VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

INFORMATION SECURITY MANAGEMENT

WINTER SEMESTER 2021-2022

TITLE:

Online Communication and Vulnerability Check

PROJECT REPORT BY:

ANANSHA SHARMA - 19BCE2170

NIPUN PUNDHIR - 19BCB0014

ANUKRITI SINGH - 19BCE2156

SLOT: L21+L22

SUBMITTED UNDER SUPERVISION OF:

PROF. RUBY D.

ABSTRACT:

Thousands of security vulnerabilities are discovered in production software each year, either reported publicly to the Common Vulnerabilities and Exposures database or discovered internally in proprietary code. Vulnerabilities often manifest themselves in subtle ways that are not obvious to code reviewers or the developers themselves. With the wealth of open source code available for analysis, there is an opportunity to learn the patterns of bugs that can lead to security vulnerabilities directly from data. Successful cyber-attacks are caused bythe exploitation of some vulnerabilities in the software and/or hardware that exist in systems deployed on-premises or in the cloud. Although hundreds of vulnerabilities are discovered every year, only a small fraction of them actually become exploited, thereby there exists a severe class imbalance between the number of exploited and non exploited vulnerabilities.

This project deals with vulnerability detection in the online communication sector. At first, we use various vulnerability detection tools to detect the system vulnerabilities and at a later stage, we deploy a model by applying suitable algorithms to the verified datasets so as to automate the process of vulnerability detection and to also increase the output efficiency.

LITERATURE SURVEY:

Sr. No.	Research Paper Topic, Name of author and year of publication	Technology used	Outcomes of Research	Drawbacks
1.	Using Machine Learning for Vulnerability Detection and Classification(Tiago Baptista, Nuno Oliveira, Pedro Rangel Henriques)	FastScan using code2seq approach	The work described in this paper aims at developing a machine learning-based tool for automatic identification of vulnerabilities on programs (source, high-level code)	There are many tools that implement the concept of static analysis and apply it to vulnerability detection. Some tools rely on only lexical analysis like FlawFinder 2 but have the tendency to output many false positives.

2.	An Improved Vulnerability Exploitation Prediction Model with Novel Cost Function and Custom Trained Word Vector Embedding Mohammad Shamsul Hoque 1, Norziana Jamil, Nowshad Amin and Kwok-Yan Lam	Novel cost function and custom trained word vector.	In this research, they have designed a novel cost function feature to address the existing class imbalance. We also have utilized the available large text corpus in the extracted dataset to develop a custom-trained word vector that can better capture the context of the local text data for utilization as an embedded layer in neural networks	For these models, predicting the most exploitable vulnerabilities with supervised classification-based machine learning algorithms, the target label (binary class) has a severe imbalance (approximately 1 to 12) in favour of the major class (number of non-exploited vulnerabilities) since only a small fraction of published vulnerabilities have validated exploitation codes available in exploit databases
3.	Vulnerability Prediction from Source Code Using Machine Learning (Zeki Bilgin, Mehmet Akif Ersoy, Elif Ustundag Soykan, Emrah Tomur, Pinar Comak, Leyli Karacay) [2020]	Abstract Syntax Tree (AST) generation algorithm	The presented method extracts and then converts AST of a given source code fragment into a numerical array representation while preserving structural and semantic information contained in the source code. Thus, it enables us to perform ML-based analysis on source code through the resulting numeric array representation.	The model can not be used in all languages.
4.	Automated Vulnerability Detection in Source Code Using Deep Representation Learning (Rebecca L. Russell, Louis Kim, Lei H. Hamilton, Tomo Lazovich, Jacob A. Harer, Onur Ozdemir, Paul M. Ellingwood, Marc W. McConley) [2018]	Neural network classification and representation learning, Ensemble learning on neural representations	They built an extensive C/C++ source code dataset mined from Debian and GitHub repositories, labelled with curated vulnerability findings from a suite of static analysis tools, and combined it with the SATE IV dataset. They created a custom C/C++ lexer to create a simple, generic representation of function source code ideal for ML training. They applied a variety of ML techniques inspired by	Restricted to C language.

