Prevention Of Sql Injection Attack Using Unsupervised Machine Learning Approach

M.N.Kavitha[1], V. Vennila[2], G.Padmapriya[3], A. Rajiv Kannan[4]

[1]assistant Professor, Department Of Computer Technology (Ug), Kongu Engineering College, Erode-638060, Tamilnadu, India,

[2] Associate Professor, Department Of Computer Science & Engineering, K.S.R. College Of Engineering, Tiruchengode- 637215, Tamilnadu, India,

[3] Associate Professor, Department Of Computer Science & Engineering, Saveethaschool Of Engineering, Saveetha Institute Of Medical And Technical Sciences, Chennai,Tamilnadu, India,

[4]professor& Head, Department Of Computer Science & Engineering, K.S.R. College Of Engineering, Tiruchengode - 637215, Tamilnadu, India,

Email: Kavithafeb1@Gmail.Com, Vennilview@Gmail.Com, Gpadmapriyame@Gmail.Com,Rajiv5757@Yahoo.Co.In

Abstract: Now A Day's Online Web Applications Or Online Database Applications Are Increasingly Exposed To Various Kinds Of Attacks. One Such Attack To Steal Data Is Called Sql Injection Attacks In Which Attackers Modify The Sql Query Initiated By The User And Adds Malicious Code To Access And Manipulate The Information In The Web Application Or Database. One Way To Prevent Such Attacks Is To Update And Test Web Application Firewall (Waf) Regularly. Due To Tremendous Growth In Technology, Attackers Who Intend To Attack The Applications Find Numerous New Ways To Enter Into The System. In This Paper, We Incorporate The Concept Of Machine Learning With Waf That Maximizes The Effectiveness Of Existing Systems. The Approach Adopted In This Paper Is Unsupervised Machine Learning Technique Which Uses K-Means Clustering Algorithm. The Flow Of The Proposed System Can Be Given As: The End User Makes A Query In The Web Application, And The Values Of Query Are Extracted And Sent To The Sql Injection Detector, Which Provides Two Layers Of Security. In The First Layer Of Security, Patterns Are Created Using Context-Free Grammar (Cfg) For Low Level Attacks. The Second Layer Of Security For High Level Attacks Is Trained Using Unsupervised Learning Algorithm.

Keywords: Machine Learning, Unsupervised Learning, Sql Injection, Waf, Cfg

1. INTRODUCTION

A Web Application Firewall (Waf) Secures Web Applications Or Online Database Applications From A Range Of Application Layer Attacks Such As Cross-Site Scripting (Xss), Sql Injection, And Cookie Poisoning, Among Others. A Http Application Uses A Web Application Firewall (Waf) As An Application Firewall. During Http Conversation It Applies A Collection Of Rules. Generally, These Rules Allow Common Attacks Like Cross-Site

Scripting (Xss) And Sql Injection. While Proxies Typically Shield Clients, Wafs Shield Servers. A Waf Is Deployed To Shield A Selected Internet Application Or Set Of Internet Applications. A Waf May Be Thought Of As A Reverse Proxy And Also Secures A Web Application That Involves Major Protection Necessities. In The Overall System Design, A Waf Is Placed In Front Of The Web Application That Needs To Be Secured. Each Query That Is Forwarded To The Web Application Is Inspected By The Waf Before It Approaches The Web Application. The Waf Delivers The Query To The Web Application Providing That The Query Adheres To The Firewall Rule Set. Since The Threat Of Cyber-Attacks Is Growing Day By Day, The Wafs Are Getting Complicated. Also, Manually Testing And Maintaining The Principles Is An Issue. Therefore, Automatic Testing Approaches For Wafs Are Vital To Prevent Malicious Queries Approaching Web Applications And Services. Sqli Attacks Have Never Lost Its Trend And Always Possess A Major Threat To The Web Applications Of Various Domains.

Waf Provides A Two Layer Security Component To Detect And Prevent Sql Injection Attacks. Sql Injection (Sqli) Could Be A Sort Of Associate Injection Attack That Causes It Likely To Execute Malicious Sql Statements. These Statements Monitor A Database Server At The Back Of Web Application. Sql Injection May Also Be Used To Manipulate Tuples Within The Database. The Objectives Of The Proposed Methodology Are Described As Follows: 1)To Extract The Values Passed In The Query Through Url, 2) Use Context Free Grammar To Create The Attack Patterns And Identify Attack And 3)Use Machine Learning Algorithm And Prevent Attack.

The Scope Of The Proposed Methodology Is To Make Use Of The Considerable Growth In Mechanisms For Tracking Various Types Of Web Application. It Adapts Machine Learning Techniques To Sense And Prevent Sql Injection Attack. This Paper Effectively Identifies Sqli Attacks Through Optimal Adaptation Of K-Means Clustering.

The Rest Of The Paper Is Organized As Follows. In Section 2, The Literature Survey Elaborates On The Research Works On Existing Systems. Section 3 Describes The System Design. Section 4 Gives The Details Regarding The System

Implementation. Section 5 Describes The Different Test Case Scenarios For The Modules Described By The Proposed System And The Performance Analysis Of The Proposed System. Section 6 Provides The Conclusion, Which Summarizes The Efforts Undertaken In The Proposed System, And States Findings And Shortcomings In The Proposed System.

