Journal of Engineering Sciences
Vol 13, Issue 04, APRIL/2022
ISSN NO:0377-9254

SQL INJECTION PREDICTION WEB APP USING DIFFERENT MACHINE LEARNING ALGORITHMS

Vinod Babu Polinati[1], Sahithya Chowdary Nekkalapudi[2], Naidu Sai Sanjana[3], Karthik Kundula[4] ,Rahul Varma Bhupathiraju[5]

#1Associate Professor, Department of CSE, GITAM (deemed to be University), Gandhi nagar Rushikonda Visakhapatnam 530045 Andhra Pradesh, INDIA

#2,#3,#4,#5 Student, Department of CSE, GITAM (deemed to be University), Gandhi nagar Rushikonda Visakhapatnam 530045 Andhra Pradesh, INDIA

ABSTRACT

As online applications get more complex and interconnected, limiting application vulnerabilities becomes increasingly important. SQLIA (SQL Injection Attack) is a serious OWASP (Open Web Application Security Project) flaw that must be avoided at all costs. The proposed method attempts to foresee the occurrence of SQL Injection Attacks on a certain server with deployed apps from a specific source at a specific time. To make this prediction, machine learning methods are applied, which are often more convenient and accurate than traditional attack detection systems. This is accomplished by simulating log data. This can be used to pre-process data, extract features, and categorise it before it is fed into a logistic regression model for SQL Injection Attack prediction.

1.INTRODUCTION

It is no secret that over the closing decade, the world has come to be more and more reliant on technology. Many motives have contributed to an extend in the use of web-based apps to allow get admission to to a range of offerings via agencies and individuals. Insecure software, on the different hand, is jeopardizing our security-critical settings, such as banking, healthcare, military, and energy. The necessity of securing software protection grows notably as purposes get greater state-of-the-art and networked. It's imperative to cast off utility vulnerabilities and make them extra resistant to assaults.

A device defect or weak spot in a web-based programme is referred to as a net software vulnerability. They've been round for a long time, thanks to a lack of validation and sanitization of shape inputs, as nicely as misconfigured net servers

and utility diagram problems. Validation verifies that the enter fulfills a set of necessities (such as a string includes no standalone single citation marks). The enter is sanitized to make sure that it is official (such as doubling single quotes).

SQL is used to cope with the facts on many of the servers that keep necessary facts for web sites and services. An SQL Injection Attack (SQLIA) is a kind of assault that makes use of malicious code to trick the server into divulging statistics it would not usually reveal. Successful SQLIA commonly takes place when a inclined utility fails to appropriate sanitize the user's input. In a Cross Site Scripting (XSS) attack, malicious code is additionally injected into a website. An attacker can use cross-site scripting to run hazardous programmes in any other user's browser. Injecting malicious code into an enter area is one of the most famous techniques an attacker may launch a cross-site scripting attack, which will be routinely achieved when extra customers see the contaminated website.

The question facts is bought by way of a publish request, and characteristic extraction is carried out after preprocessing. The logistic regression mannequin used to be used to make the prediction. The first half of the article discusses SQLIA's applicable work, whilst the 2nd area discusses the proposed device and its implementation. Finally, we debate the findings earlier then coming to a conclusion.

2.LITERATURE SURVEY

2.1 Arumugam,C.*etal.*(2019).PredictionofSQLInjectionAttacksinWebApplications. In: *,et al.* Computational Science and Its Applications – ICCSA 2019.ICCSA2019.LectureNotesinComputerScience(),vol11622.Springer,Cham.

As online applicationsgetmore complicatedandinterconnected,it'smore important thanever to prevent application vulnerabilities. SQLIA is one of the OWASP vulnerabilities thatmustbeavoided atall costs. The suggested approach seeks toforecast theincidence ofSQLIA on a given server with deployed apps from a specific source at a given time. Thepaper's main goal is to forecast SQLIA in online applications. To generate log data, ApacheJMeter, an open source programme, was utilized as a load generator. The preparation of logdata is done in order to extract features. The logistic regression model was used to make theprediction.