			classification problems in the natural language domain, fine-tuned them for our application, and achieved the best overall results using features learned via convolutional neural network and classified with an ensemble tree algorithm.	
5.	Toward large-scale vulnerability discovery using Machine Learning Gustavo Grieco, Guillermo Luis Grinblat, Josselin Feist, Sanjay Rawat	VDiscover, a tool that uses state-of-the-art Machine Learning techniques to predict vulnerabilities in test cases.	In this paper, they have presented an approach that uses lightweight static and dynamic features to predict if a test case is likely to contain a software vulnerability using machine learning techniques.	Only a few tools are able to operate on the binary code, suffering from a high percentage of false positives
6.	Automated software vulnerability detection with machine learning Jacob A. Harer, Louis Y. Kim, Rebecca L. Russell Onur Ozdemir, Leonard R. Kosta, Akshay Rangamani, Lei H. Hamilton, Gabriel I. Centeno, Jonathan R. Key, Paul M. Ellingwood, Erik Antelman, Alan Mackay, Marc W. McConkey, Jeffrey M. Opper, Peter Chin2, Tomo Lazovich	Deep Neural Networks	In this paper, they have presented a data-driven approach to vulnerability detection using machine learning, specifically applied object-oriented programming languages.	Detects a limited subset of possible errors using predefined rules

7.	Deep Learning for Software Vulnerabilities Detection Using Code Metrics (MOHAMMED ZAGANE, MUSTAPHA KAMEL ABDI, AND MAMDOUH ALENEZI)	DL-based AVP	DNN model gets very good vulnerability detection performances in terms of all TABLE 3. Results using the balanced dataset. TABLE 4. Results in terms of additional performance indicators. TABLE 5. Results using LSTM. performance indicators (precision: 74.0% - 76.9 % and recall: 73.4% - 76.6%). Obtained values in terms of	A vulnerability scanning tool will not find nearly all vulnerabilities. False positives

			FP Rate and FN Rate are slightly higher (23.36% - 26.56%) but they still in the range of acceptable values.	
8.	Survey on Vulnerability Prediction from Source Code by using Machine Learning Algorithm (B. DEEPTHI, DR.K. KUMAR)	Machine learning techniques of Support vector machines (SVM) and Naïve Bayes (NB) techniques are used to prevent the vulnerability	The presented method extracts and then converts AST of a given source code fragment into a numerical array representation while preserving structural and semantic information contained in the source code. Thus, it enables them to perform ML-based analysis on source code through the resulting numeric array representation.	False positives
9.	Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures (Matt Fredrikson, Somesh Jha, Thomas Ristenpart)	Machine-learnin g (ML) algorithms	They experimentally show attacks that are able to estimate whether a respondent in a lifestyle survey admitted to cheating on their significant other and, in the other context, show how to recover recognizable images of people's faces given only their name and access to the ML model.	Implications of vulnerability unclear
10.	An Automatic Source Code Vulnerability Detection Approach Based on KELM (Gaigai Tang, Lin Yang, Shuangyin Ren, Lianxiao Meng, Feng Yang and Huiqiang Wang)	Extreme Learning Machine (ELM)	They introduced the kernel method to improve the precision of ELM. Experimental results show that ELM with the kernel method is an effective combination of both efficiency and precision. Particularly, for the data preprocessing issue, they find that vector representation using doc2vec performs well on large datasets, and an appropriate symbolization level can effectively improve the precision of vulnerability detection. -ese experimental conclusions will provide researchers	From more than one kind of single-layer feedforward neural network that could be used for vulnerability detection, we only used ELM in this work

			and engineers with guidelines when choosing neural networks and data preprocessing methods for vulnerability detection.	
11.	A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments(Yuchong Li , Qinghui Liu)	Tools for penetration testing	cyber threats are not limited to governments, but individuals and companies will not be immune to the harms of these threats. Sixth, since security in the information age is not merely governmental, the various theoretical approaches in international relations whose theories are based primarily on government are easily overlooked or confusing.	None

