International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2017): 7.296

A Survey on Web Security Mechanisms Using Vulnerability and Attack Injections

Manjunatha K M[1], Dr. M Kempanna[2]

[1]Research Scholar, Department of Computer Science, Bangalore Institute of Technology, Bengaluru, India

[2]Assistant Professor, Department of Computer Science, Bangalore Institute of Technology, Bengaluru, India

Abstract: ***World Wide Web is capable of delivering a broad range of sophisticated applications. Web application is one of the most powerful communication channels and provides a service for information delivery over internet today. Many web applications go through rapid development phases with extremely short turnaround time, making it difficult to eliminate vulnerabilities. We study the web security estimation methods in order to identify attacks and present a survey on web application by using different injection tools and discussing existing methods. The main goal is to find out the effectiveness papers for detecting the vulnerabilities and attacks in web application.***

Keywords: Web Security, Web Application, Vulnerabilities and Attacks Injection Tool

1. Introduction

Internet is full of information and data. This is a backbone of today's work environment. Internet with its hypertext data and information is known World Wide Web, called as WWW or just Web. Web is a common part of everyone's life. Many users are interacting with the web by using web app and browser. Web application depicts combined system of server, website host by back end database and server. Web application is used to serve the different user requirement in efficient and convenient manner because of increasing in numbers. But No. of web apps and number of attacks are increased on those applications. These applications are collaborate with users behind programs and act as indicated by user input, such interfaces must be able to both program and client [06]. Something else, the interface must process content that intended to be rendered by programs and later translated by people. At that point, with the assistance of a self-learning injection database, fault injection techniques were connected to distinguish injection vulnerabilities.

The key communication of Web application's convention is HTTP, can connectionless convention intended for adaptation to non-critical failure and heartiness rather than most extreme throughput communication [01]. Communication between a server and customer in Web application typically spin around a route of Web pages, not immediate interchanges amongst customer and server side items. At one level of idea, all informing in Web application is expressed as request and reception of Web page substances. For the most part, the engineering of a Web application isn't that much variety from that of dynamic Web webpage. The difference between a Web website and a Web application include its use. Web application executes business logic on security reason and its exploit changes the condition of business. This is more imperative since it characterizes a focal point of demonstrating exertion. Web applications perform business logic thus the most imperative models of the framework center around business logic and business state, not on introduction points of interest. Introduction is critical; notwithstanding, an unmistakable partition amongst business and introduction concern must be strived [07]. Moreover the assets that work on introduction have a tendency to be more creative, and less worried about the execution of business rules.

Web constitutes a primary access to more useful services with a security requirement. The vulnerabilities on web enable attacks with catastrophic consequence range from economic loss. For example attacks against to payment providers, privacy violations and PayPal [18]. Another example is improper of electronics health records. Critical security services are more supplied by online today and this will increases for web application.

Vulnerabilities of web application have been increased rapidly. Usually web attackers found weakness in source code of web app and manipulate the code by inserting the malicious in app. The result of web application performs differently as how intend to develop. If inserting some malicious attacks might increase the administrative manipulation and privileges, data change the remote command and unauthorised data. DOS attack and SQL attack are most important threads to find in web app for security [08].

Figure 1 depicts the general block diagram of Injection Tool. Firstly, internet block will interact with web server based on http interaction. Web server is used to manage the request and response. This Web Services have become dependable platform for ecommerce and many models. Web application manages database to process the data. Then, application identifies attacker by database.

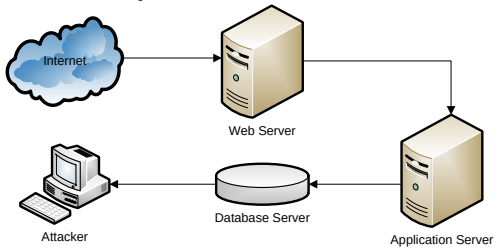

Figure 1: General block diagram of Injection Tool

In this section we presented an introduction of web security mechanisms by vulnerability and attack injection tool. Section II depicts the Different injection tool and Section III depicts the realistic vulnerably and attack types. Section IV depicts the literature survey on various vulnerabilities and attacks injection tool and gives a comparison table for existing systems. Finally, Section V depicts concluded survey on this paper.

