INTERNSHIP

2019-2020

Project Report

On

Sementic Scanner & Severity Checker

By

Ritam Ghosh (16011046)

Sarthak Agrawal (160102288)

Aayush Chauhan (160102265)

Joy Dey (160102253)

DIT University, Dehradun

(Bachelor of Technology in Computer Science & Engineering)

Under the Guidance of

Dr. Santanu Chatterjee

(Scientist)
(Directorate of Information & Communications Technology)

Research Centre Imarat, DRDO, Ministry of Defence, Govt of India

RCI Road, Vigynana Kancha, Hyderabad, Telangana 500069

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance. We are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and we would not forget to thank them.

We owe our deep gratitude to our project guide **Dr. Santanu Chatterjee,** Scientist, Directorate of Information & Communications Technology, Research Centre Imarat, DRDO, Ministry of Defence, Govt of India, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all scientists who helped us in successfully completing our project work. Also, we would like to extend our sincere esteems to all the technical staff for their timely support.
We once again sincerely thanks all those who have helped us directly and indirectly during my project work.

Ritam Ghosh

Sarthak Agrawal

Aayush Chauhan

Joy Dey

LIST OF CONTENTS

ABSTRACT

Semantic Scanner & Severity Checker is an application, which allows an organization or an individual to secure its sensitive data from getting vulnerable to third party access & its use. It also performs side by side scanning of all files that are transferable and provide a severity threat label.

If the severity is critical then the report and file is directly transferred to the risk analyzing committee of an organization to dual check and send final response. Responses are trained side by side and datasets are generated, which can be used for direct final approval.

Its main functioning begins when someone tries to transfer critical files to other third-party organization or individuals. And basic scanning and severity labeling process works in background. Also, it is used to structured and label unstructured data by assigning severity to it.

It can be helpful for incident response team for analysis and further investigation for compromised data of an organization.

Mainly it is designed for data security and surveillance for an organization. Our product act as a security intelligence system, which can used by defense services and private organization to safe guard their data from being vulnerable to third party access and use.

ABOUT DRDO

Defence Research & Development Organization (DRDO) works under Department of Defence Research and Development of Ministry of Defence. DRDO dedicatedly working towards enhancing self-reliance in Defence Systems and undertakes design & development leading to production of world class weapon systems and equipment in accordance with the expressed needs and the qualitative requirements laid down by the three services.

DRDO is one of the prestigious organizations of the country in the field of Science and Technology, which could transform our country's Defense force into one of the most modern and powerful force in the world. It was established by merging together the Scientific and Technical Development Establishment under three services headquarters in 1958, with the aim of creating an organization that can take up the challenges of developing and delivering the high technology in the field of modern warfare, weapon system, avionics and other scientific aspects of nation's defense. It has also got mandate to modernize Defense Technology

DRDO is working in various areas of military technology which include aeronautics, armaments, combat vehicles, electronics, instrumentation engineering systems, missiles, materials, naval systems, advanced computing, simulation and life sciences. DRDO while striving to meet the Cutting-edge weapons technology requirements provides ample spinoff benefits to the society at large thereby contributing to the nation building.

Vision:

Make India prosperous by establishing world class science and technology base and provide our Defence Services decisive edge by equipping them with internationally competitive systems and solutions.

Mission:

Design, develop and lead to production state-of-the-art sensors, weapon systems, platforms and allied equipment for our Defence Services.

Provide technological solutions to the Services to optimize combat effectiveness and to promote well-being of the troops.

Develop infrastructure and committed quality manpower and build strong indigenous technology base.

Research Centre Imarat (RCI)

It is a premiere DRDO laboratory located in Hyderabad. The lab is responsible for Research and Development of Missile Systems, Guided Weapons and advanced Avionics for Indian Armed Forces It was established by APJ Abdul Kalam in 1988. Scientist and avionics specialist BHVS Narayana Murthy are presently the Director RCI Laboratory.

The Research Centre Imarat is a global frontrunner in developing avionics and navigation systems for missiles

RCI is the leading laboratory which has successfully spearheaded the Indo-Israel joint development Medium Range Surface to Air Missile (MRSAM) program and had hat-trick success in its first three consecutive missions.

INTRODUCTION:

Semantic Scanner & Severity Checker is a system which is used by an organization to safeguard its data from being compromised. It firstly scans all files input in a given system by using specific parameters as file size and file types as per their extensions and maintains a safe record which is used further. If any user tries to transfer the files to any third-party organization or an individual, then it starts to check the files and label it accordingly to its severity as per parameters given. Then if the severity is normal or low then the files can get transfer easily otherwise if the severity is high or critical then its gets blocked and an alert will be send to mediator committee who is responsible for validating and analyzing the files selected to be transfer for, if the mediator provides green signal after analyzing then the file can be transfer and gets unblocked otherwise if it gets red signal from mediator its gets permanently blocked from being transfer.

Here in this project all the process is continuously working on background, till any user tries to transfer any files to third-party. And all the responses of the mediator are stored and datasets are generated out of it, which is then trained and use for auto validation.

Our project can be used for industrial use, only need few modifications and parameters to be adjusted as per their demands & needs. This project can be integrated with system with their security or firewall modules.

OBJECTIVE:

Semantic Scanner & Severity Checker main aim is to prevent data leaks and limits its data sharing capabilities and to safe guard our data from being compromised.

PROPOSED FIGURES & DIAGRAMS:

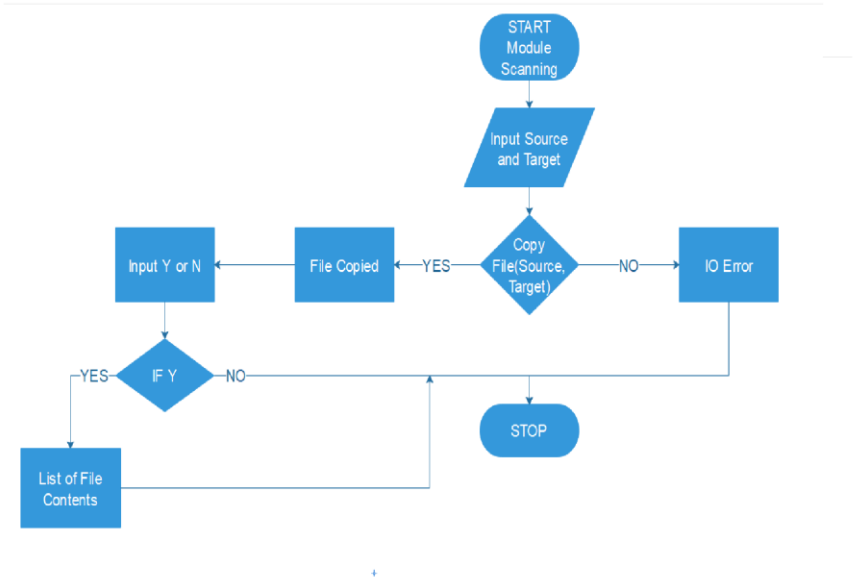


Fig.01

FIG. 1 depicts an exemplary Program diagram of a Scanning system in which exemplary aspect of validation of the file according to the various parameters is implemented.

