International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020

A Framework for Web Application Vulnerability Detection

Asra Kalim, C K Jha, Deepak Singh Tomar, Divya Rishi Sahu

Abstract: ***Hardly a facet of human life is not influenced by the Internet due to the continuous proliferation in the Internet facilities, usage, speed, user friendly browsing, global access, etc. At flip side, hackers are also attacking this digital world with new tactics and techniques through exploiting the web application vulnerabilities. The analysis of these vulnerabilities is of paramount importance in direction to secure social digital world. It can be carried out in two ways. First, manual analysis which is error prone due to the human nature of forgiveness, dynamic change in technology and fraudulence attack techniques. Second, through the existing web application vulnerability scanners that sometime may suffer from generating false alarm rate. Hence, there is a need to develop a framework that can detect different levels of vulnerabilities, ranging from client side vulnerabilities, communication side vulnerabilities to server side vulnerabilities. This paper has carried out the literature survey in direction of identifying the new attack vectors, vulnerabilities, detection mechanism, research gaps and new working areas in same field. Continuous improvement in framework is easy. Hence, a framework is proposed to overcome the identified research gap.***

Keywords: Web Vulnerability, Web Malwares, Vulnerability Databases, Web Vulnerability Scanners, Webb Application Analysis

I. INTRODUCTION

As per the report of CERT-in, over fifty three thousand eighty one security incidents were handled including twenty nine thousand five hundred and eighteen website defacements in 2017 [1]. Recent vulnerability notes-2019 reported multiple vulnerabilities in phpMyAdmin, remote code execution in WebSphere Application Server, Data breach via malware on IoT that may exploit the SQL injection and cross site request forgery at victim's machine. In recent last year malwares like ransomware, WannaCry comes in account and flare-up the accounts and breach the credentials. In consequence online service providers down for a while. There are more new versions of web site malwares are formed by attackers day by day. Few of the recent website attacks are WannaCry & Petya ransomware, NotPetya malware, attacks on the Ethereum app, Equifax breach, the Yahoo email hack, DoS attack on GitHub etc.

Revised Manuscript Received on February 05, 2020.
* Correspondence Author
Asra Kalim*, CSE Department, Banasthali Vidyapeeth, Rajasthan India. Email: rhetoric1979@yahoo.com
C. K. Jha, CSE Department, Banasthali Vidyapeeth, Rajasthan India. Email: ckjha1@gmail.com
Deepak Singh Tomar, CSE, MANIT, Bhopal, India. Email: deepaktomarmanit@gmail.com
Divya Rishi Sahu, cse, SATI, Vidisha, India. Email: drsahu.cse@satiengg.org

The internet is teemed with attackers having malicious and criminal intentions and approaches novice users by newly invented attack vectors to compromise the security. Some are doing it for the thrill while many others have treacherous intentions like misusing data to earn easy money, theft, blackmail, criminal activities etc.

Hence, continuous efforts are required to identify the attack process, intention and solutions of these newly formed attack vectors. In this direction, profitable and non profitable consortiums, groups or agencies are working. It creates the database of identified attacks with attack definition, process, intention, risk factors, alerts and solutions to prevent from. Few of the standard agencies at national and international level are MITRE's Common Weakness Enumeration (CWE), CERT-In, WASC Threat Classification, US-CERT, SANS Institute, National Cyber security, CDAC, FFRDC, OWASP (Open Web Application Security Project), Centre for Internet Security (CIS), etc.

Times to time various efforts are made by security experts to point out these vulnerabilities and incorporate the solutions in the existing scanners. But, there is need to perform deep scrutinization and effective vulnerability analysis, to handle the emerging web application threats. It requires continuous efforts to scan all databases and summarize the solutions to make update the vulnerability database definition of the existing scanners. It only can be done by the developer or provider of the scanner.

In this paper, first background analysis and literature survey has been carried out to-

a. Identify more research areas in the field of web application security and forensics.
b. Explore the existing, open source vulnerability analysis databases that are available for simulating and testing web application attacks and procedures.
c. Study various web application security standards and standards for vulnerability analysis
d. Assess the existing tools and techniques of attack detection on web environments.