2. Different Injection Tools

In this section, we present the different kinds of injection tool. The different types of attacks are generally not performed in isolation; many of them are used together or sequentially, depending on the specific goals of the attacker.

2.1 Code Injection

Code injection is a sort of misuse caused by handling invalid information input. The ideas of injection attacks are brought malignant code into a program so that to change the course of execution. Such attack might be performed by including series of malicious characters into information values in the frame or contention values in URL. This Injection is has a general name for different kinds of attacks which inject improper code into the script interpreter. This can be accomplished through various measurements, for example, web level, Application level and working framework level [24].

2.2 SQL Injection

SQL injection is a most devastating vulnerability that affects a business, as it can prompt presentation of the majority of the sensitive data put away in an application's database, including convenient data, for example, passwords, usernames, telephone numbers, names, addresses, and credit card details [12]. SQL injection is an attack in which the SQL code is appended or inserted into application/client input parameters that are later passed to a back-end SQL server for parsing and execution. Any strategy that builds SQL explanations could conceivably be vulnerable, as the differing idea of SQL and the strategies accessible for developing it give an abundance of coding alternatives [18]. The essential type of SQL injection comprises of direct inclusion of code into parameters that are linked with SQL orders and executed. A less immediate attack injects malevolent code into strings that are bound for capacity in a table or as metadata.

2.3 Email Injection

Email Header Injection has a place with an expansive class of vulnerabilities known as charge injection vulnerabilities. In Injection vulnerabilities, the attacks information can modify the charges executed. However, not at all like its more famous common injection sibling, SQL injection (where attacker input can change SQL orders), Cross-Site Scripting (where aggressor input can modify the HTML substance of a site page), and HTTP Header Injection (where attacker input can adjust the HTTP headers of a web application, generally minimal scholarly research is accessible on E-mail Header Injection vulnerabilities [21]. Likewise with different vulnerabilities in this class, E-mail Header Injection is caused because of disgraceful or missing sanitization of user input. In the event that the program develops messages from client input neglects to check for the nearness of email headers in the client input, a malicious user using a very much made payload can control the headers set for this specific email.

2.4 Frame Injection

A frame injection attack is an attack on web to discretionary code in program. This attack is caused by web not checking the goal of resultant edge. This permits when code gets injected through the frames because of contents not approving their input.

2.5 Fault Injection

Fault Injection is the approval method of reliability blame tolerant frameworks which comprises in the achievement of controlled analyses. The perception of the framework's conduct in nearness of deficiencies is initiated expressly by the written work presentation (injection) of flaws in the framework. The Fault injection procedures have been identified for quite a while as important to approve the steadfastness of a framework by breaking down the conduct of the gadgets when a fault happens [22]. A few efforts have been made to create procedures for injecting deficiencies into a framework model.

2.6 Blind Injection

At the point when the frameworks are more secure, the attacks may test for helpless parameters or concentrate information by utilizing this system. Blind injection enables the attacks to gather the database build through assessing articulations that are combined with proclamations that dependably assess to genuine or false [23].

2.7 Timing Injection

Programmers accumulate data of database by observing the planning postponement of database response. Programmers intentionally create if then injected inquiries that branch condition relates to address about the substance of database [23]. Each branch condition will cause the SQL inquiry be executed to be delay for determined time. Programmers may conclude which condition branches satisfy the injected question by observing the expansion or diminishing reaction and load the outcome page time.

3. Realistic Vulnerabilities and Attacks

In this section, we present vulnerabilities that might be inherent in web applications and that can be exploited by SQL injection attacks [23].

3.1 Invalidated input

Unchecked parameters to SQL queries that are powerfully fabricated can be utilized as a part of SQL injection attacks. These parameters may contain SQL watchwords, e.g.

INSERT or SQL control characters, for example, quotes and semicolons [23].

