Advanced Web Attacks Detection with Deep Learning

DR. SUBBA RAO KOLAVENNU[1], DR. P. S. V. SRINIVASA RAO[2], P. RUCHITHA[3], Y. AKSHARA[4], P. SATHWIK[5]

[1] *Computer Science & Engineering (CS) (JNTUH) Sphoorthy Engineering College.*

[2] *Computer Science & Engineering (CS) (Assistant Professor) Sphoorthy Engineering College (JNTUH)*

[3, 4] *Computer Science & Engineering (CS) (B. Tech, JNTUH) Sphoorthy Engineering College (JNTUH)*

Abstract*— Web applications are a popular target for cyber attacks because they are accessible over the network and are often vulnerable. Intrusion detection systems monitor web applications and provide alerts when an attack is detected. Current implementation of intrusion detection systems usually extract features from network packets or input string features and manually select features for analysis. However, manually selecting features is time-consuming and requires in-depth security knowledge. In addition, supervised learning algorithms need a lot of legitimate records and attack request data to identify bad and bad behavior; This is often expensive and impractical for developing web applications. This article contributes to research on controlling system access. First, we evaluate the feasibility of unsupervised/semi-supervised web attack detection based on the Robust Software Modeling Tool (RSMT), which works in a supervised and behavior-oriented manner on the website. Second, we describe how RSMT trains stacked denoising autoencoders to encode and reconstruct call graphs for end-to-end deep learning; where a low-dimensional representation of the raw data of unlabeled requested data is used to identify defects using defect data. Third, we analyze the results of experimental evaluation of RSMT on synthetic data and production practices with adverse effects. Our results show that the application can effectively and accurately detect attacks including SQL injection, cross-site scripting, and deserialization with minimal information.

I. INTRODUCTION

Modern web applications are like intricate puzzles, with components scattered across different systems and sometimes running on multiple devices. Ensuring secure communication between these parts, especially with powerful backend AI systems, is critical. This requires a clear understanding of how the application functions at both the business and technical levels for secure deployment. Our research offers a glimmer of hope. We've developed a system that effectively detects common attacks (SQL injection, cross-site scripting) with minimal data. Unfortunately, web applications are a prime target for attackers who aim to steal user data or disrupt services, leading to significant financial losses. Firewalls provide some defense, but vulnerabilities persist. Over half of applications released between 2015 and 2016 had weaknesses. Hacking attacks are a serious financial threat, costing US companies millions annually. The Equifax data breach, where a vulnerability exposed millions of consumers' data, is a stark example. Traditional security systems often struggle with these complex applications due to performance limitations or the need for specialized knowledge. This highlights the urgent need for new approaches to effectively secure the web applications of today**.**

II. LITERATURE SURVEY

Classification of sql-injection attacks and countermeasures.

III. EXISTING SYSTEM

Existing implementations of intrusion detection systems typically extract features from network packets or input string characteristics that are manually selected as relevant for attack analysis. However, manual feature selection is time-consuming and requires thorough knowledge of the security domain. In addition, large amounts of labeled legitimate and attack request data are required, requiring supervised learning algorithms to classify normal and abnormal behavior, which is often expensive and impractical to obtain for production web applications.

IV. DISADVANTAGES OF EXISTING SYSTEM

1.Lack of attack identification in web application.

2.Researchers found that more than half of web applications during the 2015-2016 review contained high security flaws