2. RELATED WORKS

Many Research Works Are Carried Out In Firewalls Mainly For Detecting Sql Injection Since It Becomes A Major Security Threat. Some Of The Work Includes The Use Of Certain Algorithms To Test And Detect Attacks On Firewall, And Detection And Prevention Of Sqli Attacks. A Search-Based Approach Used To Integrates The Machine Learning And Eas To Automatically Check The Attack Identification Potentials Of Wafs. The Approach Automatically Produces And Checks Various Groups Of Attacks In Wafs, And Analyzes It If They Are Properly Detected. By Gradually Learning From The Test Result Which Query Is Blocked Or Bypassing The Firewall, The Approach Chooses A Test That Demonstrate String Patterns With Maximum Bypassing Possibilities (According To Machine Learning) And Alter Them Using An Attack Rules Intended To Produce New And Hopefully Successful Attacks. Detected Bypassing Attacks Are Often Used To Study The Path Conditions, Which Illustrate Successful Attack Patterns. With Such A Group Of Bypassing Attacks And Path

Conditions That Illustrate Them, A Security Expert Will Modify The Waf Rules In Order To Block Forthcoming Sqli Attacks. In The Attacker–Defender War, Time Is Vital. Being Able To Rapidly Study And Look Forward To Further Attack That Can Get Around A Firewall In A Suitable Method, It Is Extremely Significant To Protect Business Data And Services.

A Range Of Approaches Are Existing To Examine Sqli Attacks Including White-Box Testing[7], Static Analysis[14], Model-Based Testing[7] And Black-Box Testing[1]. However, Such Approaches Provide A Number Of Restrictions That May Negatively Affect Their Practical Applicability As Well As Their Exposure Identification Ability. As An Example, White-Box Testing Approaches And Static Analysis Tools Require Access To Source Code[6], Which Might Not Be Probable To Copy The Third-Party Elements Or Industrial Appliances, And Are Connected To Specific Programming Languages[26]. Model-Based Testing Techniques Need Models To Show The Security Policies Or The Implementation Of Wafs And The Web Application Under Test[7], That Are Normally Not Offered Or Terribly Difficult To Physically Build. In Black-Box Testing, It Don't Require Models Or The Source Code However They Are Not As Much Of Effective In Sleuthing Sqli Vulnerabilities. Certainly, Inclusive Reviews On Black-Box Techniques[1],[6] Have Revealed That Several Sorts Of Protection Vulnerabilities (Including Sqli Attacks) Still Exists Unidentified And Thus Necessitate Further Examination. Static Analysis And Runtime Monitoring[12] For Analyzing The Query That Is To Be Executed On The Server Side Which Helped In Prevention Of Static And Dynamic Sql Injection Attacks. A Learning Approach[9] Produces Attacks From A Rules And Studies Which Model An Attack Cannot Include In Order To Escape From Identification. A Framework[12] Automatically Checks If An Approach Is Properly Imposed By A Firewall. As A Result, The Framework Produces A List Of Approaches, Tests Traffic And Checks How Well The Firewall Manages The Produced Traffic With Respect To The Given Approach. Structural Coverage Conditions Of Approaches[8] [10]In Which Test Are Defined And Implements A Test Production Approach Based On Impulsion Solving That Aims To Take Advantage Of Structural Coverage. Analysis For Monitoring And Neutralizing Sql Injection Attacks(Amnesia)[5] Does The Required String Analysis And We Don’t Need Any Source Code Adjustments Here. One Of The Major Issues In This Approach Is That Automatic Detections Are Not Available. Detection Rates Are Less Than 75% Which Makes This Approach Unreliable.

Akanksha Kapoor [3] Describes The K-Means Algorithm To Be Tremendously Influential In Building Clusters For Several Systematic Applications In Emerging Domains. The Approach Is Handled With Random Update To K As The Initial Center Point From The Dataset[2]. It Tends To Compute The Euclidean Distance Of Every Data Point From The Initial Cluster Centers. Then It Chooses The Model Which Is Mainly Neighboring And Then Allocates It To The Felicitous Cluster. [13] The Center Is Updated Until The Mean Square Error Results In Least Or The Cluster Center Remains Same For Subsequent Iterations. During This Position, All The Data Points Have Least Distance From The Center Point. In Addition, [8] Fuzzy C-Means (Fcm) Sanctions A Data Point Resides In All The Clusters With The Association In Between 0 And 1. If The Data Point Is Highly Closer To The Cluster Center, Its Association Towards That Cluster Is Highly Desirable And Hence Produces More Effective And Tighter Clusters[16]. The Major Drawback Of Using K-Means Algorithm Lies In Its Difficulty To Determine Right Value Of K For Productive Clustering.

Currently There Is No Automatic Identification And Prevention Of Sqli Attacks. The Existing System Makes The Detection Based On A Certain Set Of Rules[11]. The System Checks The Query With Each And Every Rule And Then It Detects For Attack. If A Certain

Type Of Rule Is Not Maintained In The Set And If That Attack Is Invoked, And Then The System Will Allow The Query To Be Executed Since That Rule Is Not Present In The Rule Set. It Is Also Difficult To Maintain The Set Of Rules And Test Them. Also We Need To Define More Complex Rules For Huge Applications.

