T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

RANSOMWARE DETECTION

OĞUZHAN SEZGİN

SUPERVISOR
PROF. İBRAHİM SOĞUKPINAR

GEBZE
2022

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

RANSOMWARE DETECTION

OĞUZHAN SEZGİN

SUPERVISOR
PROF. İBRAHİM SOĞUKPINAR

2022
GEBZE

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

RANSOMWARE DETECTION

OĞUZHAN SEZGİN

SUPERVISOR
PROF. İBRAHİM SOĞUKPINAR

2022
GEBZE

GEBZE
TECHNICAL UNIVERSITY

GRADUATION PROJECT
JURY APPROVAL FORM

The thesis of Oğuzhan SEZGİN which was defended on 31/08/2021 in front of the jury formed by the decision of the board of Gebze Technical University, Faculty of Engineering number 2021/314 dated 31/04/2021, was accepted as Final Thesis in the field of "ercaseComputer Engineering Department.

JURY

Member (Supervisor)	:	Prof. İBRAHİM SOĞUKPINAR
Member	:	Prof. MEHMET GÖKTÜRK

APPROVAL

The decision of the board of Gebze Technical University, Faculty of Engineering number 2021/314 dated 31/04/2021.

Signature/Stamp

ABSTRACT

In this project, a method has been developed against attacks aiming to take over the system by sending ransomware to the victim with the Phishing method. The program first scans the mail folder specified by the user. As a result of scanning, it gives the URL of the mails with the URL in it to the model trained with machine learning techniques and moves it to another folder determined by the user according to the model's prediction. After scanning the previously received e-mails, it listens to the new incoming e-mails. If a new e-mail is received, it performs the above-mentioned operations for the new incoming e-mail. While training the program, 751192 data were used. Of these data, 223088 are malicious URL, while the remaining 528104 URL are benign. 30% of this data was used for testing and 70% for training. The accuracy value of the program is 89%.

Keywords: Ransomware, Phishing, Machine Learning, Logistic Regression.

ÖZET

Bu projede oltalama yöntemi ile kurbana fidye yazılım göndererek sistemi ele geçirmeyi hedefleyen saldırılara karşı bir yöntem geliştirilmiştir. Program öncelikle kullanıcı tarafından belirlenen mail klasörünü taramaktadır. Tarama sonucunda içerisinde URL bulunan maillerin, içerisindeki URL'i makine öğrenmesi teknikleri ile eğitilmiş modele vermekte ve modelin tahminine göre kullanıcı tarafından belirlenmiş başka bir klasöre taşımaktadır. Önceden gelmiş maillerin taramasının ardından yeni gelen mailleri dinlemektedir. Yeni bir mail gelmesi halinde yukarıda belirtilen işlemleri yeni gelen mail için gerçekleştirmektedir. Program eğitilirken 751192 veri kullanılmıştır. Bu verilerden 223088'i şüpheli URL iken kalan 528104 URL ise zararsızdır. Bu verilerin %30'u test için %70'i eğitim için kullanılmıştır. Programın doğruluk değeri %89 dur.

Anahtar Kelimeler: Fidye Yazılım, Oltalama Saldırısı, Makine Öğrenmesi, Logistic Regression.

ACKNOWLEDGEMENT

First of all, I would like to express my endless love to my family, who have supported me in every way throughout my life.

I would like to thank the lecturers of Gebze Technical University, who helped me grow as an engineer with the education and training provided throughout my school life. Especially I present my respect and love to my supervisor Prof. İbrahim SOĞKPINAR

Oğuzhan SEZGİN

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol or Abbreviation	:	**Explanation**
URL	:	Uniform Resource Locator
ML	:	Machine Learning
TP	:	True Positive
TN	:	True Negative
FP	:	False Positive
FN	:	False Negative

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Ransomware is a type of malware from cryptovirology that threatens to publish the victim's personal data or perpetually block access to it unless a ransom is paid. While some simple ransomware may lock the system so that it is not difficult for a knowledgeable person to reverse, more advanced malware uses a technique called cryptoviral extortion. It encrypts the victim's files, making them inaccessible, and demands a ransom payment to decrypt them [1]–[4]. In a properly implemented cryptoviral extortion attack, recovering the files without the decryption key is an intractable problem – and difficult to trace digital currencies such as paysafecard or Bitcoin and other cryptocurrencies that are used for the ransoms, making tracing and prosecuting the perpetrators difficult [5].