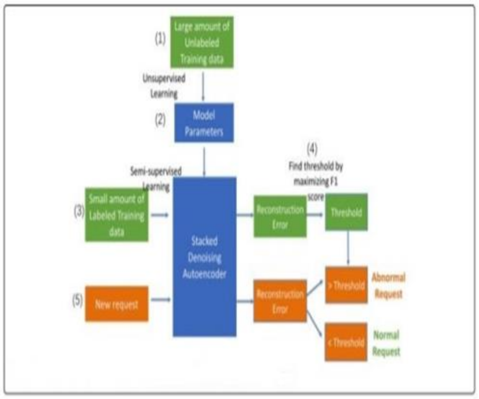
V. PROPOSED SYSTEM

In this paper, the author describes the concept of detecting attacks from a web application using Deep Learning Network and Robust Software Modeling Tool (RSMT). The RSMT tool is a web monitoring tool that monitors the startup behavior of a web application and records it in a trace file. The trace file contains low-dimensional raw data and cannot be used for Deep Learning Network. The author uses an autoencoder technique to convert this raw data into deep learning features. An autoencoder converts the raw data into deep learning features. These features will be passed to the design of the AutoEncoder algorithm, which will generate training and test data from the features. The AutoEncoder algorithm requires unlabeled train data to generate the model, and new test data will be applied to the AutoEncoder train model to identify new test data that is a normal request or contains an attack. If no new test data is available in the AutoEncoder train model, it will be considered an attack.

VI. ADVANTAGES OF PROPOSED SYSTEM

we use the LSTM (Long Short Term Memory) algorithm, which is an advanced version of a deep learning network whose prediction accuracy is greater compared to existing algorithms.

VII. SYSTEM ARCHITECTURE



VIII. FUNCTIONAL REQUIREMENTS

Functional requirements are represented or expressed in the form of input to be provided to the system, operation performed and output expected. The system should collect data from any sources. All collected data should be processed for proper use, some analysis should be done for proper understanding of the data.

1.Upload the RSMT Traces dataset
2. Data preprocessing
3. Model generation
4. Registration and login
5.View comparison chart

IX. NON-FUNCTIONAL REQUIREMENTS

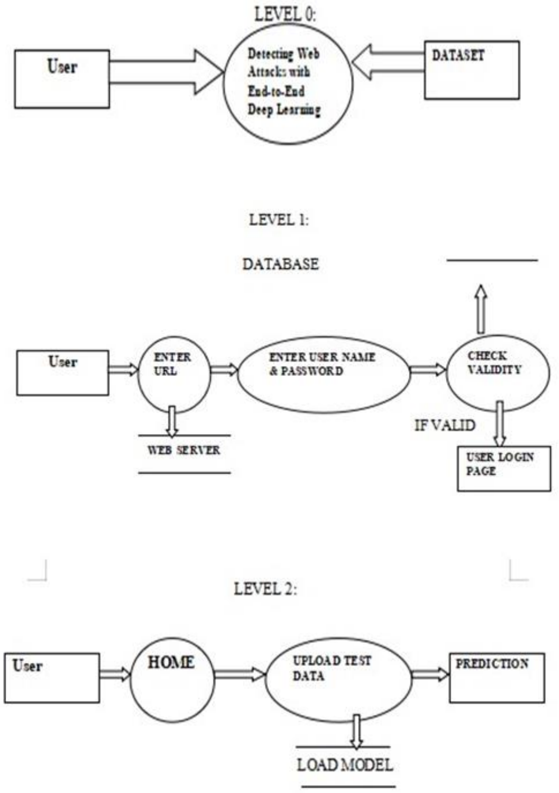
Applicability:
Usability is the main non-functional requirement for "Detection Web Attacks with End-to-End Deep Learning". The user interface should be simple enough that anyone can understand it and get the relevant information without special training. Different languages can be provided based on requirements.

Accuracy:
Accuracy is another important non-functional requirement for "Web Attack Detection with Complex Deep Learning". The dataset is used to train and test the model in python. The forecast should be correct, consistent and reliable.

Availability:
The system should be available while the user is working, and in the event of a failure, it must be restored within an hour or less. The system should respond to requests in two seconds or less.

Sustainability:
The software should be easy to maintain, and adding new features and making changes to the software should be as easy as possible. In addition, the software must also be portable..

X. DATA FLOW DIAGRAM



1. A DFD is also called a bubble map. It's a simple graphical formalism that can be used to represent a system in terms of input data to the system, colorful processing with that data, and affair data generated by the system.

2. Data inflow illustration(DFD) is one of the most important modeling tools. It's used to model system factors. These factors are the system process, the data used by the process, the external reality that interacts with the system, and the information flows in the system.