2.2 **Scholte, T., Robertson, W., Balzarotti, D., Kirda., E.: An empirical analysis of inputvalidation mechanisms in web applications and languages. In: 27th Annual ACMSymposiumonAppliedComputing,pp.1419–1426(2012)**

Web apps have become an important component of millions of people's everyday lives.Unfortunately, attackers routinely target online applications, and attacks like XSS and SQLinjection are still popular. We offer an empirical research of over 7000 inputvalidationvulnerabilities in this work, with the goal of learning more about how to mitigate theseprevalentonlinevulnerabilities.Weconcentrateonthelinkbetweentheprogrammin glanguage used to construct web apps and the most often reported vulnerabilities. Our findingsimply that utilizing simple validation techniques based on common data types, most SQLinjection andaconsiderablepercentageof XSSvulnerabilitiesmaybeavoided.Wegothrough these popular data types in detail and examine how web application frameworks mayhandle them

2.3 Alkhalaf, M., Aydin, A., Bultan, T.: Semantic differential repair for input validationandsanitization.In:ACMInternationalSymposiumonSoftwareTesting andAnalysis,pp.225–236(2014)

In order to minimize security risks and erroneous application behavior, web applications mustvalidate and sanitize user input correctly. For input validation and sanitization functions, weprovide an automatic differential repair approach. Differential repair can be used inside anapplication tofix clientand server-side code in relation to one another, or itcan be usedacross apps to improve validation and sanitization checks.Our differential repair approach,givenareferenceandatargetfunction,improvesthetargetfunction'svalidationa ndsanitization processes by using the reference function. This is accomplished by combiningthree patches: validation, length, and sanitization. Our automatic patch synthesis techniquesare based on symbolic string analyses that employ automata as a symbolic representation inboth forward and backward directions. The repaired function is created by combining thethree automatically synthesized patches with the original target function, which providesbettervalidationandsanitizationthanthe targetandreferencefunctions.

3.PROPOSED SYSTEM

As online applications get more complex and interconnected, limiting application vulnerabilities becomes increasingly important. SQLIA (SQL Injection Attack) is a serious OWASP (Open Web Application Security Project) flaw that must be avoided at all costs. In their research, Arumugam, C. et al. employed an inbuilt apache tool to create a dataset, which was then fed into a logistic regression method to produce classification results. Instead of utilizing an existing tool, we created a bespoke web application and then incorporated certain components. For the purpose of prediction, a dataset was gathered from research that included the most commonly used sql attack queries. The authors in the existing study employed a dataset that solely contained login questions, however the dataset used in this study is made up of a variety of queries.

3.1 IMPLEMENTAION

In this projectinstead of usinganinbuilttool we havebuiltacustom web application andthenthefollowingmodules areimplemented.

1) Admin: Admin can login to application by using username as ‘admin’ and password as‘admin’andthenperformbelowsteps

2) UploadDataset:usingthismoduleadmincanuploaddatasetofSQLInjectionqueries

3) Preprocess dataset: using this module dataset will be preprocess to remove emptyvaluesandthenselectfeatures fromdataset

4) Generate Prediction/Classification Model: Using this application will build logisticregressionclassificationmodelbyusingabovedataset

5) Detect attack: using this module, the user can enter queries and then the applicationwill apply that query on a regression model to predict it as a normal or abnormal attackquery.