Findings are summarized in various sections. Finally, a framework has been proposed, according to the attack scenarios that attacker may launch to exploit the web application vulnerabilities. Proposed framework can easily incorporate the detection solutions of new identified attack scenarios and vulnerabilities.

II. WORKING AREAS IN WEB SECURITY

Web application vulnerability analysis is the field of current research having broad scope in field of code vulnerability analysis, train developers and users to be secure, identifying new attack trends, etc.

In the same direction, trusted and renowned consortiums and sole entities are working as mentioned in introduction section. They are maintaining databases that can help future research.

Emerging areas including the vulnerability analysis in which mentioned consortiums are working are as follows-

- Enumerates the attacks scenarios and vulnerabilities that can lead to the website compromise
- Vulnerability note creation for further study and generalize the data about cyber attacks and prevention.
- Predict cyber attacks before exploitation and generate preventive measures to prevent novice end user
- Discover the prevention and remediation coding steps that developers can take to mitigate or eliminate the weakness
- Prepare, release and promote guiding principles, standards, advisories, and vulnerability notes regarding attack scenarios, prevention steps, secure code development, data privacy, and forensically report the cyber incidents.
- Introduce and promote industry standards to develop secure code
- Develop the tools, techniques and framework for web vulnerabilities evaluation, attack detection, online anomaly behavior analysis, etc.

III. LITERATURE REVIEW

Literature survey has been carried out to explore the existing work and identify the research gap in the field of web application vulnerability analysis, their limitations, future work. Identified future scopes of existing tools and techniques are summarized in next section.

Rao et. al. [2] presented an extension of Mozilla Firefox browser and named as XBuster. It prevents end user from the attacks including XSS, clickjacking, partial script injection, attribute injection, and HTML injection. Extension converts coarse grained features of HTTP request and response into fine grained features in context of HTML and JavaScript.

Z. Li et. al. [3] introduced a new automatic tool vulnerability pecker (VulPecker). It is based on the concept to identify the vulnerabilities in source code through analyzing the cod-similarity in snippets. It works with various code-similarity patterns and algorithms for the reason that no single algorithm works effectively for all attack patterns. Author claimed that it can identify forty vulnerabilities out of the National Vulnerability Database (NVD).

Uwagbole et. al. [4] explored and classified previously known attack vectors to make vulnerable datasets. Thereafter, applied data mining technique on the well-known vulnerable datasets and trained the classifier to react as per situation. The trained classifier to be deployed as a web service that is implemented in a custom .NET application as a web proxy Application Programming Interface (API). Implemented web proxy is capable to predict SQLIA in http requests accurately. It rejects the http request containing malicious web scripts to unauthorized access of the web contents and database.

Guojun et. al. [5] introduced and developed a crawler which interacts dynamically with web application. Authors implemented data extraction rules in XML queries and created the database through XPath. These rules are fetched as per the attack vector of known vulnerabilities. The TF-IDF technique has been used to execute implemented queries as per need. Fetching of these queries on the requirement makes it dynamic. It also update rules for new attack vectors makes it intelligent.

Medeiros et. al. [6] presented a static source code analysis method using data mining and implemented a tool to identify web application vulnerabilities. It identifies the injection vulnerabilities in PHP programs through checking the input validation conditions. The tool has been tested on numerous known vulnerable PHP snippets and open source vulnerable web application projects.

Alqahtani et. al. [7] introduced unified ontology concept based web application vulnerability testing framework named as Security Vulnerabilities Evaluation Framework (SV-EF). It creates the bi-directional link between web vulnerability databases and frequently used code repositories. Established link may trace the vulnerabilities in code and its dependency type. At the outset, SV-EF is able to create the link of NVD vulnerability database to web application projects implemented by the Maven build repository. Next, it can be update for linking between other web application projects and vulnerability databases.

Bhor et. al. [8] proposed a tool to analyze the security concepts used by the developers in web application code. The tool is named as Distributed vulnerability and Attack Detection Tool in short DVADT. It checks the security concepts by first introducing vulnerabilities in the web application code and thereafter exploiting them through realistic attack vectors. Author experimented that, the tool for SQL injection, HTTP POST and ping of death attacks.