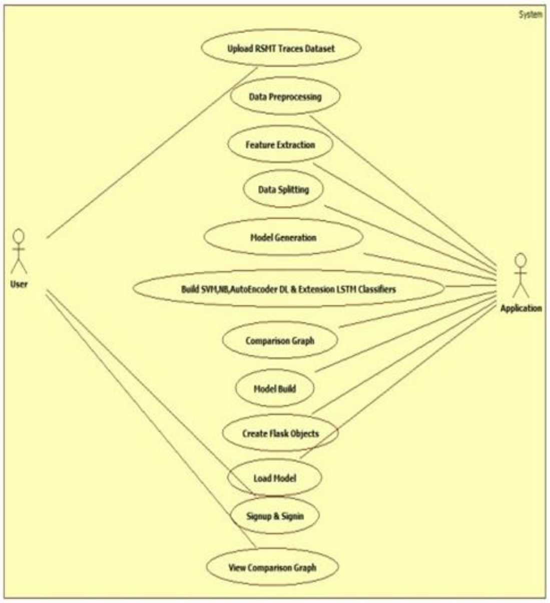
3. A DFD shows how information moves through the system and how it's modified through a series of metamorphoses. It's a graphical fashion that depicts the inflow of information and the metamorphoses that are applied as data moves from input to affair.

4. DFD is also known as bubble map. DFDs can be used to represent a system at any position of abstraction. DFDs can be divided into situations that represent adding information inflow and functional detail.

XI. UML DIAGRAM

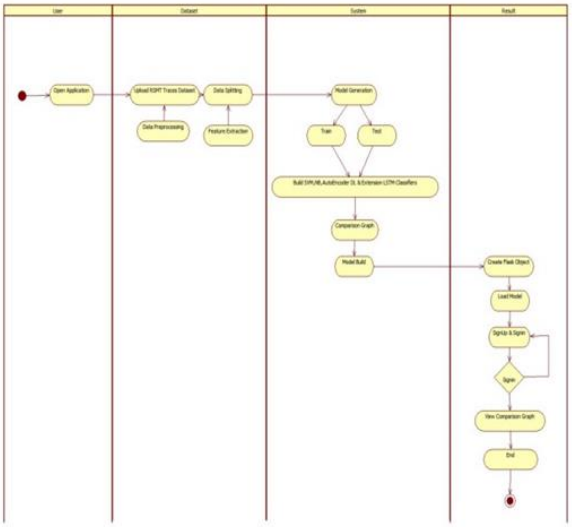
UML stands for Unified Modeling Language. UML is a standardized general modeling language in the field of object- acquainted software engineering. The standard is maintained and created by the Object Management Group. The thing is for UML to come a common language for modeling object- acquainted computer software. In its current form, UML consists of two main factors the Meta- model and the memorandum. Some form of system or process may also be added in the future or associated with UML. The Unified Modeling Language is a standard language for specifying, imaging, constructing, and establishing software system vestiges, as well as for business modeling and other non-software systems. UML represents a set of engineering stylish practices that have proven successful in modeling large and complex systems. UML is a veritably important part of object- acquainted software development and the software development process. UML substantially uses graphic memos to express the design of software systems.

XII. USE CASE DIAGRAM



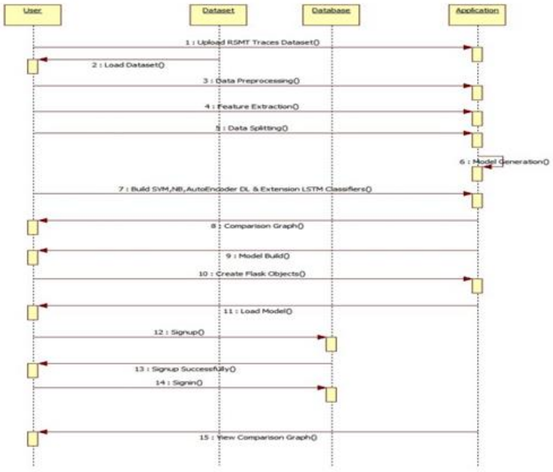
XIII. ACTIVITY DIAGRAAM

Process stream in a framework is captured in an movement diagram. Like a state graph, an action graph too includes exercises, activities, moves, starting and last states, and watch conditions.