The pictorial representation of the scanning module depicts that, firstly the module is being imported from the flask application and hence program is initialized. the path to the source and the destination is provided by the user to copy the files from the external device to the target folder. While copying the files, file type and the file size are to be validated. For transferring the files these two parameters are checked and further action is taken. If the file is unable to copy then the error is displayed, otherwise the user is asked whether to display the file contents that are copied from the external device to the system.

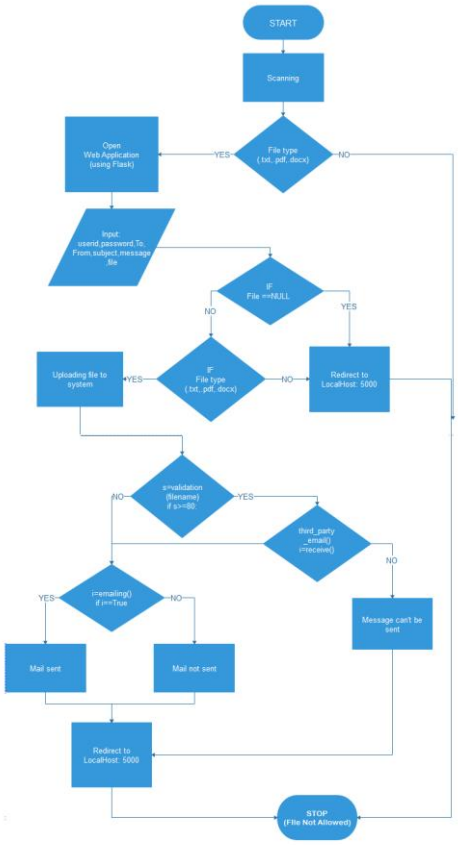


Fig.02

FIG. 02 is an exemplary block diagram of Authentication system in which the aspects of web development machine libraries i.e. Flask is being implemented.

With reference to the above fig. 02, Authentication module is implemented and the various other modules are imported for the authentication process. The authentication system validates the file according to the file type and the file size and all the credentials that are required over the web. After the validation, severity for the file i.e. uploaded is measured and on the basis of the severity measured further action is taken. If the severity is low than the credentials are transferred over the web from the sender to the receiver else the file is transferred from the authenticator to the third-party validator. The response generated by the mediator in true or false validates whether the file will be sent over the mail from the sender to the receiver or not. During any error generation or the success of the transfer of the credentials the page over the web is redirected to the localhost at port 5000.