12.	Automated Software Vulnerability Detection Based on Hybrid Neural Network (Xin Li, Lu Wang, Yang Xin, Yixian Yang, Qifeng Tang and Yuling Chen)	Hybrid Neural Network, Vulnerability Detection	The programs are transformed into intermediate representations first. LLVM IR and backward program slicing are utilized. The transformed intermediate representation not only eliminates irrelevant information but also represents the vulnerabilities with explicit dependency relations. Then, a hybrid neural network is proposed to learn both the local and long-term features of a vulnerability. The experiment results show that our approach outperforms state-of-the-art methods.	The method is applied to detect vulnerabilities in source code written in C language at present. The approach is only conducted on the SARD dataset due to the lack of labelled vulnerability datasets and falls into in-project vulnerability detection. The lack of labelled datasets is an open problem restricting the development of automated vulnerability detection technology.

13.	Machine Learning-Based Network Vulnerability Analysis of Industrial Internet of Things (Maeda Zolanvari, Marcio A. Teixeira, Lav Gupta, Khaled M.Khan, Raj Jain)	The paper analyses the ML-based IDS for ICS Vulnerabilities	They have represented how machine learning is capable of filling the identified gap by handling new types of attacks such as backdoor, command injection and SQL injection. Feature importance ranking was also studied to highlight the most salient features in distinguishing the attack traffic from normal traffic. The testbed built for this research work was designed to be as similar as possible to a real-world IIoT scenario.	False negatives, even a low number of them, mean malicious exertions against the system that stayed undetected and could lead to catastrophic results.
14.	Survey on Vulnerability Prediction from Source Code by using Machine Learning Algorithm(B. DEEPTHI, DR.K. KUMAR)	Support vector machines (SVM), Neural networks	The presented method extracts and then converts AST of a given source code fragment into a numerical array representation while preserving structural and semantic information contained in the source code. Thus, it enables us to perform ML-based analysis on source code through the resulting numeric array representation. To examine the presented source code representation technique for different objectives rather than vulnerability prediction, such as similarity analysis and code completion. and improve localization and interpretation aspects of the vulnerability prediction by using Support Vector Machine Learning (SVM) and Neural Networks.	A fully end-to-end prediction system from raw input data (code tokens) to vulnerability outcomes is not present.
15.	Using Machine Learning to Detect Software Vulnerabilities (Mingyue Yang)	LLVM Language Representations , Vulnerability Databases, Machine Learning (Bayesian Network, Naive Bayes, Logistic	They use machine learning algorithms to learn vulnerability patterns in code, and thus predict vulnerable code for further analysis. We experiment with two approaches: coarse-grained statistical features and raw feature	The quality and size of the vulnerability dataset still limit the performance for both techniques they propose.

		Regression, Neural Network, Random Tree, Random Forest, Bidirectional LSTM, Word2Vec Skip-gram Model)	representation for sliced code. Coarse-grained statistical features tradeoff the expressiveness of the model for a smaller amount of data required, while the raw feature representation overfits on the training set due to increased complexity/expressiveness of the model.	
16.	Artificial Intelligence in Cyber Security(Matthew N. O. Sadiku, Omobayode I. Fagbohungbe, and Sarhan M. Musa)	Applying AI in the following four areas: automated defence, cognitive security, adversarial training, parallel and dynamic monitoring.	Artificial intelligence has become a growing area of interest and investment within the cybersecurity community. Some early AI adopters include Google, IBM, Juniper Networks, Apple, Amazon, and Balbix. An increasing number of companies and organizations are jumping on the AI bandwagon. As cyberattacks grow, artificial intelligence is helping security operations analysts stay ahead of threats. AI automated systems will soon become an integral part of cybersecurity solutions, but it will also be used by cybercriminals to do harm. The future of AI-enabled cybersecurity is very promising.	Although artificial intelligence tools could help fight cybercrime, the tools could be exploited by malicious hackers.
17.	Machine Learning for Computer Security(Philip K. Chan, Richard P. Lippmann)	Theoretical paper.	contains several research studies on how machine learning algorithms can help improve the security of computer systems.	An adversary can defeat a system that learns to automatically extract signatures to detect computer worms.
18.	Intrusion Detection System and Vulnerability Identification using various Machine Learning Algorithms(Gauri Vilas Rasane and Dr Sunil Rathod)	Naïve Bayes Algorithm, ANN(Artificial Neural Network),	Aims to design and develop an approach for Intrusion Detection for fast learning-based neural network as well as a machine learning approach to evaluate the proposed system evaluation on different network dataset that will produce the	There are no dynamic rules for strongly unknown attack detection in a vulnerable environment.