3. METHODOLOGY

Sql Injections Using Machine Learning Serves As A Prevention Mechanism To The Utilization Of Any Database In The Server Side. In This Methodology, A Segment Is Built Up That Lives In The Server Side[12]. The System Checks For The Sql Injection Attack Patterns That Are Affixed With The Values Passed To The Application Server Prior To The Processing Of The Query Made By The Client[15]. Two Levels Of Security Are Defined In This System: In The First Level The Patterns Produced By The Cfg Rules And Relating These Values With The Pattern Produced By The Rules That Are Set For Sql Attacks[16]. If There Is No Match Of Values In The First Level It Is Moved To The Next Level Of Security That Utilizes The K-Means Clustering Algorithm That Clusters The Patterns In The Dataset And Classifies Them Into One Of Those Clusters And Avoids The Injection.

Fig.1 Shows The System Architecture[17]. The Client Uses The Web Browser To Send The Queries To The Web Server. The Web Browser Propels The Values Given By The User To The Web Server. The Sql Injection Detector In The Web Server Tests If There Is Any Sqli Attacks From The Values Passed By The User. If The Values Are Found To Be Valid Then It Is Directed To The Query Processor That Executes The Query To Process[18]. To End With, The Query Of The Client Is Processed And Executed By Accessing The Database.

Fig.2 Depicts The Functional Architecture Of The System That Describes About The Complete Working Of The System With All The Necessary Details. It Also Points Out The Algorithms Used In This Method Such As Context Free Grammar And K- Means Clustering. The User Enters The Data In The Form And The Getrequest Directs To The Url Intercept Engine That Extracts The Values From The Query. The Injection Detector Checks The Values With The Pre-Defined Patterns Using Context Free Grammar And Then Categorizes Attacks Using K-Means Clustering. If No Attacks Are Found The User Can Access The Database Using The Values Passed And The Response Is Forwarded To The Client. Otherwise, The Access Is Not Permitted And The Type Of Attack Matching The Pattern Is Found.

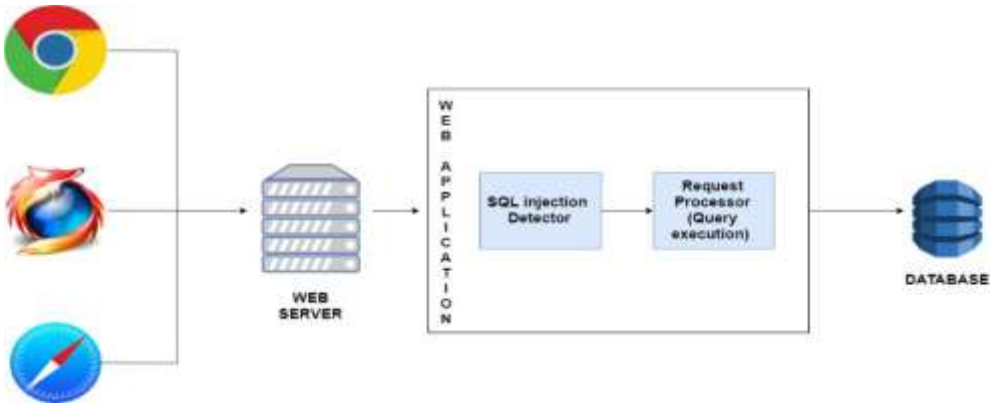


Fig. 1. System Architecture

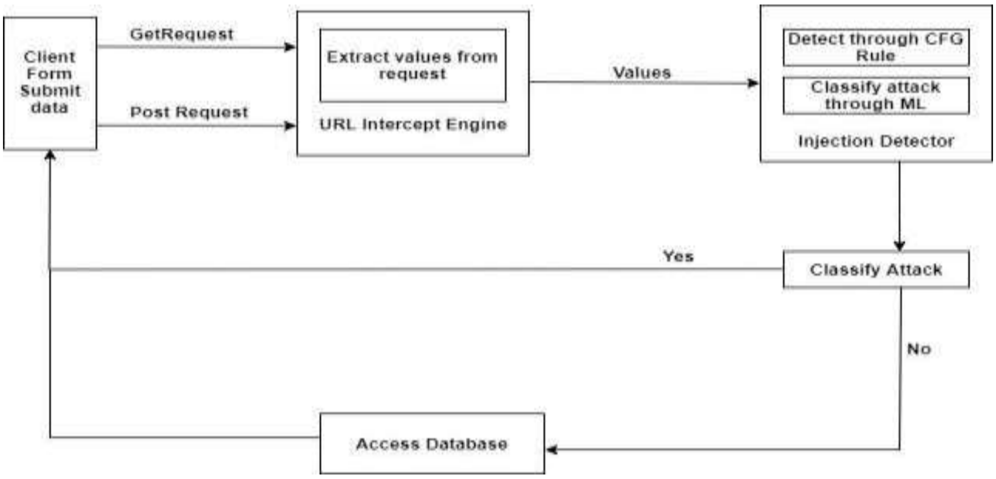
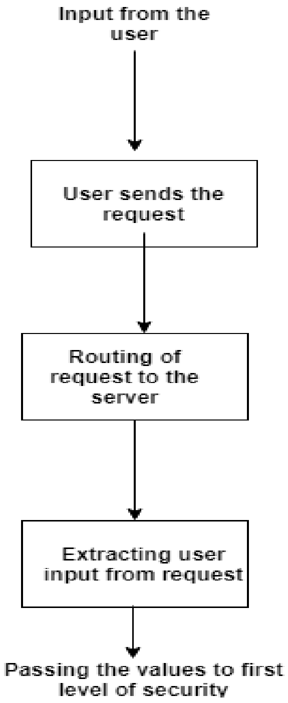