There are two basic types of ransomware available in the wild: the first one, locker-ransomware, is designed to lock the victims' computer, to prevent them from using it; the second one, and most common nowadays, is crypto-ransomware, which encrypts personal files to make them inaccessible to its victims. In both cases, users are forced to pay a ransom to regain access either to their data (assuming no backup mechanism is in place) or system [6].

Ransomware uses asymmetric encryption. This is cryptography that uses a pair of keys to encrypt and decrypt a file. The public-private pair of keys is uniquely generated by the attacker for the victim, with the private key to decrypt the files stored on the attacker's server. The attacker makes the private key available to the victim only after the ransom is paid, though as seen in recent ransomware campaigns, that is not always the case. Without access to the private key, it is nearly impossible to decrypt the files that are being held for ransom.

Often ransomware (and other malware) is distributed using email spam campaigns or through targeted attacks. Malware needs an attack vector to establish its presence on an endpoint. After presence is established, malware stays on the system until its task is accomplished [7].

There are mainly seven steps in the lifecycle of ransomware, as shown in Figure 1. The lifecycle shows the formation of a cybercriminal ecosystem [8].(Figure 1.1).

Examples of important ransomware known today are the following ransomware:

- Reveton
- CryptoLocker
- CryptoWall
- WannaCry

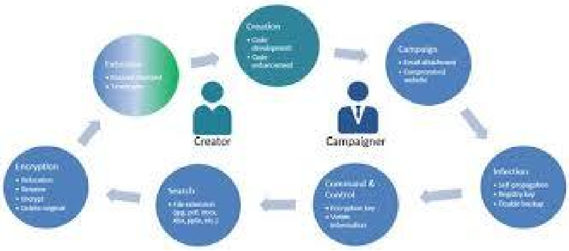


Figure 1.1: Ransomware lifecycle

To install ransomware on user's computer, several techniques are frequently used by cyber-criminals, such as [6]:

- Phishing or SPAM e-mails
- Exploit kits
- Downloader and Trojan Botnets
- Social engineering tactics
- Traffic Distribution Systems (TDS)

Ransomware has been a severe cyber threat for about twenty five years [9], [10]. Ransomware was first seen in 1989 under the name of AIDS Trojan horse [11]. The first modern Ransomware "Trojan.Gpcoder" has been seen in Russia in 2005 [12]. The Trojan.Gpcoder, which was first seen in May 2015, has been easily overcome since it had a simple and easy encryption. In time, improved versions of Ransomware were found to use the user's native language, and even some versions were found to contain voicemails in the user's native language [13]. In 2008, a Trojan.Gpcoder Ransomware called GPcode.AK emerged. It has been found that GPcode.AK uses a 1024 bit RSA key and leaves a text file containing instructions in each subdirectory of encrypted files. GPcode.AK has requested a $100 payment to decrypt the encrypted files of the victims [12], [14].

This article describes the work done to detect ransomware before it starts working. The ransomware URL in the messages sent to the victim are analyzed to determine whether the ransomware is present. While doing this, it uses ML techniques.

2. RELATED WORK

The objective of ransomware analysis is to better understand how ransomware functions. Based on this understanding, defensive steps can be formulated to prevent future infections. Two types of analysis can be performed: static analysis and dynamic analysis, Static analysis is based on the source code of the executable file. For dynamic analysis, the ransomware is executed in a controlled environment, and all its actions are recorded for analysis [15].

Static analysis can be conducted quickly by examining the features of an executable piece of code and matching it to a previously observed malicious code.

Dynamic analysis is also called behavioural-based analysis. Malicious code is executed in a controlled and monitored environment, usually a sandbox. All actions are captured for analysis.

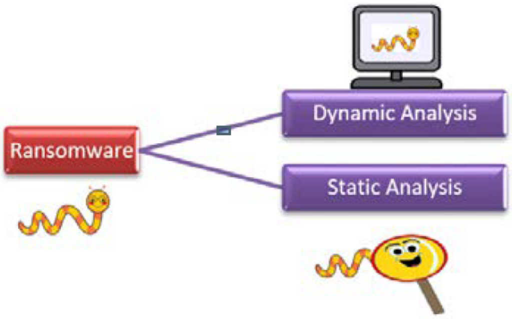


Figure 2.1: Types of ransomware analysis

2.1. MACHINE LEARNING

The advantage of ML is that it can accurately predict the outcome with adequate training data. Training data should be varied with balanced distribution of outcomes to be predicted. Because ML involves learning the pattern in the data, it is less prone to obfuscation but Finding the correct algorithm is often not straightforward and may require some runs of trial and error[15].