Satam et. al. [9] has presented fusion-based data analytical algorithm to identify the malicious snippets in web page or full web page. It analyse the files like .htm, .html files to identify the unauthorised files created by the attackers or web attacks. It classifies the structure of html files in malevolent and benevolent classes to validate it.

Maheshwari et. al. [10] presented a model on the basis of temporal classification and implemented the web intrusion detection system, works automatically in most cases. It can detect the SQL injection attack by capturing the data of injection attacks which create hurdles to access databases.

Meo et. al. [11] introduced a prototype named as SQLfast to identify the SQLi vulnerabilities in SQL database. Author first represented the SQLi attack vector through exploiting web application. Also, identifies an attack vector to exploit the Joomla a content management system. Thereafter, train the implemented model according to the test case results to detect such type of attack vectors. Author claimed that no other tools are capable enough to identify it.

Zachara [12] presented a new model to identify the web application vulnerabilities according to the shared vulnerability databases of connected web servers. It is useful to detect automated attacks. Every connected web server analyzes the http requests and shared the collected database of harmful request with another connected server.

So every model can update their database in multiple of the connected servers. Author tested it on network of seven web servers operated on approx three million http requests. The presented method also focused on confidentiality of shared databases. But, in this method trust on the connected server is a big issue.

Attacker can introduce malicious server in connected network to inject malicious database of attack detection.

Razzaq et. al. [13] presented ontology based technique to prepare semantic rules and identify the vulnerabilities in web applications. It classifies the web application attacks through semantic rules suggested according to the effects and the rules of application layer protocols. It examines the certain part of the user request possibly malicious to exploit the attack. There is possibility to inject the attack scripts through unanalyzed part of the user request.

Nalawade et. al. [14] identified and compared the pros and cons of the existing web application attack detection and prevention tools and techniques. Author also, carried out the forensic analysis through collected rules of vulnerability detection, identified by the different researchers. Further, applied identified rules on forensic evidences such as cache memory, history file, saved cookie files, logged download items, database file, etc. Improved the accuracy by considering the evidences of private browsing such as database file stored in WebCacheV01.dat file, log files, specific disk areas, etc. These evidences can be extracted by carving tools like ESECarve, extracted deleted log entries by WEFA tool. Finally, various search techniques such as regular expression, keyword search, forensic analysis on the basis of time, has been applied and report has been created

Deepa et. al. [15] conducted literature survey of web application security techniques and presented the research gap. Mostly, focuses on the prevention techniques of injection vulnerabilities and business logic vulnerabilities. This paper also presented the web application vulnerabilities and their solutions during the development of the web application. Author also, highlighted the future perspectives of different web application security analyzers including the open source tools.

Mitropoulos et. al. [16] specifically focuses on detection techniques of injection attacks of web environment and proposes one model to point out the web vulnerabilities. In same direction first classifies the attacks and techniques according to the parameters such as development, deployment, security technique, and performance parameters. Finally, concluded that most of the detection techniques are not analyzed effectively. Most prevention techniques can easily escaped by getting the process of detection technique. Hence there is need to work more in web vulnerability detection and prevention.

IV. FUTURE SCOPE OF EXISTING SCANNERS AND APPROACHES

Web application analysis tools and techniques are used to identify web vulnerabilities in web pages with or without running the source code. These techniques are useful to write secure code during SDLC. Solution of vulnerabilities during development life cycle saves the testing time and cost. Most of the static web analysis scanners suffer from false positive and false negative cases due to unavoidable conditions. Such conditions can be resolved by the developers during development. Hence, web application scanners during SDLC help most to the developers to write secure code.

Researchers introduce various web application analysis techniques and tools such as Pixy, RIPS, XBuster, VulPecker, etc. Future scope or limitations of the existing scanners are concluded in Table 1.