3.2 Error message feedback

Mistake messages that are produced by the RDBMS, or other server-side projects might be come back to the customer side and imprinted in the internet browser. While these messages can be helpful during improvement for troubleshooting purposes, they can likewise constitute dangers to the application. Attackers can break down these messages to acquire data about database or content structure so as to develop their attack [23].

3.3 Uncontrolled variable size

Factors that permit storage of information that is bigger than anticipated may enable attackers to enter changed or manufactured SQL explanations. Contents that don't control variable length may even open for different attacks, for example, buffer overwhelm [23].

3.4 Variable morphism

In the event that a variable can contain any information, it is workable for an aggressor to store other information than anticipated. Such factors are both of feeble kind, e.g. factors in PHP, or are naturally changed over starting with one kind then onto the next by the RDBMS, e.g. numeric qualities changed over into a string write. For instance, SQL catchphrases can be put away in a variable that ought to contain numeric [23].

3.5 Generous privileges

Privileges characterized in databases are decides that state which database questions a record approaches and what works the client related with that record are permitted to perform on the items. Common benefits incorporate permitting execution of activities, e.g. SELECT, INSERT, UPDATE, DELETE, DROP, on specific articles. Web applications open database associations utilizing a particular record for getting to the database. An attacker who sidesteps confirmation picks up benefits equivalent to the records. The number of accessible attack strategies and affect objects increments when more benefits are given to the record. The worst scenario is if a record is related with the framework director, which regularly has all privileges [23].

3.6 Dynamic SQL

Dynamic SQL refers to SQL queries powerfully worked by contents or projects into a query string. Commonly, at least one contents and projects contribute and progressively construct the question utilizing user information, for example, names and passwords as qualities in e.g. WHERE statements. The issue with this approach is that query building segments can likewise get SQL catchphrases and control characters, making a totally unexpected question in comparison to the expected [23].

3.7 Client-side-only control

At the point when code that performs input approval is executed in customer side contents just, the security elements of those contents can be superseded utilizing cross-site scripting. This opens for aggressors to sidestep enter approval and send invalidated input key to the server-side [23]. Most system attackers defeat the objective framework with a brute forced traffic to expend all framework resources, (for example, CPU cycles, memory, organize transfer speed, and parcel cushions). These attackers' degrade service and can in the long run prompt a total shutdown. There are two basic kinds of attackers:

(a) Server attackers: these attackers incorporate Smurf IP, TCP SYN, ICMP flood, and Ping of Death attackers. For instance, the attacker may make brute force requests to a casualty server with spoof source IP addresses. Because of TCP/IP convention stack vulnerabilities, the casualty server can't finish the association demands and squanders the greater part of its framework resources. This will come out denial of service on objective attacked system [25].

(b) Routing attacks: the principle system in routing attacks is dispersed denial of service (DDoS) attacks which focuses around switches gadgets. At the point when a switch is endangered, it will forward activity as indicated by the aggressors' goal. Like server attacks, the attacks intend to consume all switch assets, driving the switch to drop every packet, in this way adversely affecting system execution and conduct [25].

4. Literature Survey

Shashank Gupta et.al [01] presented a concept of web application by identify the threads for security purpose. They were summarized a statistics of all web apps vulnerability by referring some vulnerabilities with their classifications. There are CVE, US-CERT, NVD, CWE, OWASP and so on. Then presented weaknesses of web application and discussed how to detect, avoid and mechanism of attack pattern for all web threats which can be critical. They are incorporates 4 aspects of all possibilities such as Attack Vectors of Level Exploitation, Frequency, Affect and Discovery Ability. The normal user of application should go for explicit vulnerabilities. These vulnerabilities are carried on the process of adapting a several changes. Then the users are turned out to susceptible of new vulnerability or weakness was exposed.

Michele Bugliesi et.al [02] presented formal methods for web security. They proposing methods of effective tool design and validate web security solution. They observed that methods applied successfully to different regions of a web security such as browser security, JavaScript security, web protocol and web application security. They discusses important challenges, this should be tackled when web security approaches from formal methods. And they identify the recommendation for investigating a large scale adoption.
Ruilong Deng et.al [03] proposes a FDI Attack against to state estimation in Distribution system. This system state is obtained without paying an expensive cost. They showed that an attacker which can be approximate that the state system is based on injection measurement or power flow

without much more effort. Simulation result of IEEE test feeder which can demonstrate a proposed FDI attack with an approximate state system. This system is more likely to cooperation estimation without detecting bad data detection (BDD) in comparison of traditional attack. They studied a closed loop system when DSSE experimental result is fed back the load model to increase accuracy of pseudo measurement.