XIV. SEQUENCE DIAGRAM

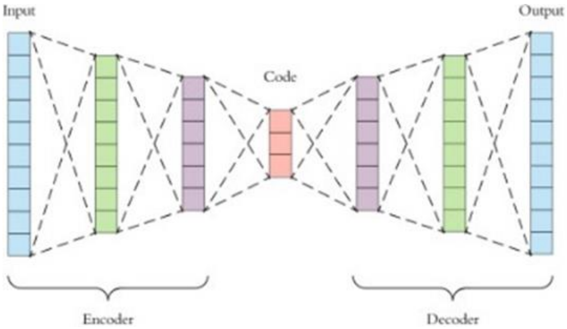
A arrangement graph appears the interaction between different objects in a framework. An critical viewpoint of a sequence chart is that it is time-ordered. This implies that a certain grouping of intuitive between objects is spoken to step by step. Diverse objects in a sequence chart connected with each other by passing "messages".



XV. IMPLEMENTATION

Auto encoders:

Imagine a neural network that compresses an image into a smaller version and then tries to recreate the original image from that compressed data. That's an autoencoder in action! By learning these compressed representations, autoencoders can be used for data cleaning or feature extraction for other machine learning models.



Support Vector Machines (SVMs):

Think of an SVM as a powerful classifier that separates data points in a high-dimensional space. Imagine a two-dimensional dataset where an SVM draws a clear line to divide different categories. SVMs excel at classification tasks, especially when dealing with clear separations between data points.

Naive Bayes:

This method relies on Bayes' theorem, a powerful tool for calculating probabilities. Naive Bayes assumes independence between features, meaning the presence of one feature doesn't influence another. This allows for efficient calculations of class probabilities, making it a popular choice for tasks like spam filtering.

Long Short-Term Memory (LSTM):

Unlike standard neural networks, LSTMs have a special talent for remembering things. These "artificial memory machines" are particularly adept at handling sequential data, like speech or text. LSTMs can analyze these sequences, identify patterns, and even make predictions based on past information.

XVI. MODULES

TensorFlow:

This open-source library is a powerhouse for dataflow programming and machine learning. It excels at symbolic math calculations and building neural networks. Originally developed by Google for internal use, TensorFlow is now a popular choice for research and production tasks alike.

NumPy:
The foundation for many data science projects, NumPy provides high-performance tools for working with multidimensional arrays. It offers sophisticated functions for calculations and integrates seamlessly with C/C++ and Fortran code. Beyond scientific computing, NumPy's efficient data containers can handle various data types, making it adaptable to diverse databases.

Pandas:
Often considered the workhorse of data analysis, Pandas offers high-performance data structures like DataFrames and Series. It streamlines the typical data processing workflow, including data preparation, manipulation, modeling, and analysis. Its ease of use has made Python with Pandas a go-to combination across various fields, from finance to academia.

Matplotlib:
For creating publication-quality visualizations, Matplotlib reigns supreme. This Python library generates various plot types, including scatter plots, histograms, and bar charts. It works seamlessly across environments, from scripts to web applications, and offers an intuitive interface for both beginners and advanced users.

Scikit-learn:
This library simplifies the implementation of machine learning algorithms. It provides a consistent interface for a wide range of supervised and unsupervised learning tasks. With its permissive license, Scikit-learn is freely available for academic and commercial use, making it a popular choice for data scientists of all backgrounds.

XVII. SOFTWARE ENVIRONMENT

Anaconda software helps you build environments for many different Python versions and package versions. Anaconda is also used to install, remove, and upgrade packages in your project environment. Additionally, you can use Anaconda to deploy any project you want with just a few mouse clicks. That's why it's perfect for beginners who want to learn Python.

