ISSN No. 0976-5697

Volume 7, No. 1, January-February 2016

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

An Approach to Detect Vulnerabilities in Web-based Applications

Rajdeep Kaur
Dept. of Computer Science & Applications
K.M.V., Jalandhar
Punjab, India

Arshdeep Kaur
Dept. of Computer Science & Applications
K.M.V., Jalandhar
Punjab, India

Er. Gurjot Singh
Asstt. Prof., Dept. of Computer Science & Applications
K.M.V., Jalandhar
Punjab, India

Abstract: The web applications are suppressed of distinctive programming models and interacting technologies that harvests ever-changing web application security. These interactions between distinctive technologies can cause new challenges for the security researchers. In present era, billions of individuals connect to numerous web-based applications for searching information; interact with other individuals for business purposes etc. Some of these crucial web- based services are targeted by the suspicious users that intend to exploit the vulnerabilities, which not only disrupt the services but also the critical information of particular organizations. Thus, securing these web applications against these vulnerabilities is very crucial and challenging. The study of these severe vulnerabilities, detection of attacks and providing solution for these vulnerabilities are the essential part in the internet world. In this paper, we describe the web based application vulnerabilities and implementing distinctive web vulnerability scanners to detect those severe web vulnerabilities in web applications.

Keywords: web application, vulnerability detector, SQL injection, XSS, uniscan, grabber, wapiti

I. INTRODUCTION

With the evolution of information technology, [1]website security is very important because the website contain critical information about a company and now a day's website impairment is very common even a script kiddies and a new born hackers can do this. The most familiar vulnerabilities like SQL-Injection and cross site scripting lead toward the defacement [4]. So to protect the users from the risk of being attacked by any illegal access, it becomes extremely more important to devise new planning and methodologies that will consider the security hole to which the user is prone to. Not only the software developed with faults makes the user vulnerable to attacks, most often network also becomes a key factor by compromising the security aspect of the users. Determine and eliminating the vulnerabilities requires the knowledge and broad understanding of these vulnerabilities. It becomes mandatory enough to know the main idea that works behind these vulnerabilities such as what makes them to show in the system, what faults need to be changed to make the system free from these vulnerabilities, what options can be further prepared for these vulnerabilities so that in future, their risk can be decreased and many more [4, 27]. So you want to secure your web application than find vulnerabilities on it before a hacker find it, use some suitable tools and find vulnerabilities and fix it. There are various vulnerabilities scanners available for both Windows and Linux platform, commercial and open source.

In the following sections, we describe the web applications, their vulnerabilities and the vulnerability scanners with which we can analyze the web based applications for distinctive vulnerabilities. In the analysis section, we implement the scanner to analyze the vulnerabilities and compare them with other present web scanners and in last we conclude the result.

II. WEB APPLICATIONS

A web application is any application that uses a web browser as a client [5]. The application can be as simple as a message or a guest sign-in book on a website, and as complicate as a word processor or a spreadsheet. The benefit of web application is to relieve the web developer their responsibility of building a client for a specific type of system or an operating system [2]. The vulnerable web applications are deployed and are made available to the whole Internet, creating easily-exploitable entry points for compromised all networks.
Web applications have quite common way for companies to conduct business with the outside nature. The static web pages are gone and now, most companies have its own dynamic, interactive web application that connects with their clients. Frequently, these applications are written by programmers not familiarized with security issues behind their Web application front-end or their work is handled by solid to get deadlines so application security is not very high on preference list. Thus these Web applications become major threat to secrecy and integrity of company's sensitive data.

Web applications security events have extremely increasing in past two years. Most circumstance was available to happen due to very well known design flaws in applications. Some of the most familiar faults are Cross Site Scripting (XSS), Injection faults, Malicious File Execution, Insecure Direct Object Reference and Cross Site Request Forgery (CSRF). Last year, top four categories of vulnerabilities, XSS, SQL Injection, Remote File Inclusion and Buffer Overflow, were important for over 50% of all frequent Vulnerabilities and Exposures (CVE's). There are many tools for Web

application security judgment, both open and closed source [3, 27].