Debayan Das et.al [04] proposes a high efficiency of power side channel attack by using noise injection in Attenuated signature field. They propose low hardware modification which attenuated critical Signature of AES provides the attackers are observed from periphery of ASIC encryption. Noise injection is achieved immunity by high efficiency. Successful side channel attack (SCA) immunity was demonstrated for correlation power attack up to 50 traces. The power SCA immunity achieved an effectiveness of power is 73.56 %.

Xuan Liu et.al [05] proposes the strategy of optimal protection against the false injection attack in power system. They formulated an optimal protection for bilevel mixed integer linear programming (MILP) model problem to verify least amount of measurements. Then reduce the complexity of computation also separate the grid power and applied to the strategy of distributed protection to some sub-systems. Then results are performed on IEEE 14 bus, 24 bus, 30 Bus and 118 bus system. These systems were verified an effectiveness and correctness of the protection strategy. This protected strategy is mitigating a risk by considering a No. of measurement attacks is greater than original value. Further they will discover possibilities of obtaining optimal solution to bi-level MILP problem.

Shailendra Singh et.al [06] proposes the performance analysis of vulnerability recognition for web application. Proposed work was done in region of vulnerability identification. But this is not sufficient to identify all vulnerabilities in web apps. They represent the criteria being severity of alert / scan time is produced in various statistics. The system tries to rank the detection of vulnerability scanner. In average, 40% detection of vulnerability occurred. This rate is increasing when provide the situations to scanner, and then increases its rate.

Marco Vieira et.al [07] proposes the method and prototype to calculate the security system of web application. The proposed system injected based on realistic vulnerability in web app. Then attacking automatically is used to support comparison of existing security system. They use the VAIT to run the set of experiments shows that effectiveness and feasibility. The experimental results include an assessment of false positive and coverage of IDS for SQL attack and effectiveness of top commercial vulnerability scanner. The results shows that attacks and vulnerability injection gives effectiveness to assess the security mechanism and point out not only weaknesses also for improvements.

Miss R. V. Bhor et.al [08] proposes a Distributed Vulnerability and Attack Detection Tool for evaluation of security method by automatically injected realistic attack in web app. This tool is analyzing web application and generates possibilities of vulnerability in distributed environment. DOS attack is to make a network resource which is unavailable to its user. This experiment is successfully exploited by inserting SQL vulnerabilities in web app, ping of death attack and HTTP POST. Thus security approaches is existing but it's not providing complete security against web vulnerabilities, there is more scope to improve effectiveness of security methods.

Shashank Gupta et.al [09] proposes the defense mechanisms and cross site script attack. They highlight some vulnerability to found in web application and exposed different vulnerabilities. XSS vulnerability attack is commonly found in today's web application which can be plague for modern apps. This attack permits attacker to run malicious script on web browser which resulting in different side effects are stealing of cookies, credit card numbers, data compromise and passwords etc. they have discussed high level taxonomy of its attack and detail incidence of this attack on web apps.

Ruzhi Xu et.al [10] proposes a competent detection method against FDI Attacks. Initially, 2 parameters are reflecting a physical asset of smart grid is investigated. First one is a control signal from the controller to static-var-compensator. Other parameter is quantitative node voltage stability index. They define 3 levels to cluster nodes into 3 swarms. In clustering progress, they use an improved cluster method for node clustering. This step is uses to find the suspicious FDI nodes easily. The simulation result is built different kind of attack vectors, which gives a sufficient result. Finally, results are demonstrated that the proposed method is detecting effectively FDIA in the smart grid. Further they have a plan to verify the proposed system in large system like IEEE 300 bus and 1354 bus system.