			classification accuracy of the system.	
19.	WEB APPLICATION VULNERABILITY DETECTION USING DYNAMIC ANALYSIS WITH PENETRATION TESTING (Sreenivasa Rao B, Kumar N)	TAINTED MODE MODEL (TMM), DYNAMIC ANALYSIS, PENTETERATION TESTING	This paper present an enhanced Tainted Mode Model that incorporates inter-module data flows. They also introduced a new approach to automatic penetration testing by leveraging it with knowledge from dynamic analysis. Penetration testing is focused on finding security vulnerabilities in a target environment that could let an attacker penetrate the network or computer systems.	Wireless vulnerabilities also add to the attack surface that can be exploited.
20.	Software Vulnerabilities, Prevention and Detection Methods: A Review(Willy Jimenez, Amel Mammar, Ana Cavalli)	Detecting Software Vulnerabilities (Static and Dynamc Techniques)	Intends to create new vulnerability detection methods based on models. In this manner, they can guarantee the reusability of the test cases and facilitate the transformation of these formal representations into the specific programming language of the tool used to perform the vulnerability detection.	Time-consuming if conducted manually. Not scalable.

21.	A Survey of Security Vulnerability Analysis, Discovery, Detection, and Mitigation on IoT Devices (Miao Yu, Jianwei Zhuge, Ming Cao, Zhiwei Shi and Lin Jiang)	Research on the Basic Framework of Vulnerability Analysis, Research on the Basic Framework of Vulnerability Analysis, Research on the Basic Framework of Vulnerability Analysis, Research on Vulnerability Mitigation	It describes the research background, including IoT architecture, device components, and attack surfaces. They reviewed state-of-the-art research on IoT device vulnerability discovery, detection, mitigation, and other related works. Then, point out the current challenges and opportunities by evaluation. Finally, they forecast and discuss the research directions on vulnerability analysis techniques of IoT devices.	Complexity and Heterogeneity of Device Limitations of device resources Closed-Source Measures
22.	Research on	K Nearest	They implement and	Does not work well

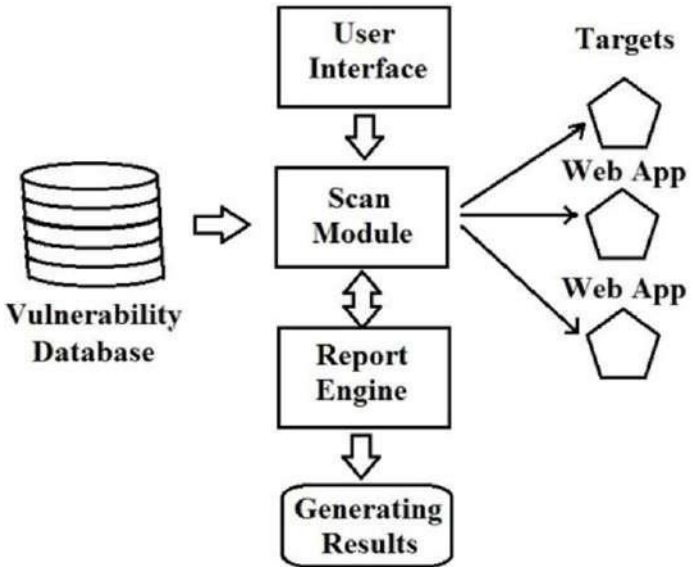
	Vulnerability Mitigation(Olufogoreh a Tunde-Onadele, Jingzhu He, Ting Dai, Xiaohui Gu)	Neighbors (k-NN), K-means	evaluate a set of static and dynamic vulnerability attack detection schemes using 28 real-world vulnerability exploits that widely exist in docker images. Their results show that the static vulnerability scanning scheme only detects 3 out of 28 tested vulnerabilities and dynamic anomaly detection schemes detect 22 vulnerability exploits. Combining static and dynamic schemes can further improve the detection rate to 86%.	with a large dataset as calculating distances between each data instance would be very costly.
23.	Detection of WordPress User Enumeration Vulnerability(Isrg Rajan)	GENERAL PREVENTION TECHNIQUE	Internet, web applications, mobile and computer applications are daily to daily usable utilities that as automated the entire process. Content management system like WordPress is powering nearly millions of websites, blogs and e-commerce websites and shockingly most of the people who are operating these websites are from a non-technical background this became possible just because of ease of usability, integrability and manageability. When millions of people are using these applications they not only invest money but also trust in that which could be compromised with loopholes and bugs.	A vulnerability scanning tool will not find nearly all vulnerabilities Constant updates required False positives
24.	Machine Learning for Web Vulnerability Detection: The Case of Cross-Site Request Forgery(Stefano Calzavara, Mauro Conti, Riccardo Focardi, Alvise Rabitti, and Gabriele	Web Vulnerability Detection	They propose a methodology to leverage Machine Learning (ML) for the detection of web application vulnerabilities.They use our methodology in the design of Mitch, the first ML solution for the black-box detection of	Constant updates are required. False positives