2.2. HONEYPOT

Honeypot involves setting up decoy files for the ransomware to attack. Once these files are accessed, the ransomware can be identified.

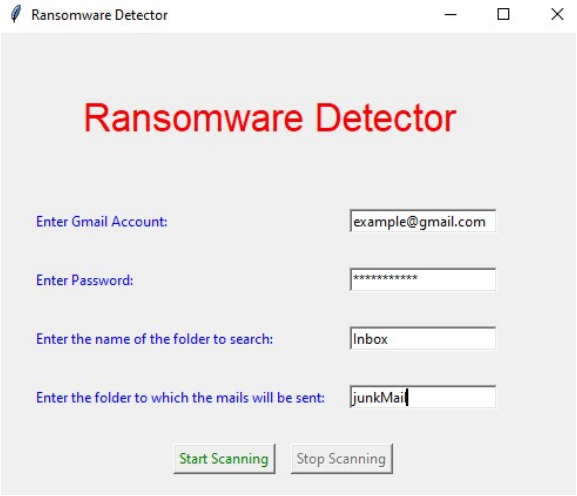
The traps or honeypot files can be set up, and then they simply wait to be attacked. Therefore, the technique does not require much maintenance or processing power from the system. There is no guarantee that the honeypot files will be attacked by a ransomware.

Several works have been done in Malware Structural Analysis using Dyanmic and Static Analysis.[16] presents the analysis of malware performing assembly code analysis to identify and classify the malware. Another work focused on malware using PE file structure analysis [17]. Taxonomy based [18] present an approach for preventing and detecting ransomware. Mercaldo et al. [19] used a static analysis method on Android system to automatically process ransomware sample. They performed with the goal of observing the malicious behavior. Modern techniques such as Software Defined Networking (SDN) have also be used to detect and mitigate the ransomware [20], [21]. Another technique such as honeypot based detection [22] was also used. In addition, recovery technique such as [23] has been done to defend against ransomware attack. There are approaches to deal with ransomware such as File Hashes (Full or Portion), Byte Signatures, System Behavior, and Network Signatures [24]. These approaches are used by different antivirus engines, security tools, intrusion detection systems etc. The main problem with these approaches is that it is very easy to perform evasion. For example, File hashes can easily be bypassed by changing the equivalent assembly instructions. Similarly, Byte Signatures and network signatures are also prone to the same problem. System behavior is a unique approach but it requires the run time behavior of ransomware which is very difficult to capture [25].

3. ANALYSIS AND DESIGN

In this project, a method was developed against attacks aiming to take over the system by sending ransomware to the victim with the Phishing method.

Before the program was started, the features that the program would be used for while training were extracted. Information about these features is explained in the **Features** section. The program makes predictions about the URL by using this previously prepared data. When the program is started, the user interface opens (Figure 3.1). The user must enter the necessary information in the relevant place (the folder where the search will be made and the folder where the mails will be sent after the search is done must be on the mail server).



Ransomware Detector

Enter Gmail Account: example@gmail.com

Enter Password: ***********

Enter the name of the folder to search: Inbox

Enter the folder to which the mails will be sent: junkMail

Figure 3.1: Program's user interface

After entering the information, the 'Start Scan' button is pressed.The scan thread starts running when the 'Start Scan' button is pressed. While scanning, it takes the body part of the incoming mail and checks whether there is a URL in it. If it finds the URL, it gives it to the logistic regression model, which was trained with machine learning techniques before. The program first scans the previously sent e-mails in the entered e-mail folder and if it finds a suspicious URL in the e-mail, it sends the specified e-mail to the specified folder. After the scanning of previously incoming mails is finished, it switches to listening to the mail server to check new incoming mails. In the case of a new mail, it checks whether there is a URL in the content of the mail as it did with the previous ones, and if it finds it, it gives it to the model and according to the result, it moves the mail to the folder determined by the user or not. After the scan starts, the 'Stop Scan' button becomes active. After the scanning process is stopped, it can be started again without closing the program.