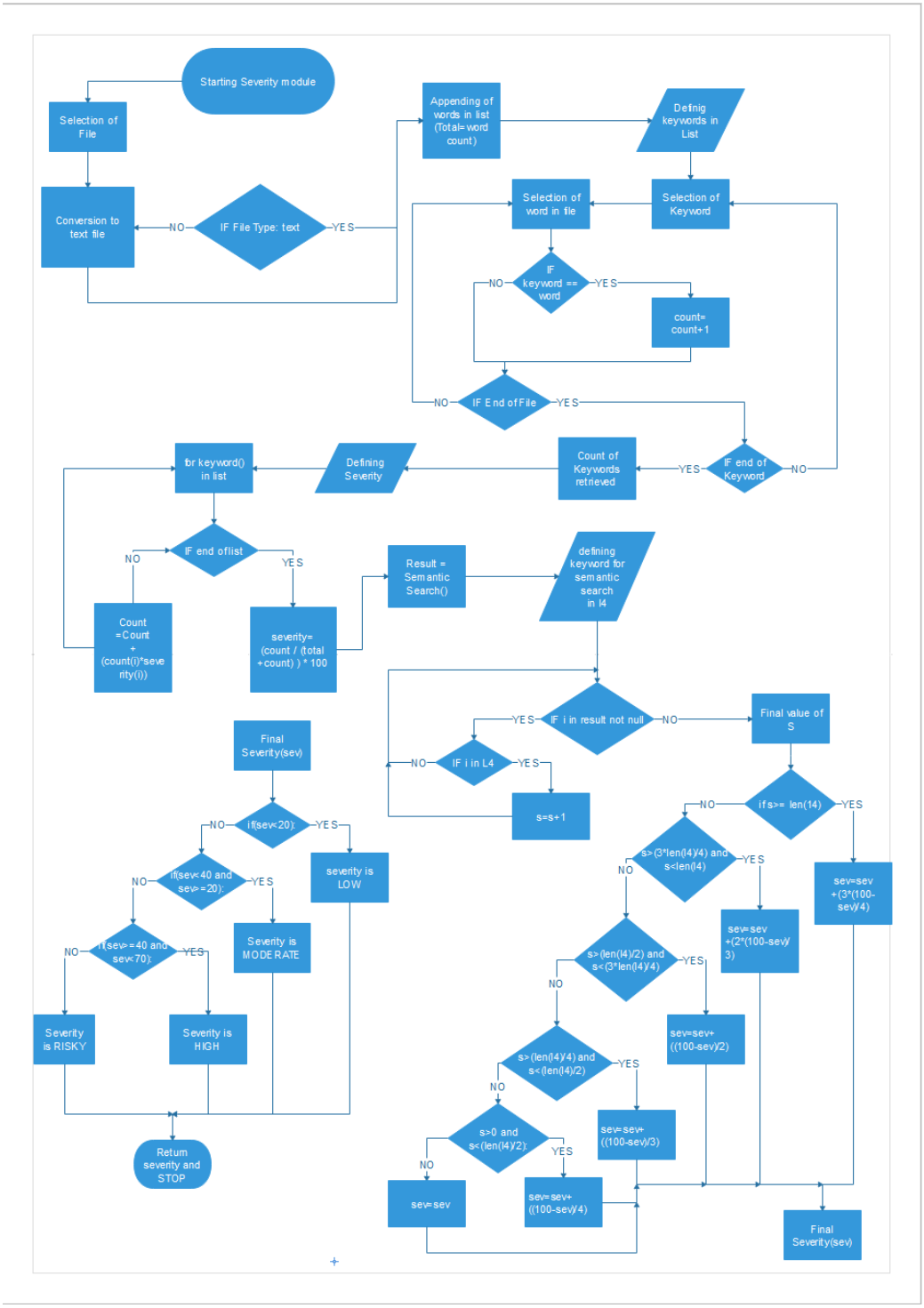

Starting Severity module
Selection of File
Conversion to text file
NO
IF File Type: text
YES
Appending of words in list (Total= word count)
Defnig keywords in List
Selection of word in file
Selection of Keyword
IF keyword == word
NO
YES
count= count+1
IF End of File
NO
YES
IF end of Keyword
YES
NO
Count of Keywords retrieved
Defining Severity
for keyword() in list
IF end of list
NO
YES
Count =Count + (count(i)*seve rity(i))
severity= (count / (total +count)) * 100
Result = Semantic Search()
defining keyword for semantic search in l4
IF i in result not null
YES
NO
Final value of S
IF i in L4
NO
YES
s=s+1
Final Severity(sev)
if(sev<20):
NO
YES
severity is LOW
if(sev<40 and sev>=20):
NO
YES
Severity is MODERATE
if(sev>=40 and sev<70):
NO
YES
Severity is RISKY
Severity is HIGH
Retum severity and STOP
if s>= len(l4)
NO
YES
sev=sev +(3*(100-sev)/4)
s>(3*len(l4)/4) and s<len(l4)
NO
YES
sev=sev +(2*(100-sev)/3)
s>(len(l4)/2) and s<(3*len(l4)/4)
NO
YES
sev=sev+ ((100-sev)/2)
s>(len(l4)/4) and s<(len(l4)/2)
NO
YES
sev=sev+ ((100-sev)/3)
s>0 and s<(len(l4)/2):
NO
YES
sev=sev
sev=sev+ ((100-sev)/4)
Final Severity(sev)

Fig.03

FIG. 3 illustrates a data leak solution in which the severity of the file is tested for the aspect of security of the important information of an organization.
With reference to the Fig. 03, initially the file that is uploaded through the flask application is sent to severity module than this file is converted in the text type file. After conversion the keywords provided by an organization is being compared and the frequency of occurrence of the keywords in the file is measured. The phrases are also defined to measure the severity of the file and hence using semantic searching and optimal matching we obtain the final severity. On the basis of the final recording of severity, popup message is displayed depicting the severity and hence at last the reading of the severity for the uploaded file is returned to the authenticator.

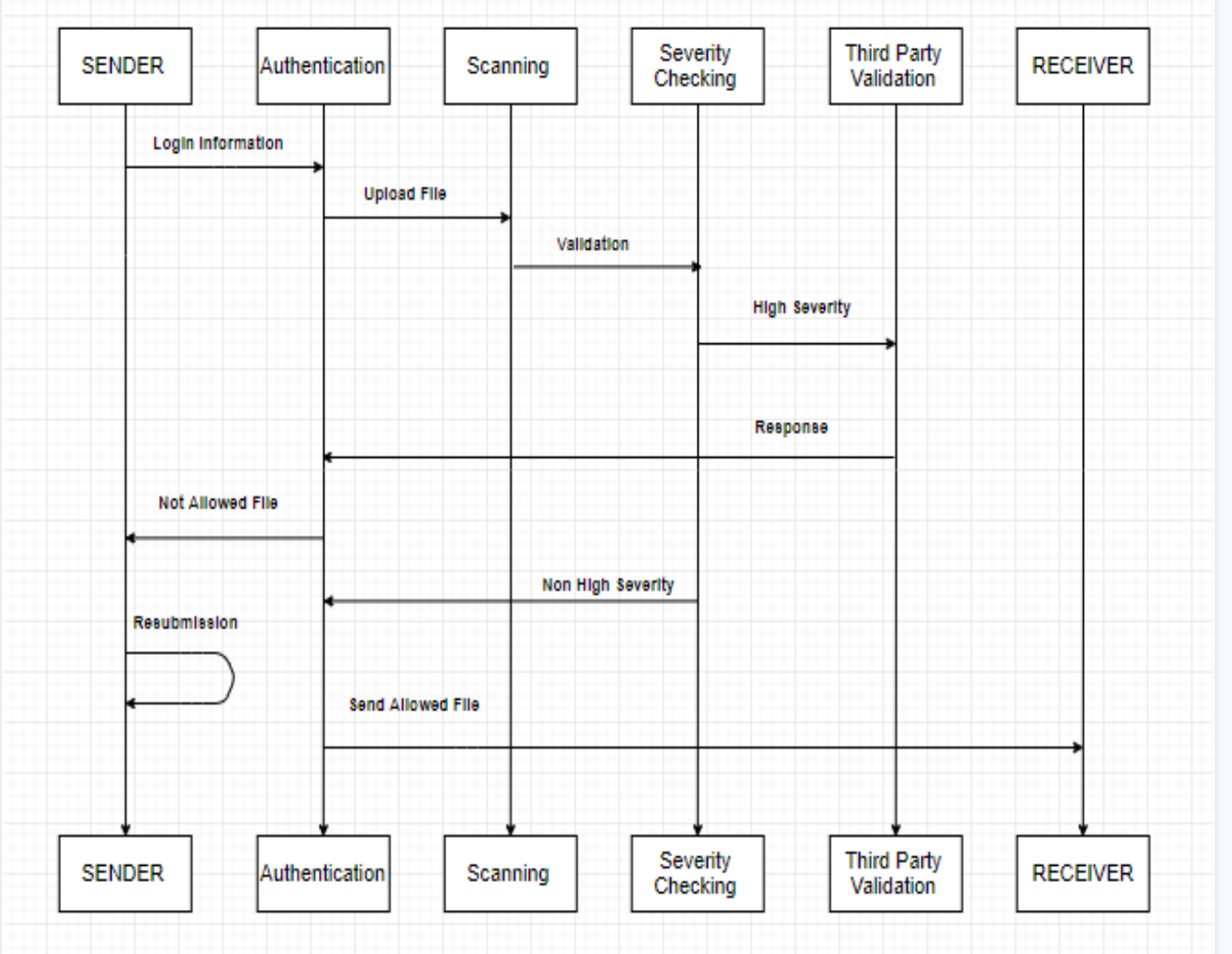


Fig.04

Fig. 4 illustrates the sequential flow of the processes from the different methodologies to one another. This sequence diagram depicts how the security is maintained for an organization.

With reference to the above Fig. 04, sender fills the credentials in the login form and the information is sent to the authenticator. Then the authenticator validates the file on the basis of file extension and file size and then the severity of the file uploaded is checked. According to the severity measured, if the severity is high then the file is sent to the mediator that is third-party validator else the credentials are directly sent to the receiver from the sender. In the case of the high severity the file sending depends on the response provided by the mediator. If the response is true then the credentials are allowed over the web else these credentials are blocked from transfer over the web.