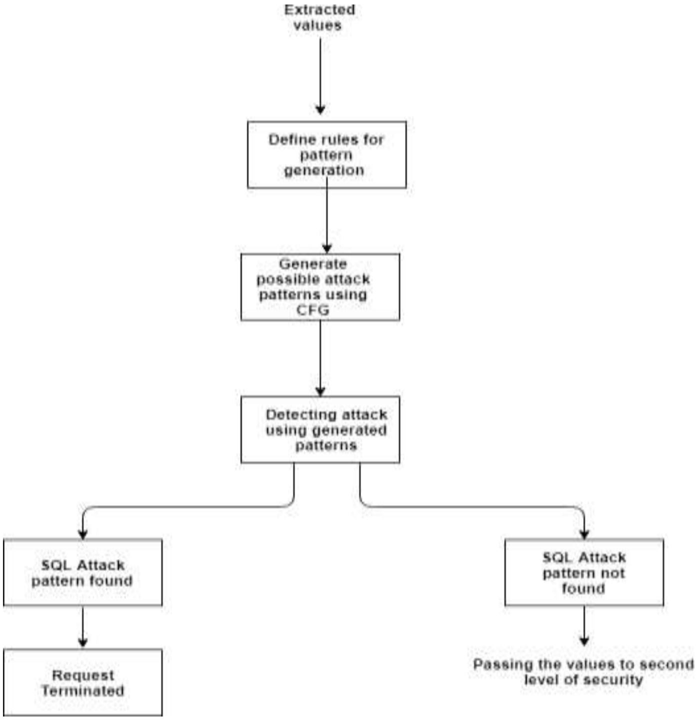
Fig. 2. Functional Architecture

The Proposed Methodology Consists Of The Following Three Main Steps Namely, (1) Url Intercept Engine (2) Context- Free Grammar For Sqli Attacks And (3) Pattern Classification Through Machine Learning.

a. Url Intercept Engine

The Most Vulnerable Part Are The Text Fields In The Web Applications Where Most Of The Sql Injection Attacks Happen. Url(Uniform Resource Locator) Is Denotes The Global Address Of Resources On The World Wide Web. The Url Directs The Users To A Specific Resource Online, Such As A Web Page, Video Or Other Document Or Resource. The Attackers Having An Intention To Affect The Database Of Any Organization, May Enter Some Harmful Sql Queries In The Text Field Which Can Be Affixed To The Already


Input from the user
User sends the request
Routing of request to the server
Extracting user input from request
Passing the values to first level of security


Extracted values
Define rules for pattern generation
Generate possible attack patterns using CFG
Detecting attack using generated patterns
SQL Attack pattern found
Request Terminated
SQL Attack pattern not found
Passing the values to second level of security

i. **(B)**

Fig. 3. Working Of A) Url Intercept Engine And B) Cfg For Sqli Attacks

Defined Sql Query In The Server Side Affecting The Database. Hence The Initial Segment Concentrates On The Values Entered In The Text Fields And Directs These Values To The Pattern Checking Algorithm. Fig. 3a Depicts How The Values Extracted Are Received By The Server.

b. Context Free Grammar For Sqli Attacks

The Next Step Provides The Initial Level Of Security From The Sql Injection Attacks. The Sql Injection Attack Pattern Are Produced By The Protocols Of Context Free Grammar. The Extracted Value From The Initial Segment Is Checked With The Protocol Creating Different Attack Patterns. If This Value Matches With The Attack Pattern Produced By The Cfg, It Is Directed To The Next Level Security To Determine Whether The Value Entered Is Harmful Or Not. Although The Value Entered Does Not Match The Pattern Created By The Cfg, It Is Passed To The Second Level Security. The Cfg Rules Thus Create Patterns Until There Are No Terminals Left. Fig. 3b Illustrates The Process Of Checking The Value Against The Generated Patterns.

c. Pattern Classification Through Machine Learning

This Step In The Proposed Methodology Contributes The Second Level Of Security To The System. Fig. 4 Illustrates The Workflow Of Pattern Classification Through Machine Learning. An Unsupervised Machine Learning Algorithm Groups The Different Types Of Attacks Into Individual Clusters Using The Fitness Value Associated With The Cluster. If The Value

Matches With The Cluster Associated With Any Of The Identified Attack Pattern, The System Thwarts The Query From Executing. Otherwise, The Query Is Executed By Accessing The Database.