III. WEB VULNERABILITIES

Multifarious trust on the input data from the users is the critical aspect of the origin of vulnerabilities in web applications. The developer of distinctive web applications have to be sentient about how user input might interfere with the control flow of the application [7]. These lax enable the malicious hackers to exploit the vulnerability on the page to attack its users. The main focus is to protect the data with regard to confidentiality aspect of security strategy, so that the critical attacks such as Denial of Service attack which only limit to the access temporarily will not be considered further [6].

The following list discloses the most significant attacks that can be injurious to web applications and will be described in further details:

- SQL injection
- XSS(Cross site scripting)
- Cross site request forgery
- Blind SQL
- Code execution attacks
- XXE(Xmle Xtemal Scripting) injection

At present, there exist distinctive web application developing tools, languages and frameworks, mainly; PHP language is most widely used programming language for writing web applications [8]. Like other programming languages, the PHP language does not provide automatic protection against web vulnerabilities thus; the web developers are responsible for checking and sanitizing all input from the user in order to evade these web vulnerabilities. The languages like Ruby and ASP.NET with MVC3 have built-in mechanisms that act as a protecting shield for various vulnerabilities also the advantage for the web developers not to woe about those severe vulnerabilities. If the configuration on the developer system may differ from the testing system in such a way that these mechanisms are disabled then it leads to critical problem for the developers. The distinctive frameworks are present which helps to avoid the vulnerabilities like SQL injections by changing the way to access the data. However, the PHP language is a dynamically typed language so there is no guarantee that variables retain the same type or not. The attacker could submit the strings instead of integers that lead to SQL injection and the web developers may not receptive of the consequence of those dynamically typed languages [10].

Likely all web applications employ a database for storing its information. MySQL is an open source database management system usually used along with PHP applications. The content management system that are written in PHP language has support for MySQL database connections, as it is quite easy to connect to MySQL databases from within a PHP application. Many web hosts facilitates a MySQL database and disk space to the customers for their web application. The WAMP, LAMP, and MAMP (Windows/Linux/Mac, Apache, MySQL, PHP /Perl/Python) are the famous software packages that set up a full operating environment of PHP/ Perl/Python and MySQL with Apache as the web server for desired operation that are effective for web developers to start writing web applications [9].

A. SQL Injection

SQL injection is a most significant application layer attacks currently being engaged on the internet. A SQL injection attacks operates on the particular application and tries to modify the descriptions performed on the database. With this strategy the user input can bypass the limitations of the statement or even change which data is to be retrieved [10]. SQL injection indicates the class of code-injection attacks in which the data served by the user be appended in an SQL query in an efficient way that the user's input query is recognize as SQL code. The cause of SQL injection is quite simple and is incomplete validation of user input. To address this problem, the web developers have scheduled a list of coding strategies that promote defensive coding measures, like encodes the user input and their validation [21].

B. Cross Site Scripting

XSS is an attack across the users of a web application where an infected page is being navigated by them, which lead their critical data vulnerable for the attacker. Basically in XSS attack the malicious JavaScript is injected into a specific page which then is implied in the user's web browser. The JavaScript either sends data, the user's cookie for session hijacking or segments of the visited page, to a site inherited by the attacker by affixing an invisible image to the page with the critical data in the URL, or by changing the target of forms on the page in such a way that the user submits the data is transfer to the attackers site [10, 20].

C. Cross Site Request Forgery

In consolidation with XSS, the Cross Site Request Forgery exploits the faith of a specific end user, where the user unknowingly implies a request to a website that trusts the user.