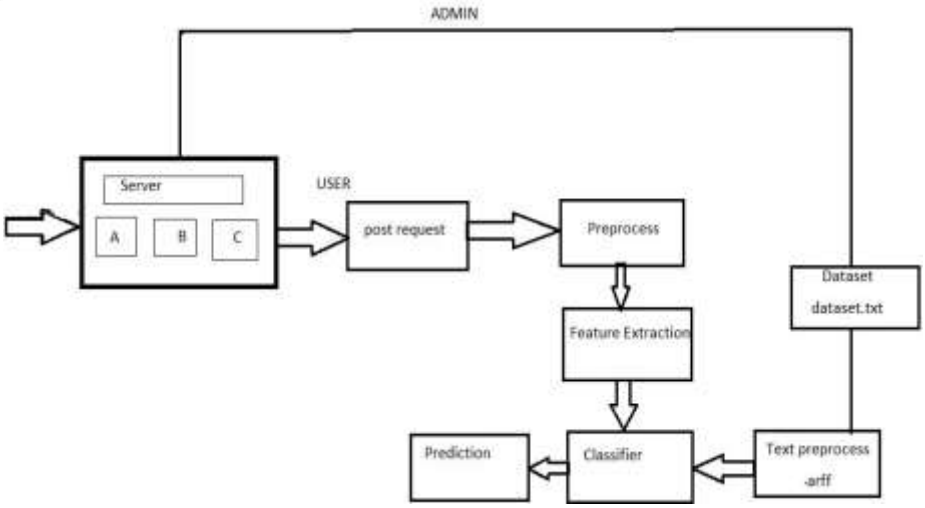

Fig 1:System Architecture

3.2 DATASET

dataset - Notepad

File Edit Format View Help

```
payload,label
select * from emp@norm
select * from emp where name='a'@norm
select * from emp where name='a' and dept='economy'@norm
select count(*) from emp@norm
select count(*) from emp where dept='maths'@norm
select count(*) from emp where dept='english'@norm
select count(*) from emp where dept='hindi'@norm
1))) and 6969=(select 6969 from pg_sleep(5)) and (((5333=5333@anom
1' and 7528=2894 and 'qoyw'='qoyw@anom
1));waitfor delay '0:0:5'--@anom
1') where 2330=2330 and (select * from (select(sleep(5)))fzno)--@anom
1)) and 2006=2006@anom
-2749') as fiho where 3531=3531 or 2777=1485#@anom
-4826 union all select 4532,4532,4532,4532,4532--@anom
-1643' where 1968=1968 or 3484=6642@anom
-4586') as shdb where 6176=6176 union all select 6176,6176,6176,6176#@anom
1'+(select 'rpds' where 5870=5870 and 4595=4595#@anom
1' and 3202=like('abcdefg',upper(hex(randomblob(500000000/2))))@anom
-6600') union all select 5566,5566#@anom
-5145' where 2334=2334 union all select 2334,2334,2334,2334--@anom
1') union all select null,null,null,null#@anom
1') waitfor delay '0:0:5' and ('fpnh'='fpnh@anom
1%' union all select null,null,null,null,null,null,null,null,null--@anom