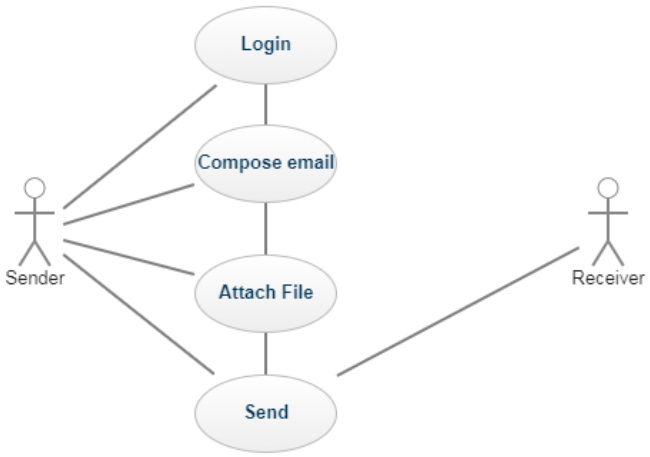


Fig.05

Fig.05 illustrates the basic sender and receiver use case model, which presents sender action over the web in which sender sends an email to receiver .First of all sender will login using his/her email id and password, then he will compose mail by adding subject and message, next he will attach a file to it which he wants to send to a receiver, finally an email will be send to a receiver.

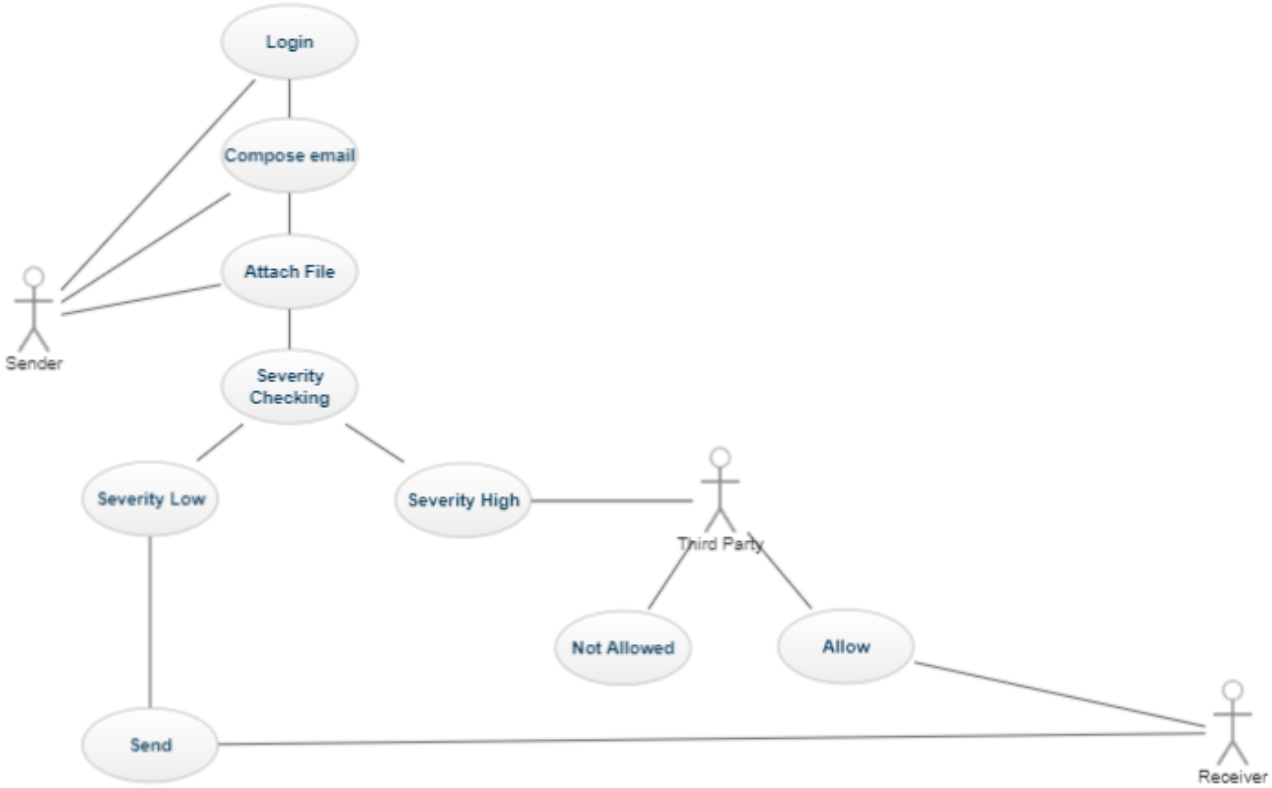


Fig.06

Fig.06 Illustrate more functionality of our project, where third party is involved between sender and receiver. In this use case diagram when sender login into his email, and after which he composed the message and attach the file which he wish to send. The attached file is check for severity which tells whether the file is risky or not. If severity is low the mail will be send directly to a receiver. If a severity is high than a mail from backend will be send to a trusted third party which will whether allow the mail to send or block it. If he allows mail then the mail will be send to receiver otherwise it will not be send.