D. Code Execution Attack

In this, the attacker is able to inject malicious code into the web application and is able to transfer the malicious data that the users were not able to detect it [6]. There are general three methods to perform this type of attack in PHP described as:

a. Reflection- PHP allows dynamic evaluation of code in a particular string using several measures, one is the eval () language structure. If user concerned data is used as an argument for these measures then the site is vulnerable as there are no restrictions on these measures.

b. File Upload- All the PHP files are considered as executable so uploading a file on web server may consider vulnerable and if the file uploaded to a publicly available location and no secure checks are performed on them, then the attacker is able to upload his own script and execute it.

c. ***File Inclusion-*** To evade having all PHP code in one file, another language structure grants loading files when needed to do these tasks. If the user data is used to regulate which files are loaded it might even be possible to consist of files from another domain or files from outside the application [6].

E. XXE (XmleXtemal Scripting)

The XXE vulnerability can be exploited to disclose sensitive information and execute denial of service (DOS)

attacks across the web application server. For illustration this vulnerability can be used to read arbitrary files from the server, along with touchy files such as the application configuration files. Port scanning of the web application and other systems on the clone intranet is also feasible.

F. Blind SQL Injection

In blind SQL injection, it asks the database, true or false queries and resolves the answer based on the applications feedback. This attack is often used when the web application is constructed to display generic error messages, but has not mitigated the code that is exposing to SQL injection. During the evolution process constantly the developer hides some error details which help the attacker to compromise with database [9].

IV. LITERATURE SURVEY

In [22] Zoran Djuric has purposed a tool named SQLIVD that is designed for efficient SQLI vulnerability detection. The specific aim of SQLIVD tool is to generate test inputs & assess those test results. Web application vulnerabilities allow attackers to operate malicious actions from unauthorized CCOUNT ACCESS. In last decade web application vulnerabilities are growing [22].The black box method is based on simulation of SQL attacks against web applications.

In [23] Jose Fonseca, Nuno Seixas, Marco Vieira, and Henrique Madeira have presented an analysis of 715 vulnerabilities, 121 exploits of 17 web applications. A few web applications were written in weak typed language and other in strong typed. According to their paper, weak typed are the performed targets. Most web application has critical bugs affecting their security. To prevent security problems it is important to understand their typical software faults. The SQL Injection & XSS are two more frequently propagate and severe web application vulnerability. To understand how these vulnerabilities are exploited by hacker's their paper presents an analysis of source code of used script.

In [24] José Fonseca and Marco Vieira have studied the use of static code analysis tool to detect vulnerabilities in the plugins. Their result exhibited that many plugins that are currently deployed worldwide have critical cross site scripting & SQL Injection vulnerabilities that can be exploited. Their paper also analyzed the security vulnerabilities of 35 word press plugins using RIPS and PHPSAFE [24, 28]. More than 350 XSS & SOL unknown vulnerabilities were detected plugins are potential source of security problems. Effectiveness of static analysis tools needs to be improved, both in term of coverage and false positive. Many web applications allow the integration of 3rd part server side plugins offer diverse responsibility and open a secondary vulnerabilities door.

In [25] Mukesh Kumar Gupta, M.e. Govil and Girdhari Singh have proposed a classification of software security measures opt to build secure software in various phase of software development life cycle. Static analysis techniques have ability to find out the origin of a security problem and also find its errors [29]. Error finding not only abate the cost of error but also quick feedback cycle efficiently improves the coding approach. Static analysis approach suffers from false positive and false negative outcomes. Dependence on web application is increasing very rapidly in recent decade and devise problem and for other purposes. Sql Injection and XSS are the most threatening security vulnerability exploited in web application i.e ebay, google, fb, etc. Most developer's repeat same programming mistake in their code because they do not follow security guidelines.

In [26] Ibéria Medeiros and Nuno Neves have presented an approach for finding and correcting vulnerabilities in web applications and a tool to imply those specific approaches for PHP programs. Static source code analysis and data mining two techniques are opt for that purpose. The security of web application continues to be a challenging task. In their paper, the combination of measures is opting to discover vulnerabilities in source code with fewer false positives. WAP tools are basically opting and large set of PHP applications are used for experimental evaluation. They found 388 vulnerabilities in 1.4 million lines of code.