Table 1: Possible Identified Future Scope of Existing Scanners

Sr. No	Scanner	Future Scope
1.	XBuster	It is developed for Mozilla Firefox client agent to prevent novice end user from attacks such as XSS, clickjacking, etc. It needs to be enhanced and made more effective, through approximate string matching [2].
2.	VulPecker	The databases tagged as Vulnerability Patch Database (VPD) and a Vulnerability Code Instance Database (VCID) need to be updated. It also be written for other languages, such as Java and Python. The scalability issue requires investigation, and vulnerabilities are detected at source code level. It also needs to detect whether a snippet has affected by known vulnerability or it is secure [3].
3.	Web Crawler	The intelligent dynamic crawlers are not event driven. It requires to updating regularly due to enhancement in structure of web contents, source of contents and distribution channels. It results in excessive burden of crawler updates, development and maintenance. Hence it requires in depth study and further work [6].
4.	DVADT	It needs to be further refined and enhanced since no single technique may offer prevention from all web application vulnerabilities. It implies that DVAT has scope to enhance it for zero day vulnerability [8].
5.	Vulnerability Scanners	Vulnerability scanner cannot crawl dynamic web contents because of lack of ability to analyze active data and complicated multimedia structures such as SilverLight, Flash, Java Applets, etc. These complexities require physical analysis of false positive and false negative cases. It can detect acknowledged patterns of attack samples due to unawareness of application context [16]. So, it cannot identify the logical vulnerabilities.
6.	Stranger	Stranger represents combination of three words STRing AutomatoN GEneratoR. It generates the signature in string form to detect the vulnerabilities in PHP 4 scripting codes. While now days PHP version 7.4 is widely held [17].

7.	Saner	Saner is a static scanner to analyze vulnerabilities in PHP script. It analyzes the flow of input data in code to identify the potentially vulnerable snippet. Although, it is lacking in support of object-oriented features of PHP scripting language [18].
8.	DevBug	DevBug is another static analysis scanner works in online mode. It is open source code scanner to analyze PHP based web applications. DevBug can identify ten types of web application vulnerabilities in code provided by user. It is developed in JavaScript and run at Linux OS [19].
9.	RATS	The RATS is short of Rough Auditing Tool for Security. It analyzes the PHP scripting snippets and tags the development errors which become vulnerability to exploit attack in web application. One of the common errors like buffer overflow. It is limited to identify the injection vulnerabilities like SQL Injection vulnerabilities [20].
10.	RIPS	The RIPS is the short of Re-Inforce Programming Security. It is static code scanner analyzes the vulnerabilities in PHP based web applications. It tokenizes the source code through built in tokenizer and analyze it through taint analysis to present lacunas in code snippets. RIPS 0.5 version was limited in features and type of vulnerability detection. RIPS version 0.55 has been released in 2017 [21-22].
11.	Pixy	Pixy statically analyzes the source code written in PHP 4 and tags the vulnerabilities. It parse the source code to analyze Taint-Style Vulnerabilities and XSS Attacks [23].

Further, the future scope or limitations of the analysis approaches used in existing web application scanners are concluded in Table 2.

Table 2: **Future scope or limitations of the approaches used in existing scanners**

1.	Online Anomaly Detection	It is used in web application scanners to identify the surprising variation in process of written code. It needs to be applied and explored for other source code features.
2.	Database Formalization	Database formalization support to understand the logical consequences between set of axioms introduced time to time in rational database. It needs to be enhanced to facilitated identification of web vulnerabilities such as second order SQL injection, Cross Site Scripting, CSRF, etc.
3.	Ontology based Techniques	Ontology based technique used in existing web application vulnerability scanners identifies the relation between code logic, tokens of snippet and its properties. It needs to create an ontology based rule generation technique to enhance it towards the commercial products.
4.	Data Flow based Technique	Data flow analysis in vulnerability detection for web application analyses the flow control of input value towards the potentially vulnerable sink. It creates the CFG for all possible function calls between the tokens of code. It needs to update for all possible attack vectors.
5	Taint based Analysis	Taint-style vulnerabilities occur due to the improper sanitization of the each input value passed by the web application. So, Taint based analysis is mainly focused on the identification of all possible updates occurs in code by the input data.

V. IDENTIFIED RESEARCH GAP

Following research gaps have been identified based on the comprehensive literature survey and conclusive findings.