	Tolome)		Cross-Site Request Forgery (CSRF) vulnerabilities. Mitch allowed us to identify 35 new CSRFs on 20 major websites and 3 new CSRFs on production software.	
25.	Literature review on vulnerability detection using NLP technology(jiajie wu)	VULNERABILITY DETECTION USING NEURAL NETWORKS	This article does a brief survey of some recent new documents and technologies, such as CodeBERT, and summarizes the previous technologies.	False positives
26.	Detecting Software Vulnerabilities Using Neural Networks(AMY AUMPANSUB, USA ZHEN HUANG)	BLSTM, LSTM	They compared different types of training data and different types of neural networks. Their result shows that the model combining different types of characteristics of source code surpasses models based on the individual type of characteristics of source code. Using a balanced number of vulnerable program slices and non-vulnerable program slices ensures a balanced accuracy in predicting both vulnerable code and non-vulnerable code. They find that BGRU performs the best among other neural networks. Its accuracy reaches 94.89% with a sensitivity of 96% and a specificity of 91%.	The neural networks are trained with only program slices extracted from the source code of 14,000 C/C++ programs.
27.	Deep Neural Embedding for Software Vulnerability Discovery: Comparison and Optimization(Xue Yuan, Guanjun Lin, Yongkang Tai and Jun Zhang)	Research Framework., Code Representation Learning. Word2Vec, GloVe, FastText, and CodeBERT, Synthetic Data Fine Tuning, Evaluate the Impact of Fine-Tuned	This paper attempts to utilize CodeBERT which is a deep contextualized model as an embedding solution to facilitate the detection of vulnerabilities in C open-source projects. .e application of CodeBERTfor code analysis allows the rich and latent patterns within software code to be revealed, having the	Restricted to C language.

		CodeBERT with the Various Sequence Length	potential to facilitate various downstream tasks such as the detection of software vulnerability. This facilitates the learning of vulnerable code patterns which requires long-range dependency analysis. Additionally, the multi-head attention mechanism of the transformer enables multiple key variables of a data flow to be focused, which is crucial for analyzing and tracing potentially vulnerable data flaws, eventually, resulting in optimized detection performance.	
28.	AndroShield: Automated Android Applications Vulnerability Detection, a Hybrid Static and Dynamic Analysis Approach(Amr Amin, Amgad Eldessouki, Menna Tullah Magdy, Nouran Abdeen, Hanan Hindy and Islam Hegazy)	Vulnerability Detection Techniques (Static Analysis, Dynamic Analysis), Android Sources and Sinks	They proposed a usable Android vulnerability detection framework. The framework can be used by both developers and normal users. A web application is built to make it easy to use. The framework analyzes any uploaded APK file by two methods: static analysis and dynamic analysis and generates an analysis report. The types of vulnerabilities that we detected in our project were Information Leaks, Intent Crashes, Insecure Network Requests (HTTP Requests), Exported Android Components, Enabled Backup Mode, and Enabled Debug Mode.	Model improvement to accommodate more vulnerabilities is required.
29.	A study on Penetration Testing Using Metasploit Framework (Pawan Kesharwani, Sudhanshu Shekhar Pandey, Vishal Dixit, Lokendra Kumar Tiwari)	Metasploit	This paper discussed a three-phase methodology consisting of test preparation, test, and test analysis phase. The test phase is done in three steps: information gathering, vulnerability analysis, and vulnerability exploit. This phase can be done manually or using	None