Shashank Guptaa et.al [11] proposes the performance in existing defensive solution of script injection attack. High level of comparison for previous solutions is based on several useful metrics. Proposed automated detection system scans numerous possible locations of web sites for Script injection vulnerabilities. The detection system is firstly scans a web site for learnt injection location. Then, it injects a malicious XSS attack vector in injection point. Finally, it takes an input list of various XSS attacks exploited in second stage and scan the attacks in vulnerable web app. The experimental results are show that proposed system discovers XSS vulnerabilities on BlogIt PHP web app with suitable runtime overhead. In future, they try to improve false negative rate.

Dennis Appelt et.al [12] presents an automated testing system of µ4SQLi, with its underpinning set of mutation operator. This mutation of SQL injection vulnerabilities are supported by tool, can focuses on mutating inputs of web service parameter. µ4SQLi is producing effective inputs which can lead to execute and unsafe of SQL statement. The experimental results demonstrated that proposed method and tool are performed is better than state of practice standard attack pattern. The probability of detection of SQL vulnerabilities gives high, even in a presence of firewall. The number of test cases is executed for single input parameter in single service.

Xiaobing Guo et.al [13] proposes a method by using optimized an attack vector repertory. This method is generated a XSS attack vector automatically, and an optimization model used to decreases the size of attack vectors repertory. They detect the XSS vulnerabilities by XSS attack vector dynamically. Machine learning methods were decreasing a size of attack vector repertory and improved the effectiveness of XSS vulnerability detection. XSS vulnerability detector implemented, for test case on 50 real-world websites. The experimental result is shows that method gives better performance in optimizing attack vectors repertory also detecting XSS vulnerabilities in web apps. The results show that detector is detecting a total 848 of XSS vulnerabilities in 24 websites effectively. Further, they will train optimization model by using more training data and optimizes a parameters of model using more test data. Table 1 depicts the comparison table for performance analysis.

Zainab S. Alwan et.al [14] presents an architecture which is intended for Energy-effective Inter-authoritative with remote sensor information gathering Framework. Ecological screen and urban sensor are two fundamental methods in IoT. Those requirement is raises a huge test to ensure that sensor data collection is gathered in opportune and vitality effective way. Despite the fact that various vitality effective techniques for IoT plans were proposed, existing works assumed finish arrange oversaw by a one relationship in which arrange association and report is pre-designed. The point of system is to allow a dynamic inter organizational collaborative topology towards to spare the vitality from information transmission by an administration arranged architecture.

Ahmed Khalid et.al [15] presents a simple algorithm by using dynamic code investigation tool for the web application to recognize SQL injection vulnerability. The proposed tools rely upon extraction of the suspected POST and GET techniques in the web application and checking the likelihood of infusing SQL helpless proclamations. The proposed technique is tried using famous sites on the web. Experiment is directed to show the execution of proposed apparatus. This tool gives great execution in identifying SQL injection vulnerability.

Beibei L et.al [16] proposed a novel DHCD technique to recognize and alleviate FDI attack in smart grid CPS. In particular, an administer detail based real-time collaborative discovery framework was intended to distinguish the irregularities of estimation data. Keeping in mind the end goal to address these difficulties, a novel circulated have based community oriented identification strategy is proposed. They showed the utility of the proposed method using reproductions of the IEEE 39-bus power scheme.

Benjamin E. Ujcich et.al [17] presents the ATTAIN, an attack injection framework for OpenFlow-based SDN designs. To give attack injection in the SDN setting, they present the ATTAIN system, comprising of an attack language, an attack model and an attack injector. They assessed proposed structure with 2 attacks, and found that various executions caused diverse attack indications in the control and information planes. They expanded unapproved get to and caused a denial of service for true blue information plane activity by interfering with control plane associations.

Chanchala Joshi et.al [18] describes a web application expected to be used to assess the proficiency of Netsparker, Acunetix and Burp Suite web application vulnerability scanners. They explain the guard measures to secure the application essentially. The assessment of 3 noticeable Web Application Vulnerability Scanners is finished by examining the outcomes that is acquired from the execution of web scanners against the helpless web application, at that point looking at the quantity of recognized vulnerabilities. The results of web application assessment recognize the most difficult vulnerabilities for scanner to distinguish, and look at the viability of scanners. The appraisal comes about propose the regions that require additionally research to enhance scanner location rate.