1%') and 6969=(select 6969 from pg_sleep(5)) and ('%'='@anom
1') as elnu where 5719=5719 or 8156=(select count(*) from generate_series(1,5000000))--
@anom
1) where 1402=1402;select sleep(5)#@anom
1') as elnu where 5719=5719 or 8156=(select count(*) from generate_series(1,5000000))--
@anom
1) where 1402=1402;select sleep(5)#@anom
Select * from table where first_name like '%kumar'#@anom
Select * from deposit_table where account_no='1234 or 1==1'#@anom
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Fig 2:In above dataset.txt we can see all queries at the end each query is mark with label as ‘norm’ or ‘anom’ where norm represents normal query and anom represents abnormal query.

4.RESULTS AND DISCUSSION

Classifier	Prediction Accuracy	precision	F-Measure	ROC
Logistic Regression	96.667%	0.971	0.967	0.915
Multilayer Perceptron	90%	0.907	0.902	0.988
SVM	90%	0.907	0.902	0.885

Comparison of different classification algorithms accuracy on our dataset:

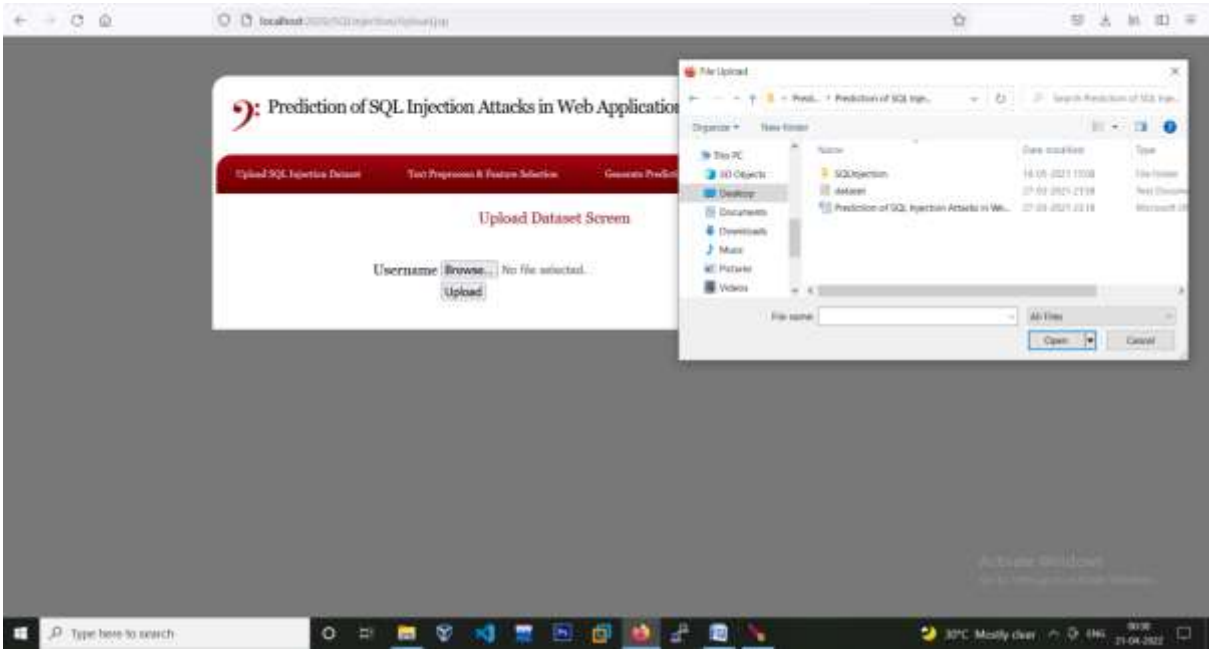

Fig 3:In above screen click on ‘Choose File’ button and then select ‘dataset.txt’ file and then click on ‘Open’ button to load dataset and to get below screen

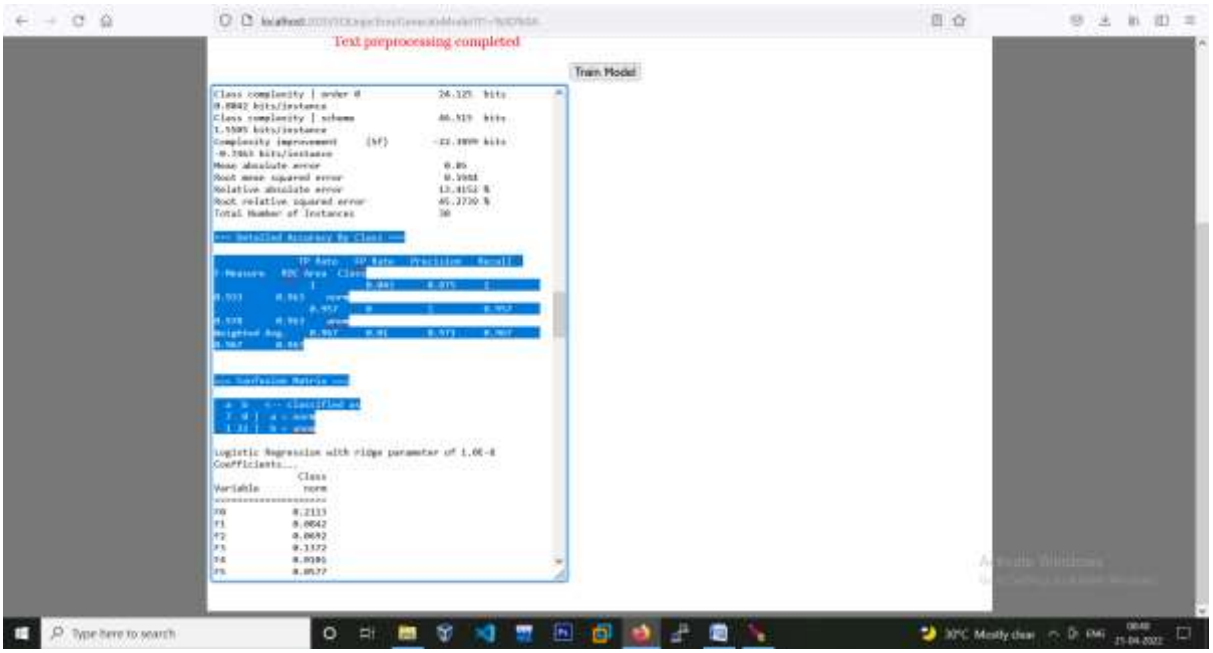

Fig 4:In above screen ROC and Confusion Matrix calculated where ROC score we got as 96% and in confusion matrix both classes total count also calculated and now scroll down above screen to get Logistic Regression accuracy