			automated tools	
30.	Smart Contract Vulnerability Detection Using Graph Neural Networks (Yuan Zhuang, Zhenguang Liu, Peng Qian, Qi Liu, Xiang Wang, Qinming He)	Graph Neural Networks	In this paper, they have proposed a fully automated vulnerability analyzer for smart contracts. In contrast to existing methods, they explicitly model the fallback mechanism of smart contracts, consider rich dependencies between program elements, and explore the possibility of using novel graph neural networks for vulnerability detection. Extensive experiments show that our method significantly outperforms state-of-the art methods and other neural networks.	Graph Structure Limitation Noise Limitation

ARCHITECTURE DIAGRAM:



Modelling of data:

INPUT DATA:

The dataset consists of CVE(Common Vulnerabilities and Exposures) data. It consists of cwe code, cve, modification date, publication date, cvss, cwe name, Summary and category. This data denotes the commonly found vulnerabilities along with their ids and the degree of threat they pose to a system.

```
[ ] data
```

	cwe_code	cve	mod_date	pub_date	cvss	cwe_name	summary	category	CategoryId
0	352	CVE-2019-16548	21-11-2019 15:15	21-11-2019 15:15	6.8	Cross-Site Request Forgery (CSRF)	A cross-site request forgery vulnerability in ...	Medium	0
1	732	CVE-2019-16547	21-11-2019 15:15	21-11-2019 15:15	4.0	Incorrect Permission Assignment for Critical ...	Missing permission checks in various API endpo...	Medium	0
2	639	CVE-2019-16546	21-11-2019 15:15	21-11-2019 15:15	4.3	Authorization Bypass Through User-Controlled Key	Jenkins Google Compute Engine Plugin 4.1.1 and...	Medium	0
3	79	CVE-2013-2092	20-11-2019 21:22	20-11-2019 21:15	4.3	Improper Neutralization of Input During Web P...	Cross-site Scripting (XSS) in Dolibarr ERP/CRM...	Medium	0
4	89	CVE-2013-2091	20-11-2019 20:15	20-11-2019 20:15	7.5	Improper Neutralization of Special Elements u...	SQL injection vulnerability in Dolibarr ERP/CR...	High	1
...	...	...	...	...	...	...	...	...	...
19995	20	CVE-2019-0921	20-05-2019 19:01	16-05-2019 19:29	4.3	Improper Input Validation	An spoofing vulnerability exists when Internet...	Medium	0
19996	119	CVE-2019-0929	20-05-2019 18:36	16-05-2019 19:29	7.6	Improper Restriction of Operations within the...	A remote code execution vulnerability exists w...	High	1
19997	264	CVE-2019-0727	20-05-2019 18:35	16-05-2019 19:29	7.2	Permissions Privileges and Access Controls	An elevation of privilege vulnerability exists...	High	1
19998	264	CVE-2019-0938	20-05-2019 18:29	16-05-2019 19:29	6.8	Permissions Privileges and Access Controls	An elevation of privilege vulnerability exists...	Medium	0
19999	284	CVE-2019-5955	20-05-2019 18:26	17-05-2019 16:29	5.8	Improper Access Control	CREATE SD official App for Android version 1.0....	Medium	0

20000 rows × 9 columns

ALGORITHM:

The dataset was divided into training and testing which was then sent passed on as input to various models and their test accuracy, precision, F1 score and Recall values were calculated and tabulated for comparison purposes.

Algorithms used:

- ➔ Logistic Regression
- ➔ Random Forest
- ➔ Multinomial Naive Bayes
- ➔ Support Vector Classifier
- ➔ K-nearest Neighbour
- ➔ Gaussian Naive Bayes