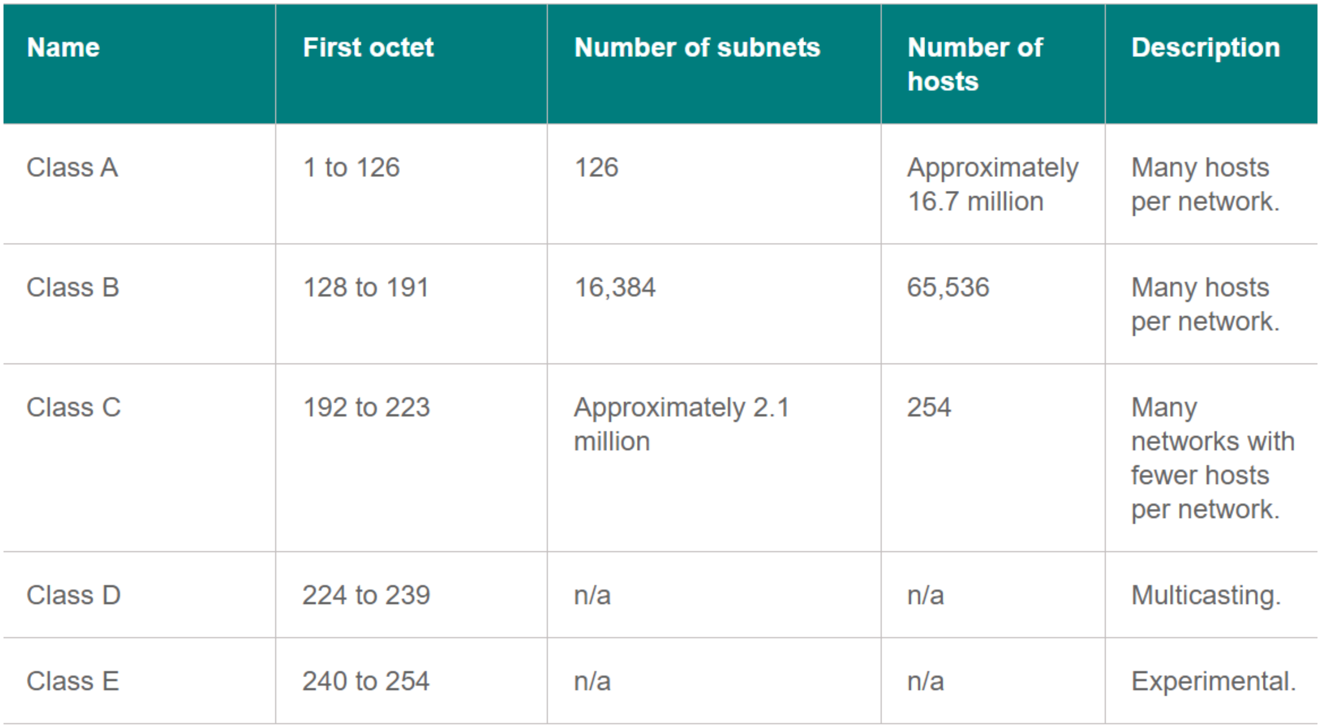
IPv4 (Internet Protocol version 4): The most widely used IP version, utilizing a 32-bit address format (e.g., 192.168.1.1). IPv4 supports approximately 4.3 billion unique addresses.

IPv6 (Internet Protocol version 6): Developed to address the limitations of IPv4, IPv6 uses a 128-bit address format (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334), providing a vastly larger address space.

Public IP Addresses: These are assigned by ISPs (Internet Service Providers) and are used to identify devices on the public internet. They are globally unique and routable on the internet.

Private IP Addresses: Used within private networks, these addresses are not routable on the public internet and are defined by specific ranges (e.g., 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8).

IP Listings



Name	First octet	Number of subnets	Number of hosts	Description
Class A	1 to 126	126	Approximately 16.7 million	Many hosts per network.
Class B	128 to 191	16,384	65,536	Many hosts per network.
Class C	192 to 223	Approximately 2.1 million	254	Many networks with fewer hosts per network.
Class D	224 to 239	n/a	n/a	Multicasting.
Class E	240 to 254	n/a	n/a	Experimental.

Cyber Security Introduction Class

During the internship, I attended an introduction class on Cyber Security, conducted by a security expert from the company. This class was designed to provide a comprehensive overview of the field of Cyber Security, emphasizing its importance in today's digital world and offering valuable career guidance for aspiring security professionals. The session lasted for three hours, covering various essential topics and practical insights.

The class began with an introduction to the fundamental concepts of Cyber Security. The expert explained the core principles of protecting computer systems, networks, and data from cyber threats and attacks.

The expert emphasized the importance of implementing effective security measures to mitigate risks. Topics covered included the use of firewalls, intrusion detection systems, encryption, and multi-factor authentication. We also discussed best practices for maintaining security, such as regular software updates, secure coding practices, and conducting security audits. The instructor provided practical tips on how to create strong passwords and recognize phishing attempts.

Career Guidance in Cyber Security

The latter part of the class focused on career guidance for those interested in pursuing a career in Cyber Security. The security expert shared insights into the various career paths available in the field, such as security analyst, penetration tester, security consultant, and security engineer. We discussed the skills and qualifications required for each role, including the importance of certifications like *Certified Information Systems Security Professional (CISSP), Certified Ethical Hacker (CEH), and CompTIA Security+*.

The expert provided valuable advice on how to build a successful career in Cyber Security. This included recommendations on gaining practical experience through internships, participating in cybersecurity competitions, and staying updated with the latest industry trends and technologies. The importance of continuous learning and professional development was emphasized, as the field of Cyber Security is constantly evolving with new challenges and innovations.