- Standard security techniques need to be update towards providing the all possible security solutions rather only insisting the developer to employ secure coding standards.
- Necessity to develop a technique that may detect against new web application attacks with zero day vulnerabilities.
- Vulnerability detection system suffers from False Positives and False Negative rate. Hence, there is need for updating its detection database with newly coming vulnerabilities.
- Need to develop multi-class classifier to identify and prevent the second order SQLIAs and frame jacking as they are predicted.
- Also need to enhance the mechanism to identify and prevent XML/XPath injection attacks for the unstructured data which necessitates migration to XML based databases.
- Most of the existing web scanners are not capable enough to handle dynamic AJAX content properly. It suffers from issues related with managing the session for end users including identification of unpredictable & random token IDs, identify corresponding session for these IDs, session hijacking & fixation, etc.
- The various mechanisms to detect attacks such as CSP, HTML sanitization and eval handling are used widely but many still not used in practice.
- Security tools needs to effectively identify and tag the vulnerabilities by applying multiple algorithms to minimize the false positive and false negative rate automatically.
- Require ease of deployment that brings ease of experimentation.

VI. PROPOSED FRAMEWORK

Most web application vulnerability scanners are implemented in software code and it can be updated by expert developer having good knowledge of web security.

All these codes are having some conditions and related instructions to identify the threat and react accordingly. An advance AI engine has been proposed to automatically and easily modification in condition with related instructions according to newly encountered attack vectors. It updates instructions by learning from result and reporting the false positive and false negative cases.

Growing in same direction following research methodology has been proposed to overcome the first six identified research gaps discussed in section V. The underline architecture of proposed framework is drafted to identify the vulnerabilities in website as shown in Figure 1.

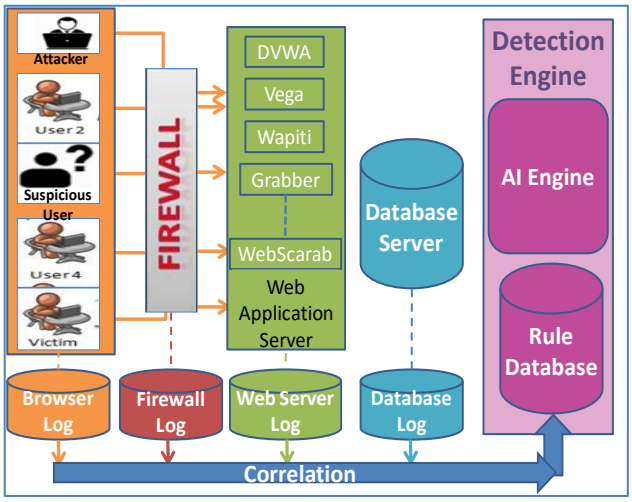

Figure 1: Architecture of Proposed Framework

Working flow to implement the proposed framework is as follows.

a. First, configure the web server to identify and analyze the vulnerabilities through vulnerable web applications such as DVWA (Damn Vulnerable Web App), Grabber, Vega, Wapiti, WebScarab.
   - The PHP modules will be created to develop some of the new attack scenarios such as frame jacking.
   - The activities of the web server at application layer will be captured by web server log.
   - The database attack such as SQL Injection, blind SQL Injection will be explored in MySQL Database and will be recorded in database log.

b. Second, setup the environment to build the attack tree that included new attack scenarios. It helps to monitor the behavior of attacker and update the vulnerability database and detection rules.

c. Third, explore machine learning technique to correlate log activities and build pattern as rule for AI engine.

d. Implement the detection engine that recognizes the security issues in web application snippets according to developed rule database.
   - Rule database can easily be updated as per the future requirements then framework may detect new attacks also.

e. Finally, fabricate an interactive and automatic IoT device compatible to computer system on the basis of step-a to step-d.

Benefits of the AI Firewalls over Static Firewalls-

- A static code scanner without interacting to human being cannot prevent users from the vulnerabilities like code injection. Whereas, a firewall with AI-engine can prevent novice end user from newly encountered vulnerabilities and automatically report it to expert team for future enhancements.
- AI & machine learning methods add extra wings in AI firewalls to web application vulnerability analysis and security.
- It would be transform into hardware product from the software product working on application layer.
- The AI engine manages and bound the system to work within the security limit and report the miscellaneous activities to the experts.