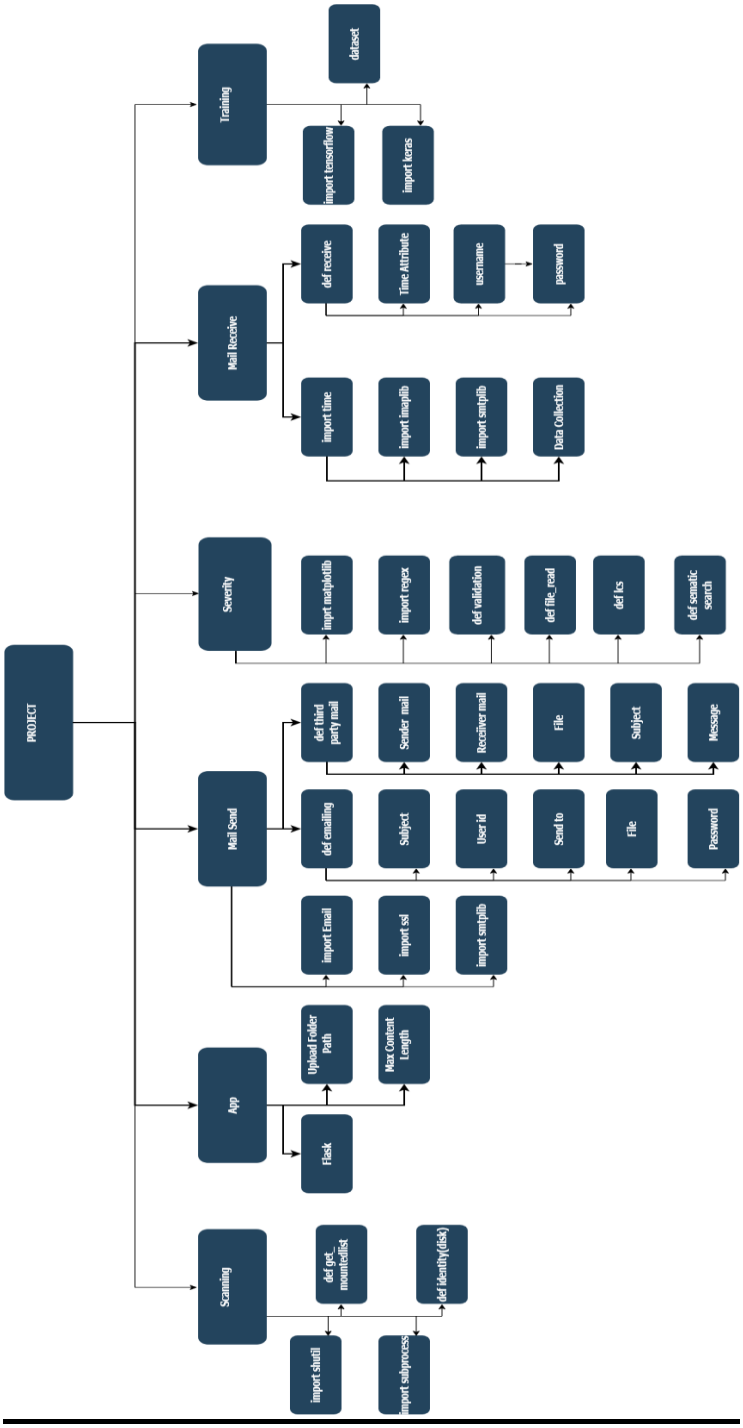


Fig.07

Fig.07 represents a full map of a project “Semantic Scanner and Severity Checker”. It is a basic skeleton structure of this project which represents all the modules, libraries and functions involved in the different files of the project.

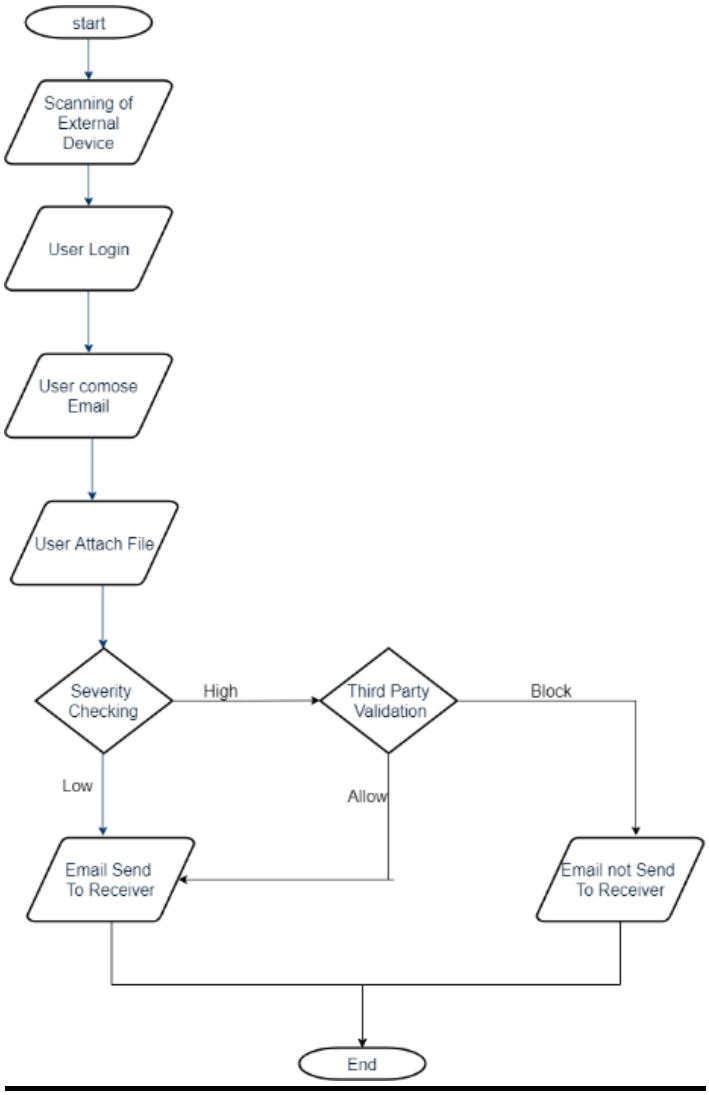


Fig.08

Fig.08 Illustrate a activity diagram of our project . In this diagram it describes that , first scanning of the external device will take place after which sender login into his email, and after which he composed the message and attach the file which he wish to send. The attached file is check for severity which tells whether the file is risky or not. If severity is low the mail will be send directly to a receiver. If a severity is high than a mail from backend will be send to a trusted third party which will whether allow the mail to send or block it. If he allows mail then the mail will be send to receiver otherwise it will not be send.