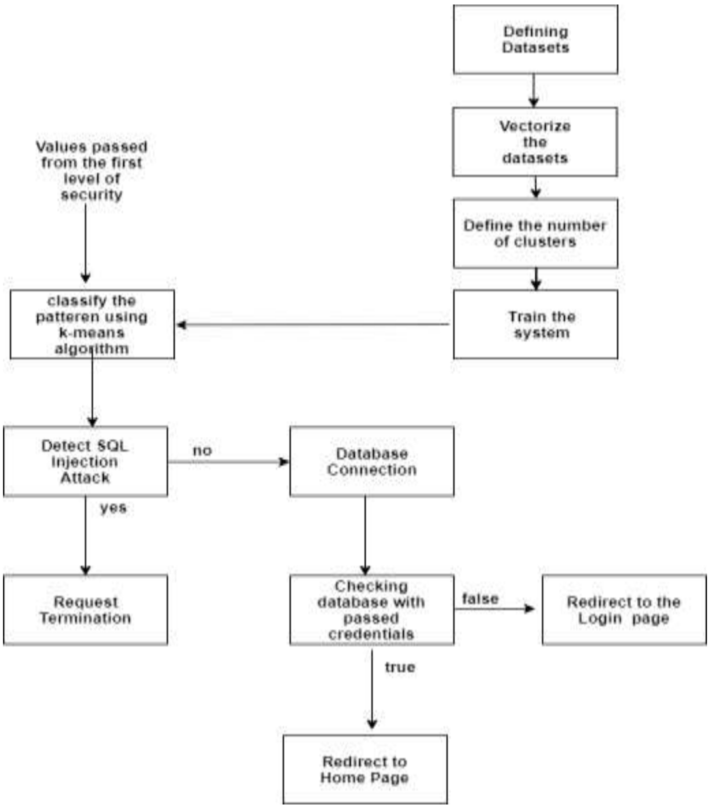
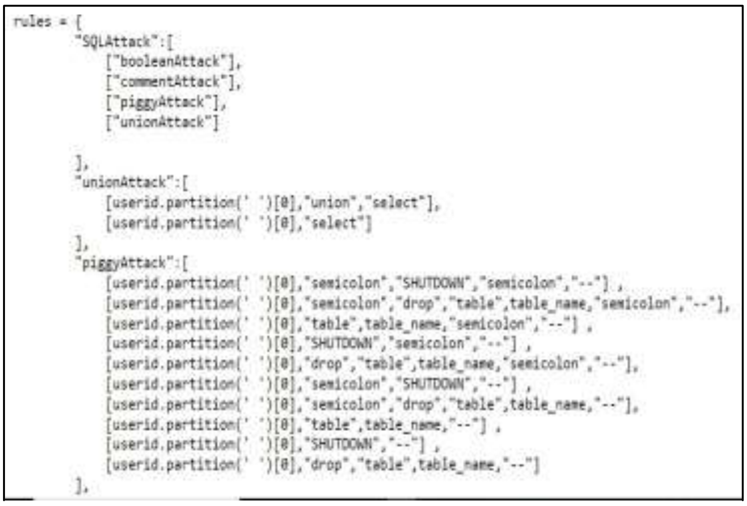

Defining Datasets
Vectorize the datasets
Define the number of clusters
Train the system
Values passed from the first level of security
classify the patteren using k-means algorithm
Detect SQL Injection Attack
no
Database Connection
yes
Request Termination
Checking database with passed credentials
false
Redirect to the Login page
true
Redirect to Home Page

Fig. 4. Working Of Pattern Classification Through Machine Learning

2. System Implementation

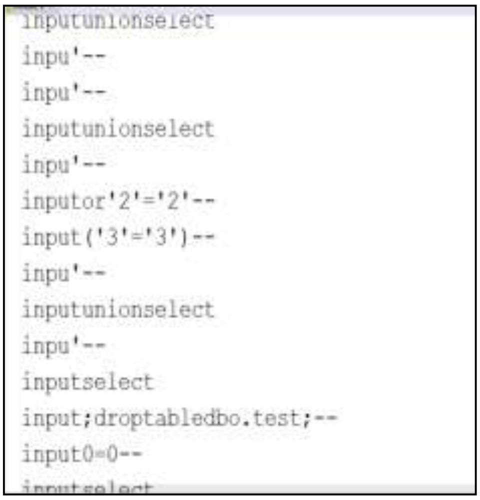
a. Implementation Of Pattern Creation



```
rules = [
        "SQLAttack":[
            ["booleanAttack"],
            ["commentAttack"],
            ["piggyAttack"],
            ["unionAttack"]

        ],
        "unionAttack":[
            [userid.partition(' ')[0],"union","select"],
            [userid.partition(' ')[0],"select"]
        ],
        "piggyAttack":[
            [userid.partition(' ')[0],"semicolon","SHUTDOWN","semicolon","--"] ,
            [userid.partition(' ')[0],"semicolon","drop","table",table_name,"semicolon","--"],
            [userid.partition(' ')[0],"table",table_name,"semicolon","--"] ,
            [userid.partition(' ')[0],"SHUTDOWN","semicolon","--"] ,
            [userid.partition(' ')[0],"drop","table",table_name,"semicolon","--"],
            [userid.partition(' ')[0],"semicolon","SHUTDOWN","--"] ,
            [userid.partition(' ')[0],"semicolon","drop","table",table_name,"--"],
            [userid.partition(' ')[0],"table",table_name,"--"] ,
            [userid.partition(' ')[0],"SHUTDOWN","--"] ,
            [userid.partition(' ')[0],"drop","table",table_name,"--"]
        ],
```

Fig. 5. Rules For Pattern Generation



```
inputunionselect
inpu'--
inpu'--
inputunionselect
inpu'--
inputor'2'='2'--
input('3'='3')--
inpu'--
inputunionselect
inpu'--
inputselect
input;droptabledbo.test;--
input0=0--
inputselect
```