Aleksandar Milenkoski et.al [19] proposes a novel approach for assessment of IDSs in virtualized situations, with an attention on IDSs intended to identify attacks targeting or leveraging on the hypervisor by means of its hypercall interface. They presented hInjector, a device for creating IDS assessment workloads that contain virtualization-particular attacks. They exhibited the use of proposed method and demonstrated its usefulness by assessing a representative IDS intended to identify hypercall attacks.

Swathy Joseph et.al [20] presents an investigation of the popular SQL Injection Attack (SQLIA) strategies and the effectiveness of conventional fixes in lessening them. For tending to the SQLIA's top to bottom, an intensive foundation examine was done and the moderation methods were assessed using both manual and automate testing. They took the assistance of a renowned penetration testing device, SQLMap, for the automated testing. The outcomes show the significance of joining these mitigation systems in the code separated from going for complex fixes that require both exertion and time.

Table 1: Comparison Table for Performance Analysis of Existing methods

S. No	*Paper*	*Year*	*Injection tool of Vulnerabilities / Attacks*	Performance Analysis
1	Deng et.al [03]	2018	False Data Injection Attacks (FDI attacks)	Improves the accuracy of pseudo measurements and performance
2	Das, D et.al [04]	2017	Power Side Channel Attack	Power SCA immunity is achieved with 73.56 % power efficiency
3	Bhor R. V et.al [8]	2016	Distributed Vulnerability and Attack Detection Tool	Successfully exploited the inserted SQL injection vulnerabilities
4	Xu, Ruzhi et.al [10]	2017	False Data Injection Attacks (FDIA)	Proposed mechanism can effectively detect FDIA in smart grid
5	Gupta, S et.al [11]	2016	Cross-Site Scripting (XSS) attacks	Proposed system discovers the XSS vulnerabilities on the BlogIt PHP web application with acceptable runtime overhead

5. Conclusion

Web applications are becomes popular and have wide spread interaction medium in our daily lives. But at same point using vulnerabilities the user sensitive information also disclosed regularly. Web is being used by the organizations for providing their business. So many uses of web applications led to security related test, as the number of applications over the internet increases so the number of attacks is also rises. Hence, security becomes one of the major concerns in web applications.

This paper surveys the area of web application mechanisms, with the existing techniques into a big picture that helps future research. We have discussed major tool injections such as E-mail Header Injection, SQL injection, frame injection, fault injection, code injection, blind injection and timing injection tools. We investigated that some best existing methods of performance analysis (improvement of accuracy, more efficient and power efficiency so on).