Conclusion

The Cyber Security introduction class was an enlightening and informative session that provided a solid foundation in understanding the key concepts and challenges in Cyber Security. The career guidance offered by the security expert was particularly beneficial, offering clear direction and motivation for pursuing a career in this dynamic and essential field. Overall, the class was a valuable addition to my internship experience, enhancing my knowledge and interest in Cyber Security.

Presentation: SentinelOne

During my internship, I prepared and delivered a presentation on SentinelOne, focusing on its Extended Detection and Response (XDR) solutions. The presentation was an in-depth review of SentinelOne products, Singularity™ XDR Platform, a next generation Extended Detection and Response (XDR) solution designed to provide holistic cybersecurity capabilities.It encompasses **AI powered prevention, detection, response, and threat hunting across user endpoints, containers, cloud workloads, and IoT devices**.Enabling modern enterprises to defend faster, at greater scale, and with higher accuracy across their entire attack surface, we empower the world to run securely.

Unified endpoint protection lies at the core of SentinelOne's services, offering organizations a consolidated solution for protecting their endpoints against a wide range of cyber threats. By integrating endpoint protection platform (EPP) and endpoint detection and response (EDR) functionalities, SentinelOne's **unified endpoint protection services** provide organizations with comprehensive protection against malware, ransomware, and other cyber threats.

Furthermore, SentinelOne offers specialized threat services focused on detecting, analyzing, and mitigating cyber threats. **These services include endpoint protection platform (EPP), endpoint detection and response (EDR), ransomware protection, cloud workload protection, and threat intelligence.** By leveraging specialized threat services, organizations can enhance their cybersecurity posture and effectively mitigate cyber risks.

Support services provided by SentinelOne encompass expert assistance in ongoing security management and incident response. **Managed detection and response (MDR)** services enable organizations to outsource their threat detection and response functions to expert security professionals, while **incident response (IR) services** provide organizations with rapid and effective incident response capabilities.

Deployment and health services offered by SentinelOne ensure the effective **implementation and maintenance** of cybersecurity operations. Automated response and remediation capabilities enable organizations to **automate the response to detected threats**, while visibility across all endpoints provides organizations with comprehensive visibility into their cybersecurity posture. Through deployment and health services, organizations can ensure that their cybersecurity operations are effective and resilient against emerging threats.

SENTINELONE VS CROWDSTRIKE:

SENTINELONE	CROWDSTRIKE
● Offers automated deployment. Singularity Ranger covers your blindspots and automatically deploys new agents in real time, as needed.	● Visibility only for managed devices, creates ongoing risk of exposure.
● Manage large deployment with ease with remote script across multiple assets. Full remote native OS tools coverage.	● Manage individual assets using remote commands, no bulk operations.
● Automated remediation. Revert malicious activities with one-click remediation and rollback.	● Manual and script-based mitigation for most alerts types. No rollback support.

SENTINELONE VS McAfee:

SENTINELONE	McAfee
● Automatically mitigate against cyber threats across Windows, macOS, and Linux.	● Does not offer options to remediate malicious actions, only offering traditional mitigation actions.
● Automatically reconstructs events into easily navigable Storylines™, offering focused, contextualized alerts for analysts for faster MTTR.	● Requires manual correlation and context-switching between products for investigation and hunting.
● Military-grade prevention, detection, and response powered by patented behavioral AI. Always on, no internet connection required.	● Human powered protection creates delays and misses. Requires heavy tuning to make cybersecurity happen.

FEATURES OF SENTINELONE :

1. **Automated Deployment:** SentinelOne offers automated deployment capabilities through Singularity Ranger, ensuring real-time agent deployment across your network.

2. **AI-Powered Threat Detection:** Leveraging advanced artificial intelligence, SentinelOne detects and responds to threats with high accuracy, minimizing false positives and negatives.

3. **Unified Endpoint Protection:** SentinelOne provides comprehensive protection across all endpoints, regardless of the operating system or device type, streamlining security management.

4. **Integration and Scalability:** The platform integrates seamlessly with existing security tools and scales effortlessly to accommodate organizational growth and evolving security needs.

5. **Automated Remediation**: SentinelOne enables one-click remediation and rollback, allowing you to revert malicious activities quickly and efficiently, reducing the impact of security incidents.

Troubleshooting References

Troubleshooting #1- Port configuration:

Check the port number specified with the flask application is the same as specified in the port number of the server.

Reference Links

https://cloud.google.com/document-ai/docs/reference/rest/v1/ProcessOptions
https://www.crowdstrike.com/en-us/
https://www.techtarget.com/searchnetworking/tip/Introduction-to-IP-addressing-and-subnetting
https://www.geeksforgeeks.org/open-systems-interconnection-model-osi/

Conclusion

My internship at Sonata Software has been an immensely valuable experience, providing me with deep insights into cybersecurity and network management. Through hands-on work with advanced tools like SentinelOne and CrowdStrike, I have developed a robust understanding of AI-driven threat detection, endpoint protection, and cloud security. Additionally, I have gained practical knowledge in network fundamentals, IP addressing, and the implementation of security measures such as Kiosk Mode. These experiences have significantly enhanced my technical skills and prepared me for future challenges in the field of cybersecurity.

I would like to extend my heartfelt gratitude to Mr.THIRUVENKADATHAN NARAYANAN, Mr.KUMAR THIRUVENGADAM, and Mr. BALA MURUGAN or their unwavering guidance and support throughout this internship. Their expertise and mentorship have been instrumental in my learning journey, and I am deeply appreciative of their contributions to my professional growth. Thank you for providing me with the opportunity and resources to succeed, and for fostering an environment that encourages learning and development.