Fig. 6. Possible Attack Patterns

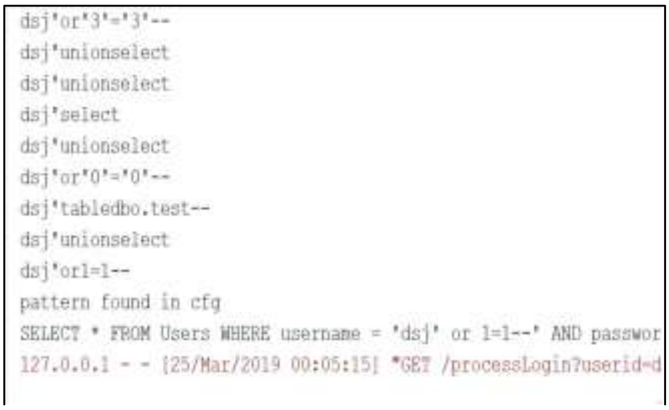
Fig. 5 Shows The Rules Specified For The Random Creation Of The Sql Injection Attack Pattern That Can Be Admitted Together With The User Input. These Rules Are Being Used By The Cfg Function That Creates The Pattern. The Rules Are Specified To Create 3 Categories Of Sql Injection Attacks Which Are Known As, Boolean Attack, Piggy Backed Attack And Union Attack. After Extracting User Input The Query Is Then Directed To The Rules Where The Possible Attack Patterns Are Created By The Function That Can Be Likely Attached At The End Of The User Input.

Fig. 6 Shows The Output Of The Possible Attack Pattern That Can Be Attached At The End Of The User Input, Created By The Rules Specified As Shown In Fig. 7. The User Input And The Attack Identified Are Compared Against With These Patterns.

b. Identification And Prevention

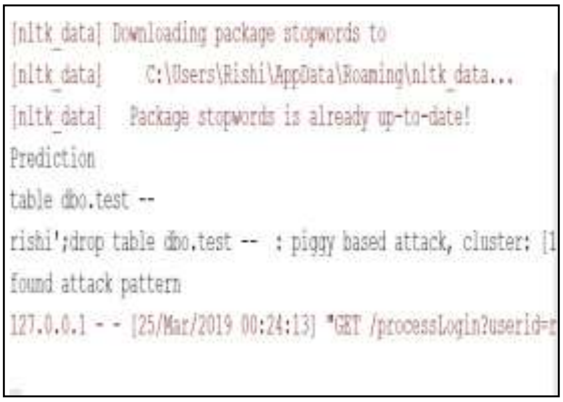
Fig.7 Denotes The Identification Of The Sql Injection Created By The Cfg Patterns And Also Creates The Sql Query Which Is Executed By The Sql Server. Fig.7 Denotes All Categories Of Attacks That Can Be Attached At The End With The Username Are Created And At Last It Matches With One Of The Attack Patterns Created By The Cfg Function.

Fig.8 Denotes The Identification Of Attack Pattern By The System Itself, As The System Is Learned By A Unsupervised Learning Technique Using K-Means Clustering Algorithm. Using Nltk The Stop Words Such As And, Or, From, To, Etc Are Identified And Removed And Only The Significant Words Are Utilized For Teaching The Machine. Based On The Cluster, It Identifies The Attack And Prevents The Query To Be Executed.



```
dsj'or'3'='3'--
dsj'unionselect
dsj'unionselect
dsj'select
dsj'unionselect
dsj'or'0'='0'--
dsj'tabledbo.test--
dsj'unionselect
dsj'or1=1--
pattern found in cfg
SELECT * FROM Users WHERE username = 'dsj' or 1=1--' AND passwor
127.0.0.1 - - [25/Mar/2019 00:05:15] "GET /processLogin?userid=d
```

Fig. 7. Identification By The Cfg Function



```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Rishi\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Prediction
table dbo.test --
rishi';drop table dbo.test --  : piggy based attack, cluster: [1
found attack pattern
127.0.0.1 - - [25/Mar/2019 00:24:13] "GET /processLogin?userid=r
```

Fig. 8. Identification By Clustering

4. RESULTS AND DISCUSSION

c. Test Cases

The User Enters The Data And If The Username And Password Is Valid, The Web Application Proceeds To The Next Page. Table 1 Shows The Test Case For Login With And Without Sql Injection Detection. If The Login Credentials Are Incorrect The User Will Be Shown The Login Page Again. Also If The Login Credentials Contains Malicious Patterns Along With Them, It Is Detected And Prevented And Returns The Login Page.

d. Performance Measure

Accuracy. The Accuracy Of A Test Is Its Ability To Differentiate The Types Of Attacks

Correctly. It Is Calculated As Shown In Equation (1),

Accuracy = Tp+Tn/Tp+Fp+Tn+Fn (1)

Where True Positive (Tp) Is The Number Of Attack Patterns Correctly Identified, False Positive (Fp) Is The Number Of Attack Patterns Incorrectly Identified, True Negative (Tn) Is The Number Of Safe Patterns Correctly Identified And False Negative (Fn) Is The Number Of Safe Patterns Incorrectly Identified.