V. WEB VULNERABILITY DETECTORS

The web vulnerability scanners are opted to find distinctive vulnerabilities in web applications [30]. Following are the details of various web vulnerability detectors:

A. Wapiti

Wapiti is an open source web application vulnerability detector that enables to analyze the security of the web applications. It executes "black-box" scans, i.e. it does not examine the source code of the web application but will investigates the web pages of the set up web application, target for scripts and forms where to inject the data [11, 12]. Once it gets this list, it behaves like fuzzier, injecting payloads to see if a script is unprotected. Wapiti can disclose the following distinctive vulnerabilities:

1. File disclosure (remote and local include/require, fopen, readfile)
2. Database Injection (PHP/JSP/ASP SQL Injections and XPath Injections)
3. XSS (Cross Site Scripting) injection i.e. reflected and permanent
4. Command Execution detection (eval(), system(), passtru())
5. XXE (XmleXternal Entity) injection
6. Use of known potential risk files (Nikto database) [12].

B. W3af

W3af (web application attack and audit framework) is an open-source web application security detector and exploitation tool. This cross-platform security tool is applicable in all popular operating systems such as Microsoft Windows, Linux, Mac OS X, FreeBSD and Open BSD and is drafted in the Python programming language. The users have the choose ability between a graphic user interface and a command-line interface to operate that tool. It recognizes stringent web application vulnerabilities with more than 130 plugins. After description, vulnerabilities like blind SQL injections, OS commanding, remote file inclusions (PHP), cross-site scripting, and unsafe file uploads, can be opt in order to achieve distinctive access to the remote system. It is cleft into two main divisions, the core and the plug-ins [11].The core

coordinates the procedure and provides features that are absorbed by the plug-ins, which detect the vulnerabilities and exploit them. The plug-ins is connected and share information with each other using a knowledge base.

C. Nikto

Nikto is an effective web application vulnerability scanner that is designed to analyze a web server to find severe security issues or flaws. Identifying security problems proactively, and fixing them, is a crucial step towards verifying the security of the web server [1]. Running nikto on a regular basis will verify that you identify common problems in your web server or web applications. Nikto is completely open source and is written in Perl. It is a non-invasive scanner can runs at the command line, without any graphical user interface. It is ideal for running the tool remotely over SSH connections [1, 13]. If you have Web servers allocate CGI scripts, Nikto can be an efficient resource for auditing the security of these web servers and applications [13].

It supports SSL, full http proxy, text, HTML, XML and CSV to save reports. It can scan for multiple ports and server by taking inputs from files like nmap output. It is capable enough to identify installed software with headers, files and favicons. It can authenticate hosts with Basic and NTLM and Scans can be Auto-paused at specified time. It can be integrated in Nessus and is automatically configured to launch Nikto when it finds a particular web server. It can capable of sending data along with requests to servers (cross site scripting and SQL injection) [31]. It can scan tuning to include or exclude entire classes of vulnerability checks and also guess sensitive information for authorization realms that includes many default id/pw combos. Authorization guessing shafts any directory, not just the root directory. It can enhance the false positive reduction via multiple measures like headers, page content, and content hashing [31].

Following are the list of Vulnerabilities scan by nikto scanner

1. Server and software miss configuration
2. Default files and programs
3. Insecure files and programs
4. Outdated servers and programs [13, 14]

D. Burp-Suite

Burp Suite is one of the terrific tools that accessible for web application testing. It has extensive features that help us to perform various jobs, from intercepting a request and modifying it on the fly, to scanning a web application for vulnerabilities, to brute forcing login forms, to execute a check for the randomness of session tokens and many other functions. Some of the features that are not available in the free edition are Burp Scanner, Task Scheduler, Target Analyzer, etc. Some common features of Burp-Suite that comes with a proxy, which runs on port 8080 by default [1, 15]. The features of burp-suite includes the automate detection of numerous forms of vulnerabilities. It is an intruder tool for implementing dominant attacks to find and exploit unusual vulnerabilities. It is an application-aware spider for crawling content and functionality. It is a repeater as well as sequencer tool, for manipulating and resending individual requests and also for testing the randomness of session tokens. It also has a feature of Extensibility that easily allows you to customize your own plugins, to perform complex and highly customized tasks within Burp [1, 15].