VII. RESULT ANALYSIS

From confusion matrix, Precision, Recall (True Positive Rate), Accuracy, False positive rate (FPR), and False Negative Rate are calculated for all five iteration as shown in Figure 2.

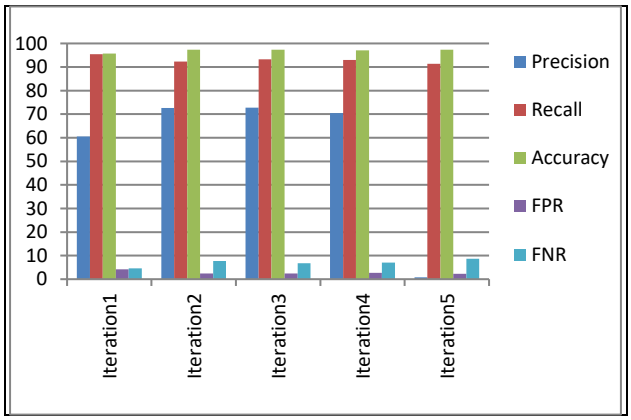

Figure 1: Performance Evaluation Matrices

It is illustrated that FPR and FNR in all iteration for standard dataset is very low and all the above factors are greater than 10 %. It means the accuracy level for all iteration is high. It signifies the accuracy in all iteration is high. It is not sufficient to analyze the accuracy of a graph because a large change in the number of false positives leads to a small change in the false positive rate.

The Accuracy, TPR, FPR, Precision and Recall are better in proposed model. The accuracy and model building time is also reduced in comparison to base classifier as shown in Figure 3.

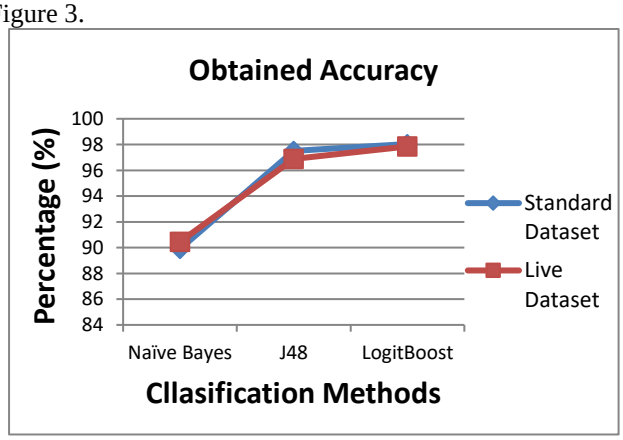

Figure 3: Obtained accuracy of both classifiers for standard dataset

The correctly identified malware rate of True Positive Rate for the standard as well as live dataset is compared and shown in Figure 4.

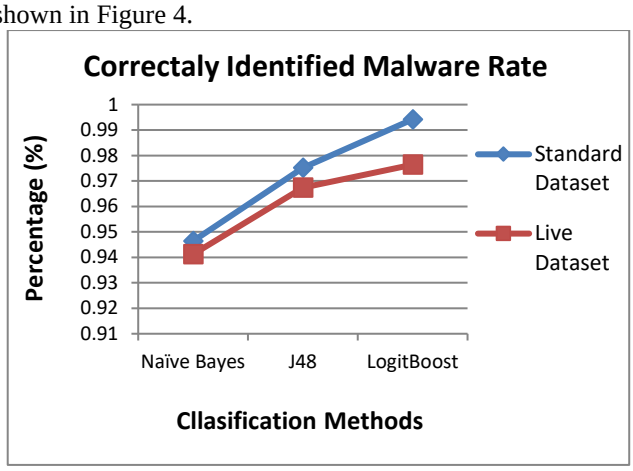

Figure 4: Correctly detected malware rate (TPR)

The figure represents that total correctly classified instances are 2448 hence its percentage is 98.0376 %. So, true positive rate for standard dataset in this dissertation is 98.0376 %. The incorrectly identified malware rate of False Positive Rate for the standard as well as prepared dataset is compared and shown in Figure 5.