XVIII. SYSTEM TESTING

System testing, also known as system-level tests or system-integration testing, is the process in which a quality assurance (QA) team evaluates how the various components of an application interact together in a complete, integrated system or application. System testing verifies that the application performs the designed functions. This step, a type of black box testing, focuses on the functionality of the application. System testing, for example, can check that each type of user input produces the desired output throughout the application.

XIX. UNIT TESTING

Unit testing, a testing technique in which individual modules are tested to see if problems are encountered by the developer himself. It is about the functional correctness of the individual modules. The main objective is to isolate each unit of the system for fault identification, analysis and repair.

XX. DATA FLOW TESTING

Data flow testing is a family of testing strategies based on choosing paths through the control flow of a program to examine the sequence of events related to the state of variables or data objects. Data flow testing focuses on the points at which variables are received and the points at which those values are used.

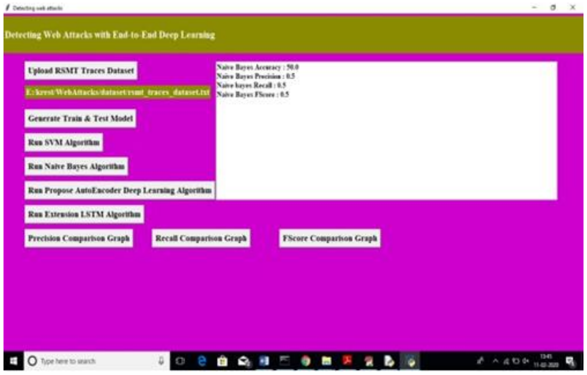
XXI. INTEGRATION TESTING

Integration testing, which is done after unit testing is completed, units or modules are to be integrated, which also enhances integration testing. The purpose of integration testing is to verify functionality, performance, and reliability between the modules that are being integrated.

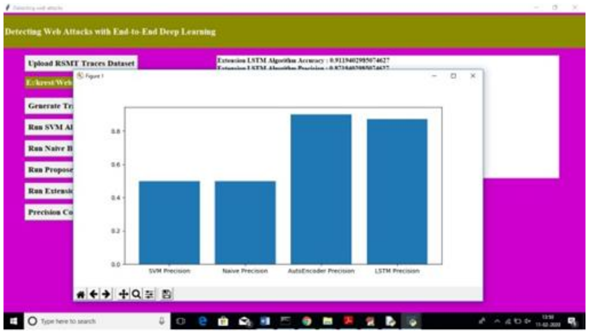
XXII. BEHAVIORAL TESTING

The final phase of testing focuses on the software's reactions to various activities rather than the mechanisms behind these reactions. In other words, behavioral testing, too known as black-box testing, presupposes running various tests, for the most part manual, to see the item from the user's point of see. QA engineers usually have some specific information about the business or other purposes (the 'black box') of the software to run usability tests, for example, and react to bugs as regular users of the product would. Behavioral testing can moreover include computerization (relapse tests) to kill human blunder if monotonous exercises are required. For example, you may need to fill 100 registration forms on a website to see how the product copes with such activity, so automating this test is better.

XXIII. RESULT

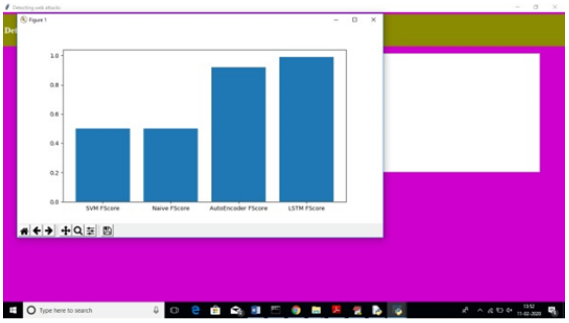

Detecting web attacks
Detecting Web Attacks with End-to-End Deep Learning
Upload RSMT Traces Dataset
E:/kresl/WebAttacks/dataset/rsmt_traces_dataset.txt
Generate Train & Test Model
Run SVM Algorithm
Run Naive Bayes Algorithm
Run Propose AutoEncoder Deep Learning Algorithm
Run Extension LSTM Algorithm
Precision Comparison Graph
Recall Comparison Graph
FScore Comparison Graph
Type here to search

```
Naive Bayes Accuracy : 50.0
Naive Bayes Precision : 0.5
Naive bayes Recall : 0.5
Naive Bayes FScore : 0.5
```



```
Extension LSTM Algorithm Accuracy : 0.9119402985074627
```