References

[1] Gupta, S., & Gupta, B. B, "Detection, avoidance, and attack pattern mechanisms in modern web application vulnerabilities: present and future challenges", Research Gate, International Journal of Cloud Applications and Computing (IJCAC), Vol. *7*, No.3, pp. 1-43, 2017
[2] Bugliesi, M., Calzavara, S., & Focardi, R, "Formal methods for web security", Elsevier, Journal of Logical and Algebraic Methods in Programming, Vol. 87, pp. 110-126, 2017
[3] Deng, Ruilong, Peng Zhuang, and Hao Liang, "False Data Injection Attacks Against State Estimation in Power Distribution Systems", IEEE, Smart Grid, pp 1-1, 2018
[4] Das, D., Maity, S., Nasir, S. B., Ghosh, S., Raychowdhury, A., & Sen, S, "High efficiency power side-channel attack immunity using noise injection in attenuated signature domain", Hardware Oriented Security and Trust (HOST), IEEE, pp. 62-67, 2017
[5] Liu, Xuan, Zhiyi Li, and Zuyi Li, "Optimal protection strategy against false data injection attacks in power systems", IEEE, Smart Grid, Vol. 8, No. 4, pp. 1802-1810, 2017
[6] Singh, S., & Singh, K, "Performance Analysis of Vulnerability Detection Scanners for Web Systems", Springer Singapore, pp. 387-399, 2018
[7] Fonseca, Jose, Marco Vieira, and Henrique Madeira, "Evaluation of web security mechanisms using vulnerability and attack injection", IEEE, Dependable and Secure Computing, Vol. 1, pp. 1, 2014.
[8] Bhor, R. V., & Khanuja, H. K, "Analysis of web application security mechanism and Attack Detection using Vulnerability injection technique", IEEE, In Computing Communication Control and automation (ICCUBEA), pp. 1-6, 2016
[9] Gupta, S., & Gupta, B. B, "Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art", International Journal of System Assurance Engineering and Management, Research gate, Vol. 8, No. 1, pp. 512-530, 2017
[10] Xu, Ruzhi, Rui Wang, Zhitao Guan, Longfei Wu, Jun Wu, and Xiaojiang Du, "Achieving efficient detection against false data injection attacks in smart grid", IEEE, Vol. 5, pp. 13787-13798, 2017.
[11] Gupta, S., & Gupta, B. B, "Automated discovery of javascript code injection attacks in PHP web applications", Elsevier, Vol. 78, pp. 82-87, 2016
[12] Appelt, D., Nguyen, C. D., Briand, L. C., & Alshahwan, N, "Automated testing for SQL injection vulnerabilities: an input mutation approach", ACM, Research Gate, pp. 259-269, 2014
[13] Guo, Xiaobing, Shuyuan Jin, and Yaxing Zhang, "XSS vulnerability detection using optimized attack vector repertory", Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), IEEE, pp. 29-36, 2015.
[14] Halfond, William GJ, and Alessandro Orso, "Detection and prevention of SQL injection attacks", In Malware Detection, Springer, pp. 85-109, 2007.
[15] Khalid, A., & Yousif, M. M, "Dynamic analysis tool for detecting SQL injection", International Journal of Computer Science and Information Security, Vol. 14, No. 2,pp. 224-232, 2016
[16] Li, B., Lu, R., Wang, W., & Choo, K. K. R, "Distributed host-based collaborative detection for false data injection attacks in smart grid cyber-physical system", Journal of Parallel and Distributed Computing, Vol. 103, pp. 32-41, 2017
[17] Ujcich, B. E., Thakore, U., & Sanders, W. H, "ATTAIN: An Attack Injection Framework for Software-Defined Networking", In Dependable Systems and Networks (DSN), IEEE, pp. 567-578, 2017
[18] Joshi, C., & Singh, U. K, "Security Testing and Assessment of Vulnerability Scanners in Quest of Current Information Security Landscape", International Journal of Computer Applications, Vol. 145, No. 2, pp. 1-7, 2016
[19] Milenkoski, A., Payne, B. D., Antunes, N., Vieira, M., Kounev, S., Avritzer, A., & Luft, M, "Evaluation of intrusion detection systems in virtualized environments using attack injection", International Workshop on Recent Advances in Intrusion Detection, Springer, pp. 471- 492, 2015
[20] Joseph, S., & Jevitha, K. P, "Evaluating the Effectiveness of Conventional Fixes for SQL Injection Vulnerability", In Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics, Springer, pp. 417-426, 2016.
[21] Chandramouli, Sai Prashanth, Ziming Zhao, Adam Doupé, and Gail-Joon Ahn, "E-mail Header Injection vulnerabilities", it-Information Technology, Vol. 59, No. 2, pp. 67-72, 2017.
[22] Huang, Yao-Wen, Shih-Kun Huang, Tsung-Po Lin, and Chung-Hung Tsai, "Web application security assessment by fault injection and behavior monitoring", In Proceedings of the 12th international conference on World Wide Web, ACM, pp. 148-159, 2003.
[23] Singh, A. K, "Detection and Prevention of SQL Injection Attack in Web Application", KIIT University Bhubaneswar, 2011
[24] A newspaper for IT Professional, "Code Injection", Issue 4, 2010.
[25] Farraposo, S., Gallon, L., & Owezarski, P, "Network Security and DoS Attacks", 2005.