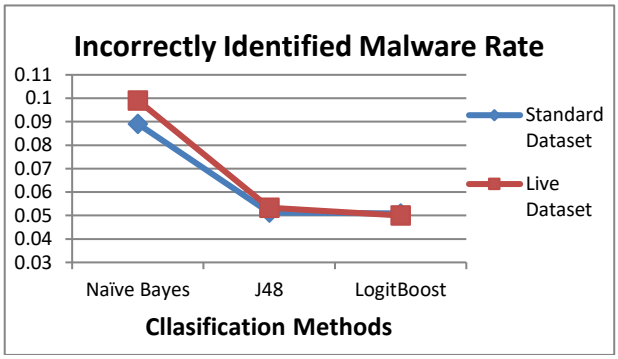

Figure5: Incorrectly detected malware rate (FPR)

Figure represents that incorrectly classified instances are 49 and its percentage is 1.9624 %.

VIII. CONCLUSION

In today's era one of the roles of web applications is to provide graphical user interface to the end users for communicating the devices through internet. Development and hosting of web application is too easy. Hence, new attack vectors are encountering frequently to breach the end user's information. Literature survey of this paper concluded that there is a need of an AI engine to update instructional database of vulnerability scanner automatically for newly encountering attack vectors. Hence, in this paper a framework to identify the taint style attacks has been proposed. It performs several types of scanning like taint type, ontology based, etc. to detect security vulnerabilities.

It updates its instructional database by the record gathered from the data flow analysis phase. It can identify the web vulnerabilities like XSS and SQLIAs, Frame-Jacking, Zero day vulnerabilities, etc. Proposed framework facilitates deeper understanding about attacker's behaviour/ intention on web application. It also facilitates to security experts and developers to easily update detection database as per new requirements. Finally, it generates a detailed report which contains a detailed explanation of each potential vulnerable function that represents security vulnerability in web application.

REFERENCES

1. Indian Computer Emergency Response Team (CERT-In), "Annual Report-2017", Ministry of Electronics & Information Technology, Government of India, 2018.
2. K. S. Rao, et Al., "Two for the price of one: A combined browser defense against XSS and clickjacking", IEEE, International Conference on Computing, Networking and Communications (ICNC), 2016 pp. 1-6.
3. Z. Li, et Al., "VulPecker: an automated vulnerability detection system based on code similarity analysis", ACM, Proc. of the 32 Annual Conference on Computer Security Applications, pp. 201-213, 2016.
4. S. O. Uwagbole, W. J. Buchanan, & L. Fan, "Applied machine learning predictive analytics to SQL injection attack detection and prevention", IEEE, Symposium on Integrated Network and Service Management (IM), 2017 IFIP/IEEE, pp. 1087-1090, 2017.
5. Z. Guojun, et. Al., "Design and application of intelligent dynamic crawler for web data mining" IEEE, In Automation (YAC), 2017 32nd Youth Academic Annual Conference of Chinese Association, pp. 1098-1105, 2017.
6. I. Medeiros, N. Neves, & M. Correia, "Detecting and removing web application vulnerabilities with static analysis and data mining", IEEE, IEEE Transactions on Reliability, Vol 65, Issue 1, pp. 54-69, 2016.
7. S. S. Alqahtani, E. E. Eghan, & J. Rilling, "SV-AF—A Security Vulnerability Analysis Framework", IEEE, 27th International Symposium on Software Reliability Engineering (ISSRE), pp. 219-229, 2016.
8. R. V. Bhor, & H. K. Khanuja, "Analysis of web application security mechanism and Attack Detection using Vulnerability injection technique", IEEE, International Conference on Computing Communication Control and automation (ICCUBEA), pp. 1-6, 2016.
9. P. Satam, D. Kelly, & S. Hariri, "Anomaly behavior analysis of website vulnerability and security", IEEE/ACS, 13th International Conference of Computer Systems and Applications (AICCSA), pp. 1-7, 2016.
10. K. G. Maheswari, & R. Anita, "An Intelligent Detection System for SQL Attacks on Web IDS in a Real-Time Application" Springer, Proc. of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC–16',) pp. 93-99, 2016.
11. F. De Meo, M. Rocchetto, & L. Viganò, "Formal analysis of vulnerabilities of web applications based on SQL injection", Springer, In International Workshop on Security and Trust Management, pp. 179-195, 2016.
12. M. Zachara, "Identification of possible attack attempts against web applications utilizing collective assessment of suspicious requests", Springer, In Transactions on Computational Collective Intelligence XXII, Berlin, Heidelberg, pp. 45-59, 2016.
13. A. Razzaq, et. Al., "Semantic security against web application attacks", Information Sciences, Vol. 254, pp. 19-38, 2014.
14. A. Nalawade, S. Bharne, & V. Mane, "Forensic analysis and evidence collection for web browser activity", IEEE, International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), pp. 518-522, 2016.
15. G. Deepa, & P. S. Thilagam, "Securing web applications from injection and logic vulnerabilities: Approaches and challenges", Information and Software Technology, Vol. 74, pp. 160-180, 2016.
16. D. Mitropoulos, P. Louridas, M. Polychronakis, & A. D. Keromytis, (2017). Defending against web application attacks: Approaches, challenges and implications. IEEE Transactions on Dependable and Secure Computing.
17. Yu. Fang, Alkhalaf Muath, & Tevfik Bultan, "Stranger: An automata based string analysis tool for PHP", Springer, In Tools and Algorithms for the Construction and Analysis of Systems, pp. 154-157. Berlin Heidelberg, 2010. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-12002-2_13
18. D. Balzarotti, et. Al. "Saner: Composing static and dynamic analysis to validate sanitization in web applications," Security and Privacy, SP 2008. IEEE Symposium, pp. 387- 401, 2008. [Online]. Available: http://dx.doi.org/10.1109/SP.2008.22