Response Time. Response Time Denotes The Length Of The Time Taken By A Person Or A System To Respond To A Given Stimulus Or Event As Given In Equation (2),

Response Time = Reporting Period / Check Frequency Interval (2)

Time Complexity. The Time Requirement For K-Means Is Given In Equation (3),

O (I*K*M*N) (3)

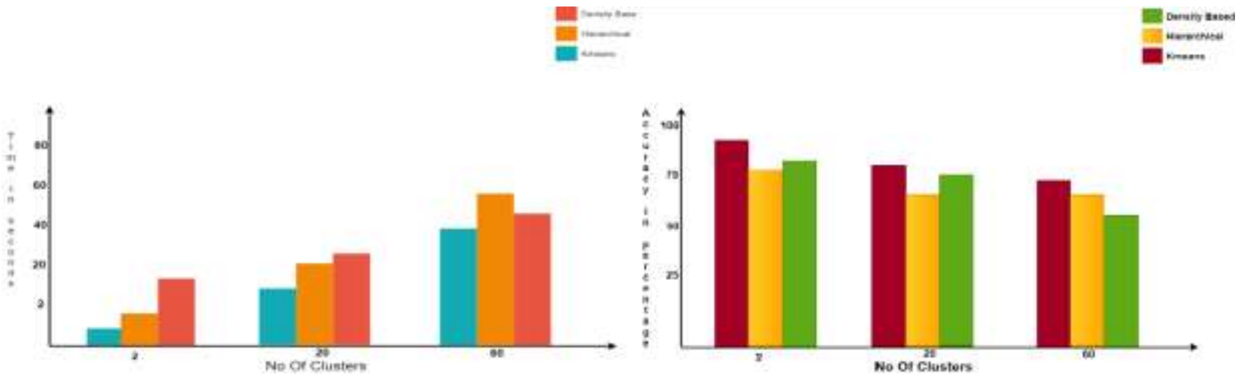
Where I Is The Number Of Iterations Required, N Is The Number Of Attributes, M Is The Number Of Points, K Is The Number Of Clusters And I Is Often Small And Can Usually Be Safely Constrained, As Most Of The Changes Occur In The Initial Few Iterations.

Table 1. Login With And Without Sql Injection Detection

Test Case Id	**Test Description**	**Test Input**	**Expected Result – Without Sql Injection**	**Expected Result – With Sql Injection**	**Actual Result - Without Sql Injection**	**Actual Result– With Sql Injection**
1.	Check Login Using User Details With Attack	Boolean Attack Pattern	Should Open The Home Page With Wrong Credentials	Attack Should Be Detected And Return The Login Page	Home Page Opened	Attack Detected And Login Page Returned
2.	Check Login Using User Details With Attack	Piggyback Attack	Should Open The Home Page With Wrong Credentials And Database Should Be Affected	Attack Should Be Detected And Protect The Database And Also Return The Login Page	Home Page Opened And Database Affected	Attack Detected, Login Page Returned And Database Protected

e. Performance Analysis

The Performance Of The Proposed System Can Be Measured By Accuracy Of The Algorithm And Time Taken To Cluster The Datasets.



i. (B)

Fig. 9. Performance Analysis – (A) Time Taken To Cluster And (B)Accuracy

Time Taken To Cluster. The Performance Of The Proposed System Based On The Time Taken To Cluster The Datasets Of Different Clustering Algorithms Are Depicted As Shown In The Fig. 9(A). The Bar Graph Has Been Created Based On The Time Taken By The Three Different Clustering Algorithms. The Vertical Axis Consists Of The Time Taken To Cluster The Data In Seconds And The Horizontal Axis Consists Of The Number Of Clusters.

Accuracy. The Performance Of The Proposed System Based On Accuracy Of Various Algorithms Can Be Depicted As Shown In Fig.9(B). The Bar Graph Has Been Created Based On The Accuracy From Three Different Algorithms. The Vertical Axis Consists Of The Accuracy Percentage And The Horizontal Axis Consists Of A Number Of Clusters.
Thus The Performance Analysis Of A System Which Proved To Improve The Level Of User Satisfaction Compared To The Existing System.

5. CONCLUSION AND FUTURE WORK

The Proposed Methodology Uses A Machine Learning Approach To Detect And Prevent Sql Injection Attacks Through User Queries In A Web Application. There Are Three Modules In The System Namely, Url Intercept Engine, Cfg For Sqli Attacks And Classify Pattern Through Machine Learning. At First, The User Enters The Values On The Client Side Which Is Extracted Using Url Intercept Engine. It Is Passed To The First Level Of Security That Detects The Pattern Using The Context-Free Grammar Rules. The Patterns Are Passed To The Machine Learning Algorithm Which Classifies The Value To Clusters Based On The Pattern. If The Value Is Found To Be Malicious By The Technique, Then The Query Made By The Client Will Be Cancelled. Else, The Query Will Be Executed Normally. Future Enhancements Of This System Can Include The Prevention Of Cross-Site Scripting (Xss) And Path Traversal Attacks In Web Applications Using Machine Learning Techniques. Xss Permits Attackers To Inject Client-Side Scripts Into Web Pages Viewed By Other Users. This Vulnerability May Be Used By Attackers To Bypass Access Controls Like The Same-Origin Policy. A Directory Traversal Or Path Traversal Attack Targets To Access Files And Directories Stored Outside The Web Root Folder. This Attack Is Also Known As “Dot-Dot-Slash”, “Directory Climbing” Or “Backtracking”. It May Be Possible To Access Arbitrary Files And Directories Stored On The File System Including Application Source Code Or

Configuration And Critical System Files By Changing The Variables That Reference Files With “Dot-Dot-Slash (../)” Sequences And Its Variations Or By Using Absolute File Paths. The Access To Files Is Limited By System Operational Access Control (Such As In The Case Of Locked Or In-Use Files On The Microsoft Windows Operating System). Both These Attacks Are Major Threats To Web Applications. The Machine Learning Techniques Can Be Developed For The Detection And Prevention Of Cross-Site Scripting And Path Traversal Attacks Using The Efficient Use Of Technology.