E. Uniscan

It is an efficient Web vulnerability detector that focused at information security to find stringent vulnerabilities in Web systems and is authorized under the GNU GENERAL PUBLIC LICENSE 3.0. It can be build by opting Perl language. This tool can be used to test the given below vulnerabilities [16].

1. Remote file include (RF)
2. Local file include (LFI)
3. Remote command execution (RCE)
4. Cross-site scripting (XSS)
5. SQL and Blind SQL injection

a. The characteristics of uniscan scanner are as:

1. Identification of system pages through a Web Crawler and use of threads in the crawler.
2. Control the maximum number of requests and variation of system pages identified by Web Crawler.
3. Control of file extensions that are ignored.
4. Test of pages found via the GET and POST method.
5. Support for SSL requests (HTTPS) and proxy.
6. Generate site list using Google and bing.
7. Plug-in support for Crawler, dynamic tests, static tests and for stress tests [16].

The main advantages of uniscan web scanner are its quick installation and starting up of several sensors that are easily connected through flat cables and an additional remote control enable sensors. The Y-Adaptor offers a simple cabling solution with only one cable for two sensors on both sides of the door. The two door loop sets DLP 6 and Door Loop provide a clean connection to the wall [17]. Its testing method that can be selected in accordance with the respective drive types with no interference through synchronization of the optics and we can individually turning off of single beams.

F. Grabber

It is an open source web application detector that helps in detecting security loopholes in web applications. It performs scans to detect the vulnerabilities like cross site scripting, SQL injection, Ajax testing, backup file check and more. It's a very simple and portable scanner which is mainly used for small web applications and it also doesn't offer any GUI interface. It can knob the JavaScript files, parse it to retrieve the server side's scripts names and tries to grab some specific parameters name. This tool is mainly helpful for personal use and not any kind of professional purpose. It is written in Python, and its source code is also available which can be modified as per requirements [18, 19].

Its features includes Cross-Site Scripting, a special Blind SQL Injection module, File Inclusion and Backup files check, Simple AJAX check It can parse each JavaScript and grab the URL and try to obtain the parameters. *It also has* Hybrid analysis/Crystal ball testing for PHP application using PHP-SAT, JavaScript source code analyzer, Generation of a file [session_id, time (t)] for next stats analysis [19].

VI. ANALYSIS AND DISCUSSION

In this section, we implement Nikto, Uniscan and Grabber, a web vulnerability detection tools on linux operating system

that detect numerous web vulnerabilities such as Sql injection, XSS attack, Remote code execution, blind sql, local file include, file insertion attack and javascript code test in web based applications. The following table 1 compares the different web scanners for the distinctive vulnerabilities. As seen from the table 1, Grabber and Uniscan vulnerability scanner that has detected most of the specific vulnerabilities followed by Nikto scanner. In [11] various vulnerability scanners like nmap, nessus, acunetix wvs, nikto and burp suite are opted to detect the various web based vulnerabilities.

Table1. Comparative view of the vulnerabilities detected by the scanners

Vulnerabilities	*Nikto*	*grabber*	*uniscan*
Sql injection		✓	✓
Cross site scripting (XSS)	✓	✓	✓
Remote code execution			✓
Rogue server	✓		
Blind sql injection		✓	✓
Local file include			✓
File insert include		✓	
Javascript code test		✓	

VII. CONCLUSION

Web applications have the most imperative way to provide access to online services. A most thriving obstacle is client-side attacks in which suspicious software is automatically installed on the individual's machine. In this paper, we analyze various web based vulnerabilities through web vulnerability scanners. Vulnerability assessment plays a significant role in securing the network system. Our experimentation on web vulnerabilities show that different web scanners detect different type of vulnerabilities but not a single tool is as efficient to detecting all types of web vulnerabilities.