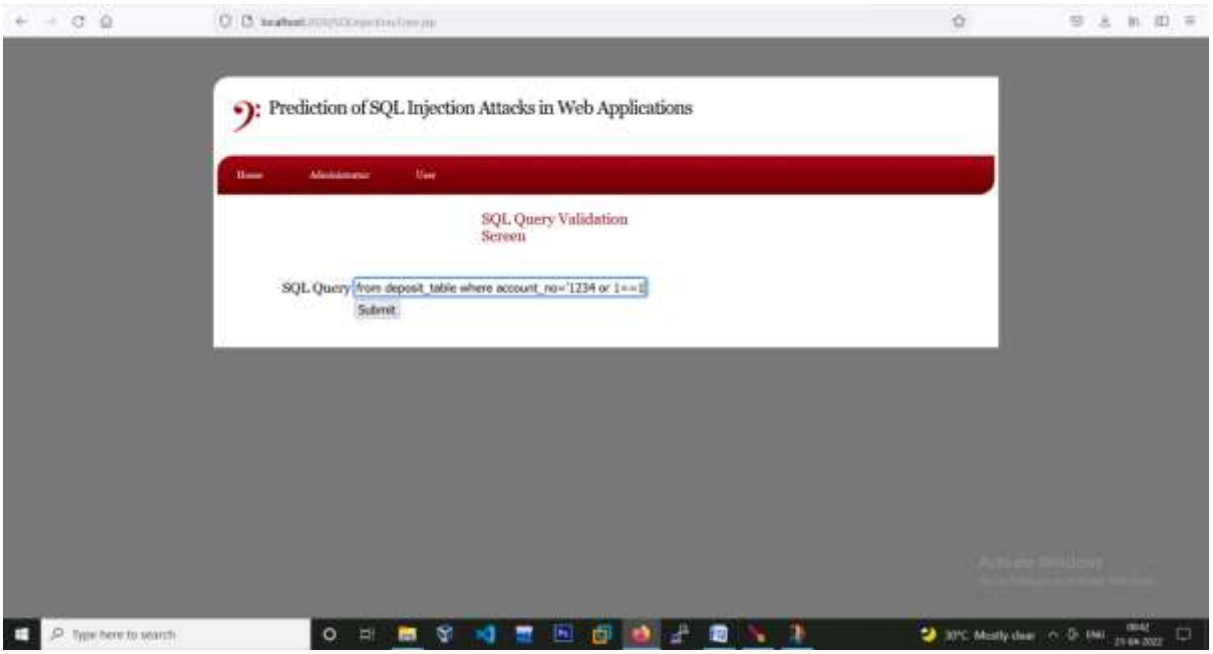

Fig 5:In above screen I gave some abnormal query which contains '1==1' operations and then upon executing that query with logistic regression model will get below classification result

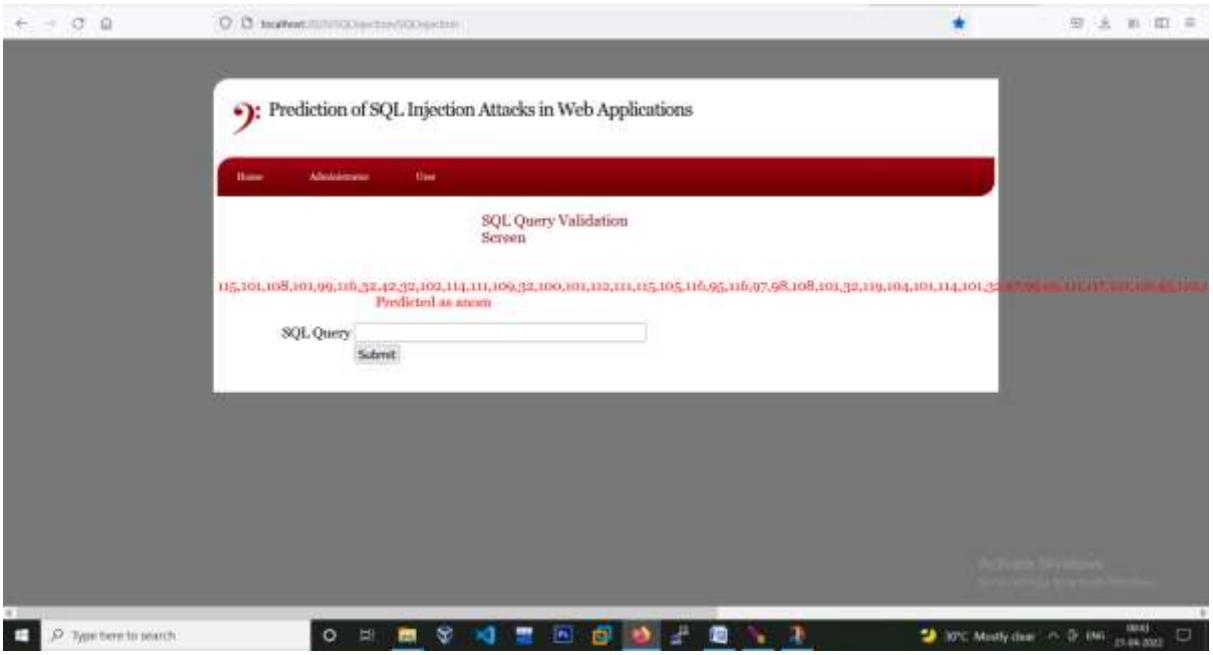

Fig 6:In above screen all numeric values are the features extracted from query and then we got result as 'anom' which is abnormal query and now test with normal query.

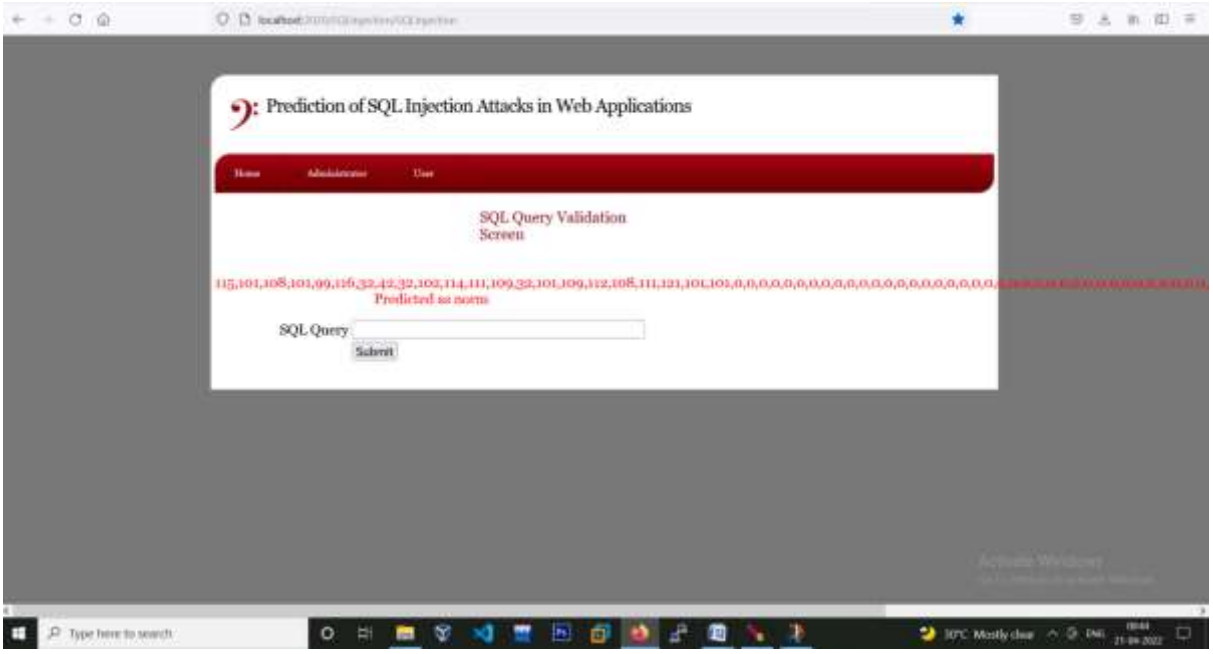

Fig 7:In above screen we got above query result as 'norm' which means query is normal. Similarly you can give query available in dataset then logistic regression model will give classification output as norm or anom.

In paper author has used inbuilt apache tool to build dataset and then this dataset input to logistic regression algorithm to get classification result. In propose paper author is using label as 1 or 0 and we are using label as norm or anom. Here we don't have any Internet service provider options so we get only localhost as IP

address. In propose paper author is using SQLinjection only for login queries but our dataset built on various queries.

Note: used dataset available with this code to build machine learning logistic regression model