6. REFERENCES

[1] A.Doupe, M.Cova, And G.Vigna (2010), “Why Johnny Cań AˆăZt Pentest : An Analysis Of Black-Box Web Vulnerability Scanners,” In Proc. Int. Conf. Detection Intrusions Malware, Vulnerability Assessment, Pp. 111–131.

[2] A.Gupta ; H. Shivhare ; S. Sharma (2015) “Recommender System Using Fuzzy C-Means Clustering And Genetic Algorithm Based Weighted Similarity Measure” In International Conference On Computer, Communication & Control,Pp.1-8

[3] Akanksha Kapoor, Abhishek Singhal (2017) “A Comparative Study Of K-Means, K-Means++ And Fuzzy C-Means Clustering Algorithms”,3rd Ieee International Conference On "Computational Intelligence And Communication Technology",Ieee-Cict,Pp.74–81

[4] E.Al-Shaer,A. El-Atawy, And T. Samak Apr. (2009), “Automated Pseudo-Live Testing Of Firewall Configuration Enforcement,”Ieee J. Sel. Areas Commun., Vol. 27, No. 3, Pp.302–314,.

[5] Inyong Lee, Soonkjeong, Sangsoo Yeo And Jongsub Moon (2012),‘A Novel Method For Sql Injection Attack Detection Based On Removing Sql Query Attribute Values’, Elsevier, Vol. 55, Pp.58 -68.

[6] J.Bau,E.Bursztein, D.Gupta,And J.Mitchell (2010), “Stateoftheart: Automated Black-Box Web Application Vulnerability Testing,” In Proc. Ieee Symp. Security Privacy, Pp.332–345.

[7] J. Jurjens And G. Wimmel(2001), “Specification-Based Testing Of Firewalls,” In Perspectives Of System Informatics (Lecture Notes In Computer Science, Vol. 2244), D. Bjørner, M. Broy, And A. Zamulin, Eds. Berlin, Germany: Springer ,Pp.308– 316.

[8] M.Yedla, S. R. Pathakota,T M Srinivasa,(2010) “Enhancing K-Means Clustering Algorithm With Improved Initial Center” In International Journal Of Computer Science And Information Technologies, Vol.1(2), Pp.121-125.

[9] O. Tripp, O. Weisman, And L. Guy (2013), “Finding Your Way In The Testing Jungle : A Learning Approach To Web Security Testing,” In Proc. Int. Symp. Softw. Testing Anal., Pp.347–357.

[10] W.G.J.Halfondand A.Orso (2006), “Preventing Sql Injection Attacks Using Amnesia,” In Proc. 28th Int. Conf. Softw. Eng. , Pp.795–798.

[11] W. Halfond, J. Viegas, And A. Orso (2006), “A Classification Of Sql-Injection Attacks And Countermeasures,” In Proc. Ieee Int. Symp. Secure Softw.Eng., Vol. 1, Pp.13–15.

[12] Witt Yi Win, And Hnin Hnin Htun, (2013), “A Simple And Efficient Framework For Detection Of Sql Injection Attack”, Ijccer, Vol.1, Pp. 26 -29

[13] Xiulisha, Huichao Lee, Bo Shen, Yiwei Liu, (2017),” Automatic K Selection Method For K-Means Algorithm”, International Conference On Systems And Informatics, China,Pp.1573-1578

[14] X. Fu, X. Lu, B. Peltsverger, S. Chen, K. Qian, And L. Tao (2007), "A Static Analysis Framework For Detecting Sql Injection Vulnerabilities," In Proc.31st Annu. Int. Comput. Softw. Appl. Conf., July, Vol.1, Pp. 87–96.

[15] Y.-F. Li, P. K. Das, And D. L. Dowe (2014), "Two Decades Of Web Application Testing : A Survey Of Recent Advances," Inf. Syst., Vol.43, Pp. 20–54.

[16] Z.Min,Kai-Fei(2015),"Improved Research To K-Means Initial Cluster Centers" In Ninth International Conference On Frontier Of Computer Science And Technology,Pp.349-353.

[17] Sujatha, K & Shalini Punithavathani, D 2016, 'Fuzzy Based Weight Estimation And Sub Band Architecture In Image Fusion For Multi Exposure Images', Asian Journal Of Information Technology (Ajit), Issn:1682-3915, Vol. 15, No.3, Pp.384-392.

[18] Basha, A.J., Balaji, B.S., Poornima, S. Et Al. Support Vector Machine And Simple Recurrent Network Based Automatic Sleep Stage Classification Of Fuzzy Kernel. J Ambient Intell Human Comput (2020)