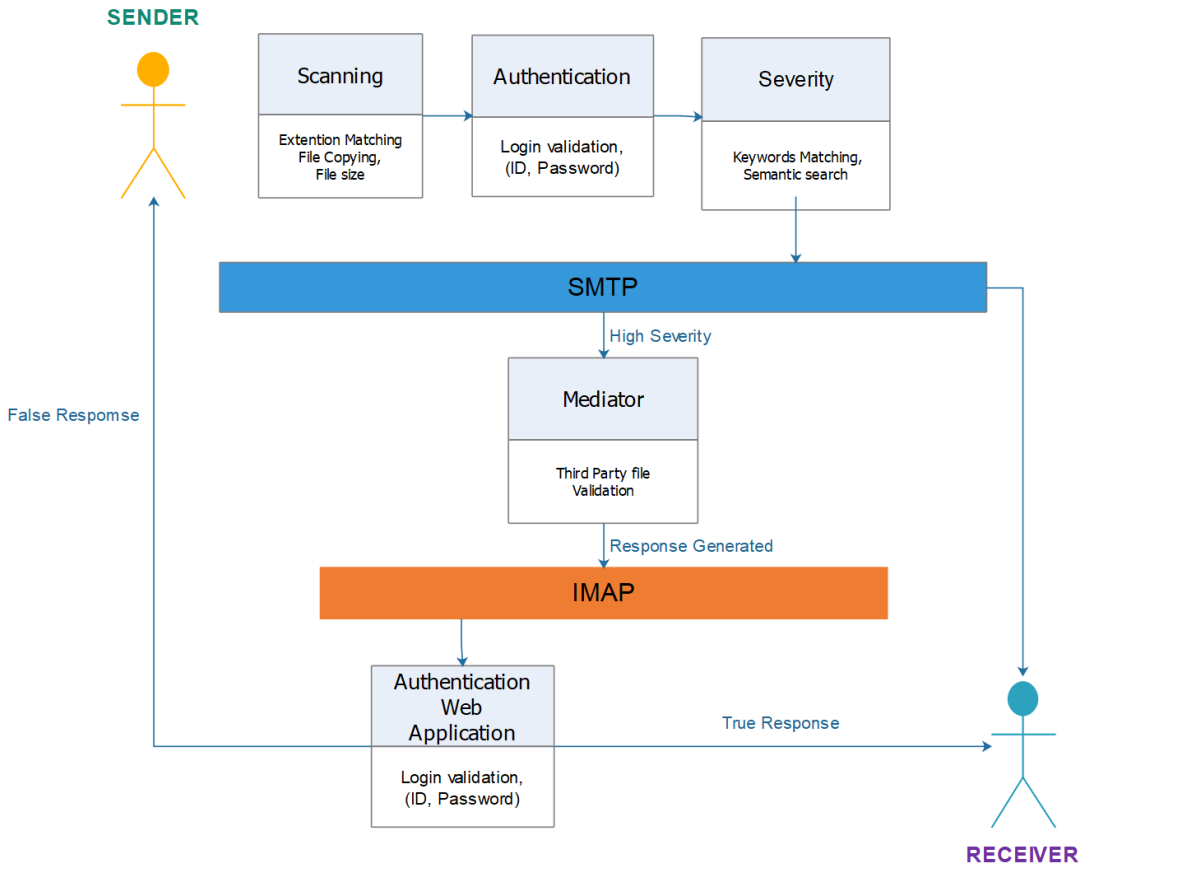


Fig.09

Fig.09 Illustrates the Architecture of the System, which includes functionality of scanning, authentication, and severity together then, passes the result through SMTP. If there is high severity then it proceeds to mediator which analyzes and validates and generate a response which is them passes through IMAP to authenticator otherwise it will be directly send to the receiver. If mediator sends true response then message is sent otherwise if false it will blocked.

METHODLOGY:

It contains 5 basics stages:

- Scanning
- Authentication
- Severity
- Mediator
- Auto Validation

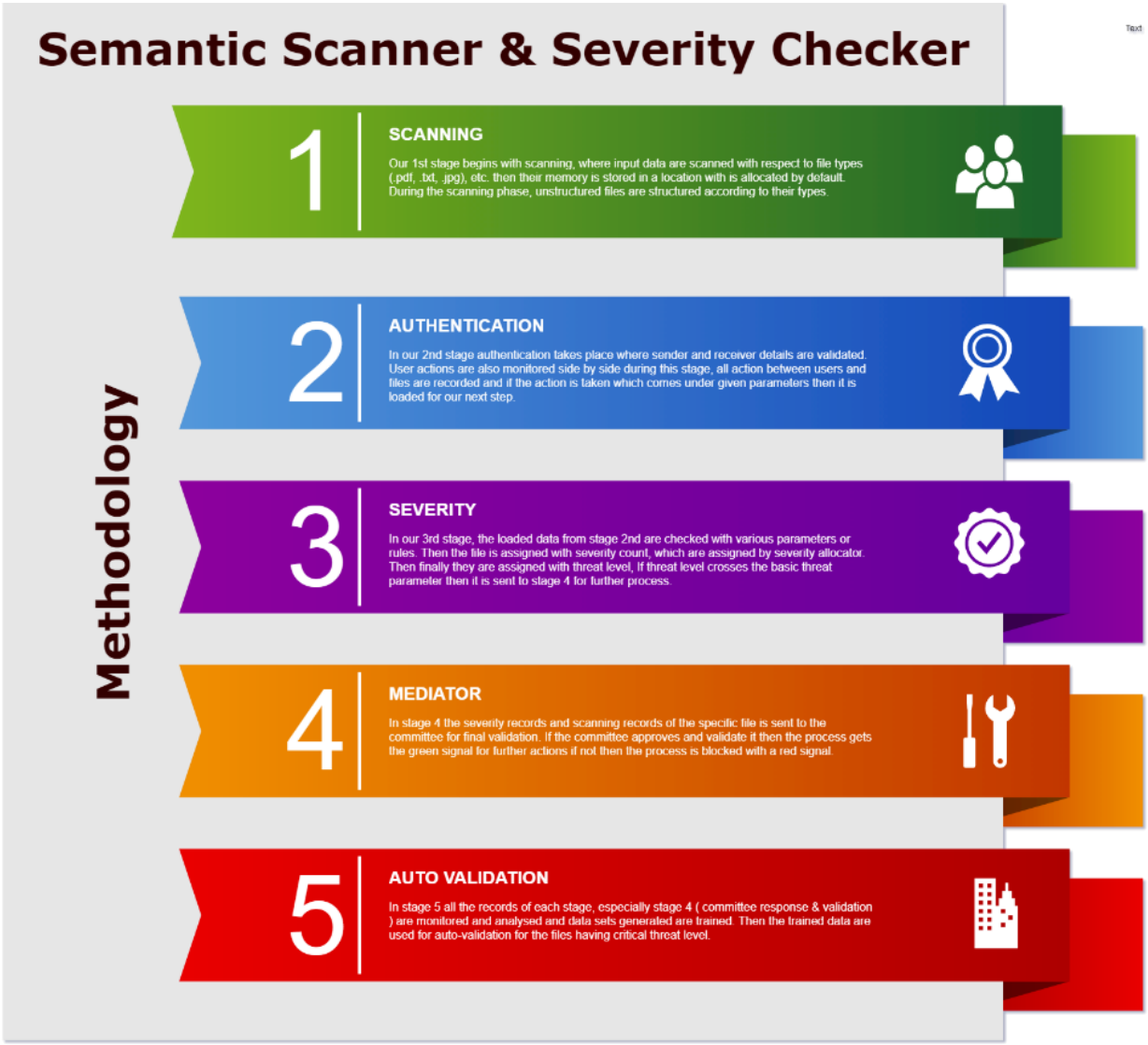


Fig. A

i. Scanning:

In the proposed Application for Severity Checking, the first stage of the Project was implementing the scanning of the file on the basis of its file type i.e. Text File, PDF, Word File, JPEG, PNG, JSON, etc. and on the basis of file size i.e. whether the file size is within the required conditions.

The Scanning Phase can be implemented either during the copying of files from some external device to the system or during the uploading of file from the system to the web application (using Flask). It can be done for both the cases (i.e. while copying or uploading) together also.