19. Ryan. "DevBug—PHP static code analysis" [Online]. http://www.devbug.co.uk/RATS
20. Secure Software Inc., "Rough auditing tool for security (RATS)" [Online]. https://code.google.com/p/rough-auditing-tool-forsecurity/
21. D. R. Sahu and D. S. Tomar, "Analysis of Web Application Code Vulnerabilities using Secure Coding Standards", Springer, Arabian Journal for Science and Engineering, Vol 42, Issue 2, pp 885–895, 2017.
22. J. Dahse & T. Holz, "Simulation of Built-in PHP Features for Precise Static Code Analysis," Symposium on Network and Distributed System Security (NDSS), 2014. [Online]. Available: http://dx.doi.org/10.14722/ndss.2014.23262.
23. N. Jovanovic, C. Kruegel & E. Kirda, "Pixy: A static analysis tool for detecting web application vulnerabilities," IEEE, IEEE Symposium on Security and Privacy, pp. 6-12., 2006. [Online]. Available: http://dx.doi.org/10.1109/SP.2006.29.

AUTHORS PROFILE

Asra Kalim Lecturer & Website Contents Manager at Jazan University, Saudi Arabia has pursuing PhD in CSE Department, Banasthali Vidyapith. She completed her Masters in Computer Application. Major research areas are Web App Security, Block-chain, IoT and Cloud Computing.

Prof. C. K. Jha Professor & HOD, Computer Science at Banasthali Vidyapeeth, Rajasthan India, has completed their M.C.A. and Ph.D. He has more than 20 years of teaching and administrative experience. In addition with he has the 7 years of experience in Indo Gulf Industries as Dy. Manager Systems. Major research areas are Networking, Data Mining, Big data, WSN, etc. on which he published more than 120 research papers.

Dr. Deepak Singh Tomar Associate Professor in CSE Department, MANIT, Bhopal (MP), India has completed PhD in Computer Science and Engg., and M.Tech & B.E. in Computer Technology. He has more than 25 years of teaching and administrative experience. Major research areas are Data Mining, Internet Technology, Computer & Network Security, Digital Forensics, Machine Learning on which he has published more than 77 research papers with 04 book chapters.

Dr. Divya Rishi Sahu Assistant Professor in CSE Department, SATI Vidisha (MP), India, has completed PhD in CSE and MTech in Information Security. He has more than 05 years of teaching and administrative experience. Major research areas are Web Security, Web Forensics, Data Mining & Machine Learning on which he has published 16 research papers with 02 book chapters.