5.CONCLUSIONANDFUTURESCOPE

Predicting vulnerabilities is a hard job, which is addressed in this paper. The internet software will be written and deployed on the tomcat server. The admin uploads a customized dataset containing a number of SQL queries. For prediction, this dataset will be used, and SQL question statements will be employed. On the records points, logistic regression will be utilized to make this prediction. 10 fold Cross Validation trying out is used for the coaching of dataset. The mannequin predicts SQL injection with excessive accuracy. When a SQL Injection Attack happens, a module can be developed in the future to increase an alarm. Also, based totally on the Source IP, one may additionally pinpoint the bodily vicinity from whence the assault originates. During the procedure, the proper consumer would possibly be recognised. For increased outcomes, deep mastering algorithms like Tree regressors can be used.

REFERENCES

[1]. Arumugam, C. *et al.* (2019). Prediction of SQL Injection Attacks in Web Applications.In: *,et al.* Computational Science and Its Applications – ICCSA 2019. ICCSA 2019.LectureNotesinComputerScience(),vol11622.Springer,Cham.

[2]. Scholte, T., Robertson, W., Balzarotti, D., Kirda., E.: An empirical analysis of inputvalidation mechanisms in web applications and languages. In: 27th Annual ACM SymposiumonAppliedComputing,pp.1419–1426(2012)