The design plan of the project is shown in Figure 3.2.

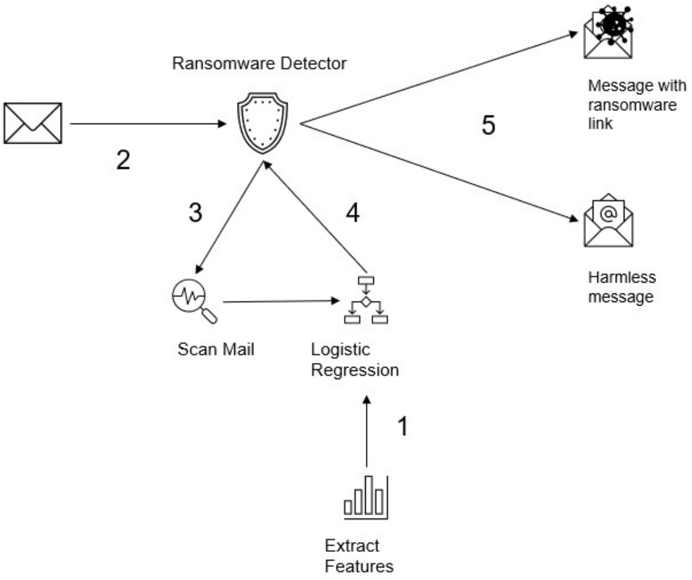


Figure 3.2: Project design

3.1. REQUIREMENTS

The elements used in creating the program are as follows;

- **Gmail server**
- **Excel**
- **Kaggle**
- **Python Libraries**
  - pandas
  - numpy
  - os
  - urllib
  - tld
  - ipwhois
  - pydnsbl
  - socket
  - tkinter
  - seaborn
  - sklearn

  - math
  - threading
  - imaplib
  - email
  - time

3.2. FEATURES

In the light of the information obtained from the studies in the literature, the features that should be used have been determined as follows [26]:

- **Lexical features:** The justification for using lexical features is that URLs to malicious sites tend to “look different” in the eyes of the users who see them. Hence, including lexical features allows us to methodically capture this property for classification purposes, and perhaps infer patterns in malicious URLs that we would otherwise miss through ad-hoc inspection.
- **Host-based features:** The reason for using host-based features is that malicious Web sites may be hosted in less reputable hosting centers, on machines that are not conventional web hosts, or through disreputable registrars. To an approximate degree, hostbased features can describe “where” malicious sites are hosted, “who” own them, and “how” they are managed.

In this study, only lexical features were used.The list and description of the extracted features are as follows (Figure 3.3);

- **url_length :** Length of URL
- **hostname_length :** Hostname length of URL
- **path_length :** Path length of URL
- **tld_length :** Top level domain length
- **n- :** Number of ’-’
- **n_ :** Number of ’_’
- **n@ :** Number of ’@’
- **n? :** Number of ’?’
- **n% :** Number of ’%’
- **n. :** Number of ’.’

- **n= :** Number of ’=’
- **n-http :** Is there ’http’
- **n-https :** Is there ’https’
- **n-www :** Is there ’www’
- **n-digits :** Number of digits
- **n-letters :** Number of letter
- **use_of_ip :** Is there IP number
- **short_url :** Is used short URL
- **n_param :** Number of paramater
- **entropy :** Shannon entropy of URL
- **login :** Is there ’login’ keyword
- **server :** Is there ’server’ keyword
- **admin :** Is there ’admin’ keyword
- **client :** Is there ’client’ keyword