With reference to Fig.01, the file is being uploaded from an external device to the system by providing the path of the source from where the file is to be copied and the path to the destination to which the file is to be copied. In accordance to the Fig.04 the scanning is implemented during the uploading of file from the system to the web application. In both the cases whether copying or uploading, the file is transferred according to the extension of the file and the size of the file. With the help of the regular expression extension matching is done to find the file type.

In scanning process, we are using modules like shutil, subprocess for copying and modules like Flask for uploading the files according to the allowed extensions and the size.

ii. Authentication:

The second stage in severity checking application is Authentication. In this stage, there is a Web Application developed using Machine Learning's Web Development module of Flask and other languages like HTML and CSS. Using Flask, we develop a Mail Compose interface with various different fields including username, password, sender's email address, receiver's email address, message field, subject filed, file attachment and a submit button.

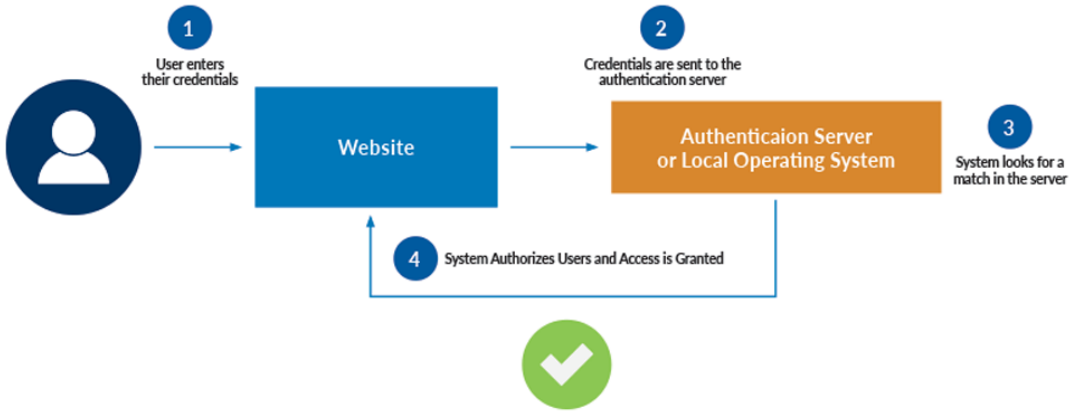
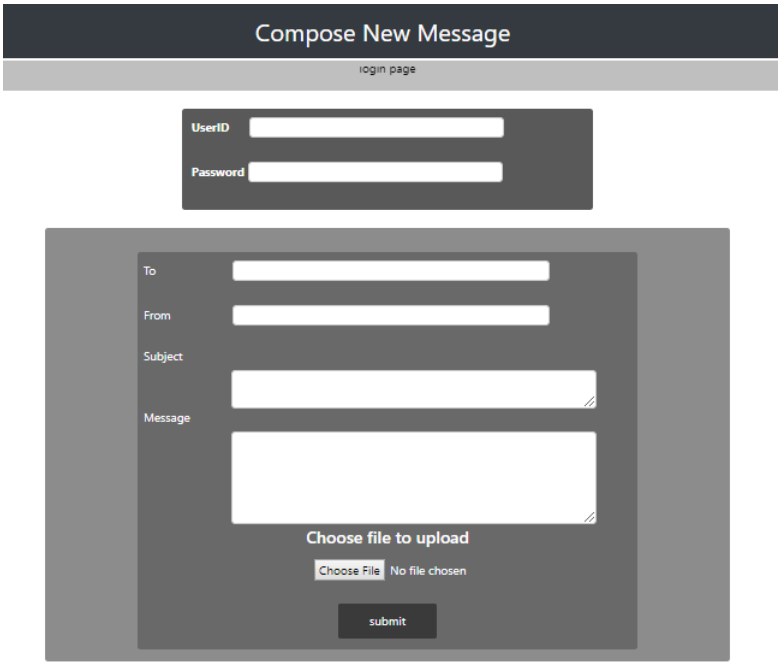


FIG. B

The information filled by the sender in the web application form is retrieved in the flask program, where the file that is uploaded after the scanning is forwarded to the next module for severity checking. All the information should be according to the rules and the regular expression defined. Regular expression is used to validate the email addresses of sender and the receiver.

The file content is sent for the severity checking i.e. next stage and the result is being returned through which the file is sent to the mediator or the sender on the basis of the result of the severity checker. For sending the file through mailing site we use the SMTP and MIME libraries to send the file from sender to receiver or from static backend user to the third-party validator.

In fig.09 as we see after authentication of the login form the file is sent for severity checking and according to the result mail is sent. For sending the message we use SMTP protocol and for receiving the message we use IMAP protocol. In fig.07, for sending the mail to the third party or the sender we use authenticator to validate the details and the mail is sent. When the credentials are entered in the website then for sending those credentials these credentials are sent to local operating server and system looks for match in the database as can be seen in the fig. A. After matching details in the server, the access is granted to user and the system authorizes the user.

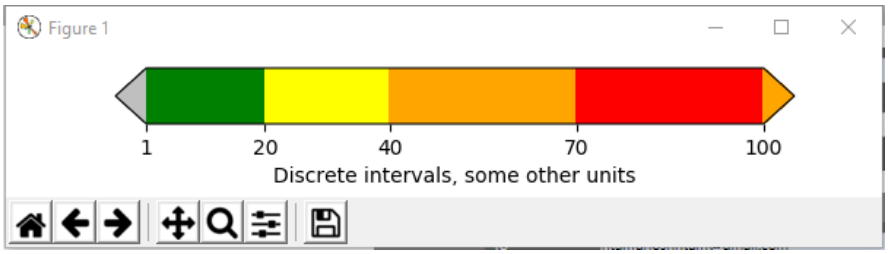


iii. Severity:

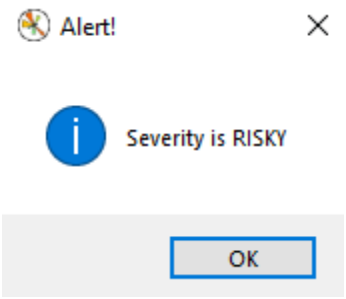
The stage that plays an important role in the proposed application is of Severity checking. In this, the file that is to be uploaded is sent for checking the severity level in it. In severity checking, two major part is of keyword matching and the semantic search on basis of the rules, keywords and the phrases defined earlier by the organization.

These rules that are defined should be according to the safety of the organization. If the file or the content that is sent over the web contains any threat to the organization, then the suitable action is to be performed to keep the important information from leaking.

In fig. 03 the file that is uploaded is sent for severity checking where the words in the file are appended to the list. The file is checked for the file type and all files are first converted to the text file and then further operations are carried on. Then the keywords that are defined is rules are matched and counted in the file, this count is then compared with the total count and a rough severity is calculated. Then we are provided with the phrases that may be treated as the threat. These phrases are then checked in the file through the semantic search and then final severity is calculated through the combination of the rough severity from the keyword matching and the semantic search.