[3]. Alkhalaf, M., Aydin, A., Bultan, T.: Semantic differential repair for input validation andsanitization. In: ACM International Symposium on Software Testing and Analysis, pp. 225–236(2014)

[4].Frajták,K.,Bureš,M.,Jelínek,I.:Reducinguserinputvalidationcodeinwebapplicatio nsusingPex extension.In:ACM15thInternationalConferenceon ComputerSystemsandTechnologies,pp.302–308(2014)

[5]. Li, X., Xue, Y.: A survey on server-side approaches to securing web applications. ACMComput.Surv.(CSUR)46(4),54:1–54:29(2014)

[6]. Cho, S., Choi, J., Kim, G., Park, M., Cho, S., Han, S.: Runtime input validation for Javaweb applications using static bytecode instrumentation. In: ACM International Conference onResearchinAdaptiveandConvergentSystems,pp.148–152(2016)

[7]. Shar, L.K., Tan, H.B.K.: Mining input sanitization patterns for predicting SQL injectionandcrosssitescriptingvulnerabilities.In:34thInternationalConferenceonSoftwareEngineering,pp.1293–1296(2012)

[8]. Solomon, O.U., William, J.B., Lu, F.: Applied machine learning predictive analytics toSQLinjectionattackdetectionandprevention.In:IFIP/IEEEIM2017Workshop:3rdInternational Workshop on Security for Emerging Distributed Network Technologies, pp.1087–1090(2017