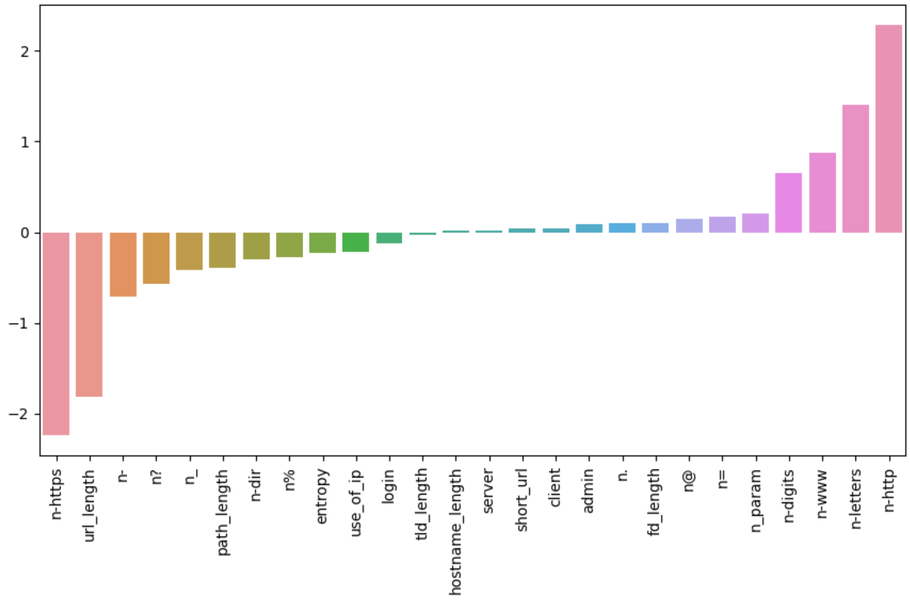


Figure 3.3: Feature coefficients

3.3. DATASET

Datasets with harmless and malicious URLs were used to train the program. For this, datasets from Kaggle [27], PhishTank [28] (dataset with suspicious urls) were used.
There are 751192 pieces of data in the dataset. This data is 223088 malicious URL and the remaining 528104 URL is benign URL.

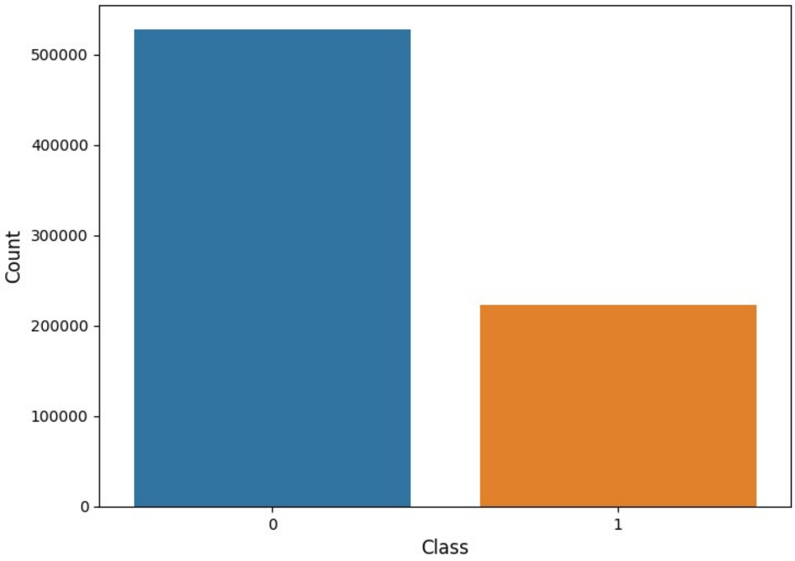


Figure 3.4: Dataset

3.4. ANALYZING URL

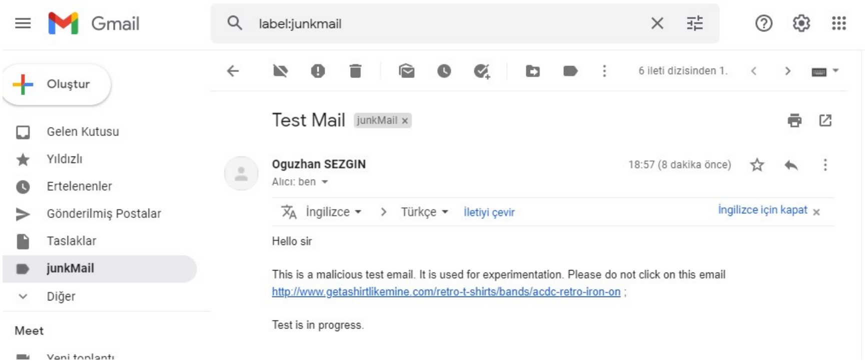
Logistic regression method, which is one of the machine learning techniques, was used while analyzing the URL. The features extracted from the training set were given to the logistic regression model. Afterwards, the data obtained from the scanned e-mails were given to the trained model as input and action was taken according to the result.