In the graph above, the x-axis represents the name of the algorithm and the y-axis represents the accuracy value. In all algorithms, design an AutoEncoder showing good performance.



In the graph above, the x-axis represents the name of the algorithm and the y-axis represents the FScore value. In all algorithms, Extension LSTM shows good performance.

CONCLUSION

This study explores a novel, unsupervised deep learning approach to automatically identify attacks on web applications. Unlike traditional methods requiring labeled data, this system can learn normal behavior patterns without prior examples of attacks. The secret lies in a tool called RSMT, which continuously monitors web applications and captures their actions. This data is then fed into a specialized neural network called a denoising autoencoder. This network acts like a smart compression tool, analyzing the captured data (call paths) and learning to create a low-dimensional representation that retains key characteristics. To validate their system's effectiveness, the researchers created test web applications and simulated attack scenarios. While traditional validation techniques may not be ideal for deep learning due to computational costs, the researchers were able to assess the system's unsupervised learning performance against this data. Overall, this research offers a promising avenue for securing web applications. The ability to learn and detect attacks without extensive labeled data makes this unsupervised deep learning approach a valuable tool in the ongoing fight against cyber threats.

REFERENCES

[1] Halfond WG, Viegas J, Orso A. A classification of sql-injection attacks and countermeasures. In: Proceedings of the IEEE International Symposium on Secure Software Engineering. IEEE; 2006. p. 13–5.

[2] Wassermann G, Su Z. Static detection of cross-site scripting vulnerabilities. In: Proceedings of the 30th International Conference on Software Engineering. ACM; 2008. p. 171–80.

[3] Di Pietro R, Mancini LV. Intrusion Detection Systems vol. 38: Springer; 2008.

[4] Qie X, Pang R, Peterson L. Defensive programming: Using an annotation toolkit to build dos-resistant software. ACM SIGOPS Oper Syst Rev. 2002;36(SI):45–60.

[5] https://doi.org/https://www.acunetix.com/acunetix-web-applicationvulnerability-report-2016. Accessed 16 Aug 2017.

[6] https://doi.org/http://money.cnn.com/2015/10/08/technology/cybercrime-cost-business/index.html. Accessed 16 Aug 2017.

[7] https://doi.org/https://www.consumer.ftc.gov/blog/2017/09/equifaxdata-breach-what-do.Accessed16August-2017.

[8] https://doi.org/https://theconversation.com/why-don't big-companieskeep-their-computer-systems-up-to-date-84250. Accessed 16 Aug 2017.

[9] Ben-Asher N, Gonzalez C. Effects of cyber security knowledge on attack detection. Comput Hum Behav. 2015;48:51–61.

[10] Japkowicz N, Stephen S. The class imbalance problem: A systematic study. Intell Data Anal. 2002;6(5):429–49.

[11] Liu G, Yi Z, Yang S. A hierarchical intrusion detection model based on the pca neural networks. Neurocomputing. 2007;70(7):1561–8.

[12] Xu X, Wang X. An adaptive network intrusion detection method based on pca and support vector machines. Advanced Data Mining and Applications. 2005;3584:696–703.

[13] Pietraszek T. Using adaptive alert classification to reduce false positives in intrusion detection. In: Recent Advances in Intrusion Detection. Springer; 2004. p. 102–24.

[14] Goodfellow I, Bengio Y, Courville A. Deep Learning: MIT press; 2016.

[15] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2012. p. 1097–105.