This severity is then classified according to different percentages in terms of low, moderate, high and risky, then the output of the severity is send to the Flask application to take further action according to the severity of the file that is being uploaded as we can see in fig.04 the file is sent to the authenticator if the severity is high and then it is forwarded to the third party validator and if the severity is low then the authenticator authenticates and send the file directly to the receiver through the SMTP protocol.



In fig. 07, to check the severity we have imported various machine learning module like Regex, to find the words in the list of contents of the file using regular expression, Matplotlib, to plot the severity meter identifying the various values to indicate the type of severity for the file that is uploaded, Tkinter, is used to display the pop up message box indicating the severity of the file and Autocorrect, is used to correct the spelling of the words in the phrases or the sentences for the better match of the provided phrase according to the rules and the various sentences in the file that is being uploaded.

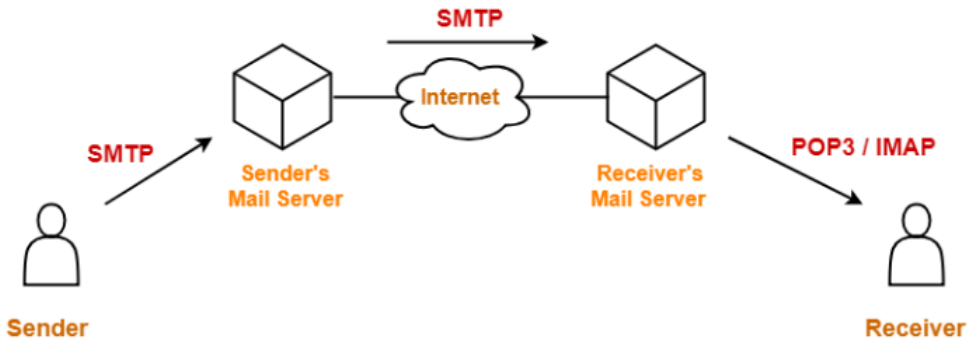
iv. Mediator:

The fourth stage of the severity checking application is Mediator. Here the mediator is referred to the third-party validation. The main application of this module is that the severe file that is being sent from the authenticator to the third-party validator is to be analyzed by the officials on the mediation level. This file is sent to the mediator through the local sharing network or through the web using the SMTP protocols over the port 465.

The response is to be generated and send back to the authenticator to validate the file send over the web according the true or false response. If the response is true then the file will be allowed to be sent over the web and if the response is false then the authenticator blocks the file to be sent over the web and the important information of the organization will be safeguarded from being leaked.

The response that is sent over the web to the authenticator is sent through the use of IMAP protocol over the port 993. This response is sent to the authenticator and the message is being retrieved from the inbox.

In the fig. 09 we can see that the authenticator validates and send the file the for the severity check. If the severity is low then the file is directly sent to the receiver through the SMTP protocol and if the severity is high then the file is sent to the mediator through SMTP protocol and the response is generated through the mediator and being sent to the authenticator using IMAP and then the mail is forwarded to the receiver or blocked according to the response sent from the third-party.



v. Auto Validation:

At last stage all the responses are analyzed and stored here and datasets are generated out of it, which are then trained using TensorFlow and keras. Before training period raw data are structured and grouped as per functions and similarities, then they go under different process such as data augmentation and normalization to generate suitable datasets. Then finally the datasets are trained and stored, which are used for providing auto responses. Due to continuous gathering and creating raw data to trained data the accuracy increases and loss decreases, which helps the system to provide auto validated response which are more accurate, fast and stable responses.

PROJECT PERSPECTIVE:

- ## Developers View:

As per developers' points of view our system can efficiently scan and generate report related to any file transfer and assign it with severity and only allow if severity is low or if it gets validated. Which in return can blocks sharing of sensitive files with unknown sources that could hinder the security and integrity of an organization.

- ## Users View:

As per user point of view, users can use our system, integrate it in their security and firewall module to safeguard organization sensitive data and to have all records of file transfer. It ensures the data being shared is within the required norms and doesn't violate any rules set also gets validate with response team of an organization.

PROJECT FUNCTIONALITY:

- ## Scanning:

  The System's basic functionality is to Scan the files that are inputted or present in the system. They are scanned with respect to their extensions and file size and file types.

- ## Severity:

  Severity is assigned and label as per given parameters and rules. Which enables us to differentiate between low and high severe files and helps us to take response accordingly.

- ## Training:

  Responses taken from mediator are stored in form of datasets which are used for training purposes for providing auto validation which are more accurate and efficient.

- ## Auto Validation:

  Our system provides manual and auto validation, which is more useful and accurate for faster response.

- ## Structure:

  It helps us to structure unstructured data, also to maintains a record of all files stored or transferred.

INTERFACE REQUIRED:

- ## HARDWARE INTERFACE:

Processor: Intel® Core™ i7-6500U CPU @ 3.0GHz - 3.3GHz

Installed memory (RAM): 16.00 GB

GPU: Nvidia K8

System type: 64-bit operating system, x64-based processor

- ## SOFTWARE INTERFACE:

Platform: OS - Windows 8 and above /Mac/Linux.

Tools: Colab, Anaconda Navigator, python libraries, web browser.

SYSTEM FEATURES:

- ## Platform Independent:

  Our program is platform independent; it can run on any platform having python libraries.

- ## Stable:

  Our code is stable can be run on any python compiler or can be run online without any difficulties.

- ## Adaptable:

  Our program can adapt itself with any pre-defined datasets, parameters or rules loaded or trained and can work according to it.

- ## Simple design:

  Our code is simple in design can be understand or use by anyone, who have basic knowledge of python.

- ## Maintainability:

  Our system can be change according to different parameters, rules & data sets and can be trained or used accordingly as per required.

- **Usability:**

  Our project main purpose is to use it in multiple sectors such as defense and security domain for all organizations to safeguard its data from being vulnerable or being compromised.

- **Faster:**

  Our system is faster as compared to other systems with a greater efficiency, gives rapid response.

- **Accurate:**

  Our program has more accuracy with great operating power.

- **Secure:**

  Our system uses SSL/TLS protocols during mail & data transfer which makes its more secure and safe to use.

CONCLUSION:

In this project “Semantic Scanner & Severity Checker” we use five stages scanning, authentication, severity, mediator and auto validation. In our project we initially begin with our scanning part then with authentication to validate details and to find and assign severity level if there is any process related to file transfer. Then to forward it to mediator committee for response and to train it for auto validation and response.

Here in this project we successfully be able to assign severity label to files during file transfer and also being able to communicate with mediator committee to get validated response. We are in continuity with our work to create an auto validation response by training response datasets.