3.4.1. Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression[29] (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name. The unit of measurement for the log-odds scale is called a logit, from logistic unit, hence the alternative names. Analogous models with a different sigmoid function instead of the logistic function can also be used, such as the probit model; the defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a constant rate, with each independent variable having its own parameter; for a binary dependent variable this generalizes the odds ratio[30].

4. IMPLEMENTATION AND TEST

First of all, the python programming language was used within the scope of this project. General information about the operation of the program is explained in the ANALYSIS AND DESIGN section. In addition to this information, images of the operation of the program are shown in the figures below.



Test Mail junkMail ×

Oguzhan SEZGIN 18:57 (8 dakika önce)

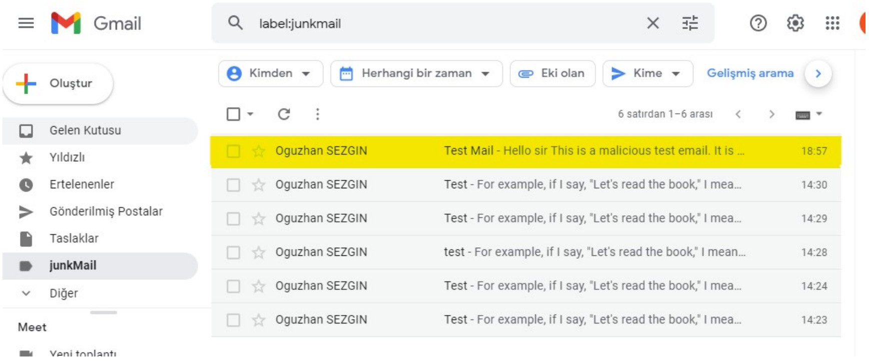
Alıcı: ben

Hello sir

This is a malicious test email. It is used for experimentation. Please do not click on this email http://www.getashirtlikemine.com/retro-t-shirts/bands/acdc-retro-iron-on ;

Test is in progress.

Figure 4.1: Email with suspicious URL

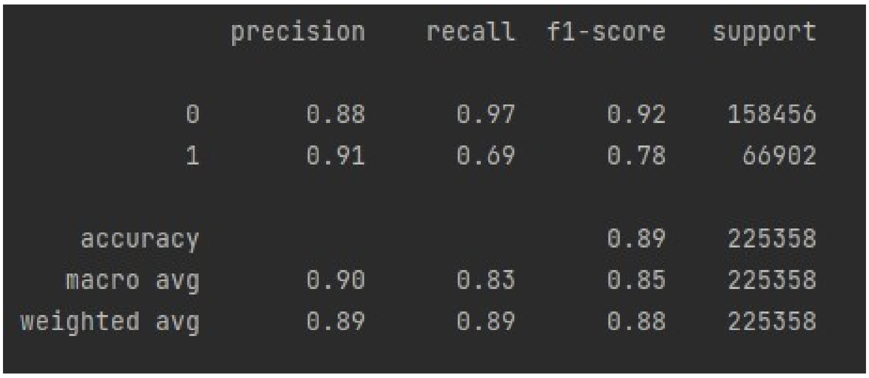


Oguzhan SEZGIN	Test Mail - Hello sir This is a malicious test email. It is ...	18:57
Oguzhan SEZGIN	Test - For example, if I say, "Let's read the book," I mea...	14:30
Oguzhan SEZGIN	Test - For example, if I say, "Let's read the book," I mea...	14:29
Oguzhan SEZGIN	test - For example, if I say, "Let's read the book," I mean...	14:28
Oguzhan SEZGIN	Test - For example, if I say, "Let's read the book," I mea...	14:24
Oguzhan SEZGIN	Test - For example, if I say, "Let's read the book," I mea...	14:23

Figure 4.2: ’JunkMail’ folder

The mail shown in Figure 4.1; was sent before the program was running (It gives the same result in the mails sent while the program is running). After the program ran, it detected the suspicious URL in the mail and moved it to the ’junkMail’ folder determined by the user. In Figure 4.2, the folder is shown after the migration is performed.

30% of the data set was used for testing. The outputs obtained as a result of the tests made with these data are given in the figures.



```
              precision    recall  f1-score   support

           0       0.88      0.97      0.92    158456
           1       0.91      0.69      0.78     66902

    accuracy                           0.89    225358
   macro avg       0.90      0.83      0.85    225358
weighted avg       0.89      0.89      0.88    225358
```

Figure 4.3: Accuracy of program

- **Accuracy:** Accuracy is defined as the ratio of true positives and true negatives to all positive and negative observations.
- **Precision:** Precision is defined as the ratio of true positives to the sum of true and false positives.
- **Recall:** Recall is defined as the ratio of true positives to the sum of true positives and false negatives.
- **F1 Score:** The F1 is the weighted harmonic mean of precision and recall. The closer the value of the F1 score is to 1.0, the better the expected performance of the model is.

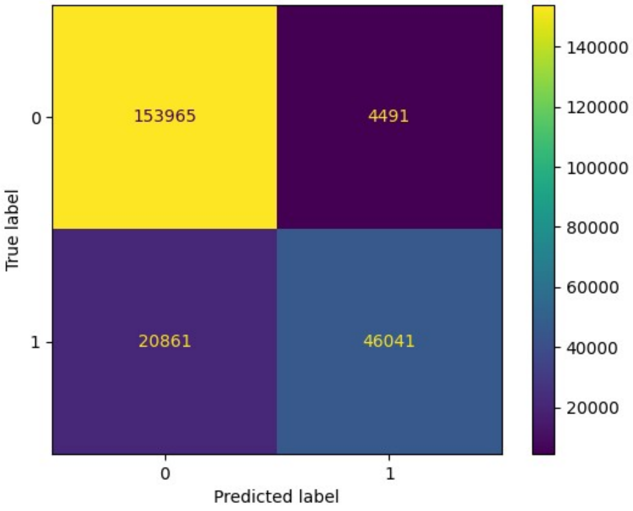


Figure 4.4: Confusion matrix

In this matrix;

- The value specified in the upper left corner shows the number that the URL is benign and the program predicts as benign (TN).
- The value specified in the upper right corner shows the number that the URL is benign and the program predicts as malicious (FP).
- The value specified in the lower left corner indicates the number that the URL is malicious and the program predicts as benign (FN).
- The value specified in the lower right corner shows the number that the URL is malicious and the program predicts as malicious (TP).

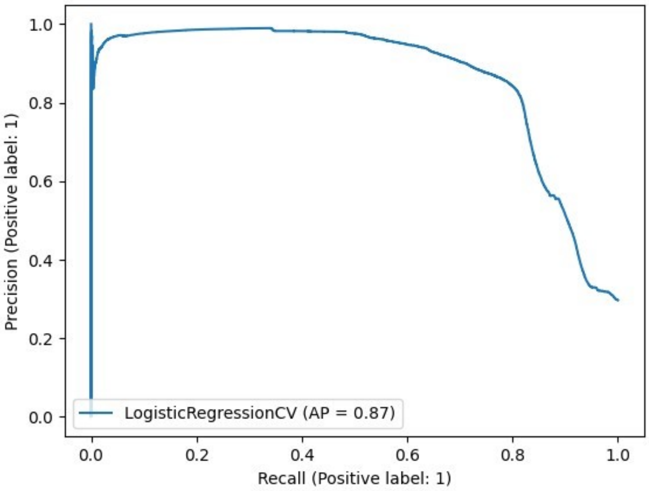


Figure 4.5: Precision-recall graph

Precision-Recall curves summarize the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds.

5. CONCLUSIONS

In this project, a method has been developed against attacks aiming to take over the system by sending ransomware to the victim with the phishing method. The program first scans the mail folder and then waits for the incoming mails. It moves the e-mails that it detects with malicious URLs to another specified e-mail folder.

751192 data were used in the testing and training of the program. While 223088 of these data are suspicious URLs, the remaining 528104 URLs are harmless. %30 of this data was used for testing, %70 for training. As stated in the confusion matrix, as a result of the test, the TP value is 46.041, the TN value is 153.965, the FP value is 4491, and the FN is 20861. While the total number of correct predictions is 200006, the number of incorrect predictions is 25352.The accuracy of the program is %89. Host-based features can be included in the program to increase the accuracy of the program.

BIBLIOGRAPHY

[1] A. M. Y. Young, “Cryptovirology: Extortion-based security threats and countermeasures,” *IEEE Symposium on Security and Privacy*, pp. 129–140, 1996.

[2] J. Schofield, “How can i remove a ransomware infection?” *The Guardian*, 28 July 2016.

[3] M. Mimoso. “Petya ransomware master file table encryption.” (28 March 2016), [Online]. Available: `threatpost.com`.

[4] J. Luna, “Mamba ransomware encrypts your hard drive, manipulates the boot process,” *Neowin*, 21 September 2016.

[5] Wikimedia Foundation. “Ransomware.” (2021, November 2), [Online]. Available: `https://en.wikipedia.org/wiki/Ransomware`.

[6] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, “Automated dynamic analysis of ransomware: Benefits, limitations and use for detection,” 2016.

[7] “Business home. mcafee. (n.d.)” (2021, November 2), [Online]. Available: `https://www.mcafee.com/enterprise/en-us/security-awareness/ransomware.html.`.

[8] “Ransomware, threat and detection techniques,” *A Review*,

[9] J. Z. Kolter and M. A. Maloof., “Learning to detect and classify malicious executables in the wild,” *The Journal of Machine Learning Research*, 2006.

[10] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection,” *ACM Comput. Surv.*, no. 41, 2009.

[11] H. L. K. Savage and P. Coogan, “The evolution of ransomware,” *Symantec*, 2015.

[12] K. Rieck, P. Trinius, C. Willems, and T. Holz, “Automatic analysis of malware behavior using machine learning,” *J. Comput. Secur.*, no. 19, Dec. 2011.

[13] Richardson, R., North, and M, “Ransomware: Evolution, mitigation and prevention,” *International Management Review*, no. 13, 2017.

[14] Kara, İ., Aydos, M., and Bozkır, “Characteristic behavioral analysis of malware: A case study of cryptowall ransomware,” *Avrupa Bilim ve Teknoloji Dergisi*, pp. 486–493, 2020.

[15] D. Distler, “Malware analysis : An introduction,” *SANS Institute,*,

[16] D. Bilar, “Fingerprinting malware for classification and analysis,” *Proceedings of Black Hat Federal*, 2006.

[17] J. H. Yang and Y. Ryu., “Design and development of a commandline tool for portable executable file analysis and malware detection in iot devices,” *International Journal of Security and Its Applications*, pp. 127–136, 2015.

[18] M. M. Ahmadian, H. R. Shahriari, and S. M. Ghaffarian, "Connectionmonitor connection-breaker: A novel approach for prevention and detection of high survivable ransomwares. in information security and cryptology (iscisc)," *12th International Iranian Society of Cryptology Conference,IEEE*, pp. 79–84, 2015.

[19] F. Mercaldo, V. Nardone, and A. Santone., "Ransomware inside out. in availability, reliability and security (ares)," *11th International Conference IEEE*, pp. 628–637, 2016.

[20] K. Cabaj, M. Gregorczyk, and W. Mazurczyk., "Software-defined networking-based crypto ransomware detection using http traffic characteristics," *arXiv*, 2016.

[21] K. Cabaj and W. Mazurczyk, "Using software-defined networking for ransomware mitigation: The case of cryptowall," *IEEE Network*, pp. 14–20, 2016.

[22] C. Moore, "Detecting ransomware with honeypot techniques," *Cybersecurity and Cyberforensics Conference (CCC) IEEE*, pp. 77–81, 2016.

[23] K. P. Subedi, D. R. Budhathoki, B. Chen, and D. Dasgupta, "Ransomware defense strategy by using stealthily spare space," *In Computational Intelligence (SSCI), 2017 IEEE Symposium Series*, pp. 1–8, 2017.

[24] M. Stevanovic, "Linux toolbox. in advanced c and c++ compiling," *Springer*, pp. 246–276, 2014.

[25] Subedi, K. P., Budhathoki, and D. R., "Forensic analysis of ransomware families using static and dynamic analysis.," *IEEE Security and Privacy Workshops (SPW).*, 2018.

[26] Ma, J. Saul, L.K., *et al.*, "Beyond blacklists: Learning to detect malicious web sites from suspicious urls.," *In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1245–1253, 28 June–1 July 2009.

[27] "Your machine learning and data science community." (November 9, 2021), [Online]. Available: `https://www.kaggle.com/`.

[28] "Join the fight against phishing. phishtank." (November 9, 2021), [Online]. Available: `https://phishtank.org/`.

[29] Tolles, J. Meurer, and W. J, "Logistic regression relating patient characteristics to outcomes," 2016.

[30] "Logistic regression." (30 December 2021), [Online]. Available: `https://en.wikipedia.org/wiki/Logistic_regression`.