This paper addressed various tools opted for scanning web vulnerabilities and their comparative analysis. We identified that what vulnerabilities a specific tool is efficient to detect by running various web applications on each tool.

Grabber and uniscan tool has many significant features including within it that are not present in other tools i.e. nikto, wapiti and nessus. The vulnerabilities such as file insertion attack, local file include and blind sql injection are not present in other tools and hence can be indulged with the other tool that works differently and produces distinctive results.

REFERENCES

[1] Sheetal Bairwa, Bhawna Mewara and Jyoti Gajrani, "Vulnerability scanners: a proactive approach to assess web application security", International Journal on Computational Sciences & Applications, Vol.4, No.1, February 2014.

[2] Davide Balzarotti, Marco Cova, Viktoria V. Felmetsger, and Giovanni Vigna' "Computer Security Group University of California, Santa Barbara Santa Barbara", CA, USA, Multi-Module Vulnerability Analysis of Web-based Applications

[3] G. Vigna, W. Robertson, V. Kher, and R.A. Kemmerer, "A Stateful Intrusion Detection System for World-Wide Web Servers", In Proceedings of the Annual Computer Security Applications Conference (ACSAC 2003), pages 34–43, December 2003.

[4] http://www.ehacking.net/2011/08/top-6-web vulnerability-scanner-tool.html.

[5]http://webtrends.about.com/od/webapplications/a/web_application.htm

[6] C. Kruegel and G. Vigna, "Anomaly Detection of Web-based Attacks", In Proceedings of the ACM Conference on Computer and Communication Security, pages 251–261, October 2003.

[7] D. Scott and R. Sharp, "A bstracting Application-Level Web Security", In Proceedings of the International World Wide Web Conference, pages 396–407, May 2002.

[8] René Rydhof Hansen and Mads Christian Olesen, "A Study of Web Application Vulnerabilities and Vulnerability Detection Tools", Aalborg University Department of Computer Science, January 5th, 2012.

[9] Shakti Kumar, Subhendu Dey , R.Karthikeyan and K.G.S. Venkatesan, "Prevention of SQL Injection Attack on Web Applications", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 3, March 2015

[10] Gary Wassermann Zhendong Su, "Sound and Precise Analysis of Web Applications for Injection Vulnerabilities", University of California, Davis, ACM 978-1-59593-633-2/07/0006, 2007.

[11] Rim Akrout, Eric Alata, Mohamed Kaaniche and Vincent Nicomette,"An automated black box approach for web vulnerability identification and attack scenario generation", Journal of the Brazilian Computer Society, 2014.

[12]A. Dessiatnikoff, R. Akrout, E. Alata, M. kaaniche and V. Niconette, "A Clustering Approach for Web Vulnerabilities Detection", 17th IEEE Pasific Rim International symposium on dependable computing, pp.194-203, 2011.

[13] Saleh M., AI-Saleem, "A Critical Survey of different Security aspects in Saudi Arabian Web Servers", International Journal of Computer Science and Network Security, VOL.15 No.2, February 2015.

[14] Yuji, Kosuga, Kenji Kono, Miyuki Hanaoka, Miho Hishiyama and Yu Takahama, "Sania: Syntactic and Semantic Analysis for Automated Testing against Sql Injection", IEEE computer security application conference, Miami Beach, pp.107-117, 2007.

[15] James Walden, "Integrating Web Application Security into the IT Curriculum", ACM, New York, USA, pp.187-192, 2008.

[16]Rocha, D., Kreutz, D. and Turchetti, R., "A free and extensible tool to detect vulnerabilities in web systems", IEEE (information systems and technologies), 7th lberian conference 2012, pp.1-6

[17]http://reglomat.bircher.com/fileadmin/_migrated/content_ uploads /UniScan_EN_03.pdf

[18] M. Almgren, H. Debar and M. Dacier., "A Lightweight Tool for Detecting Web Server Attacks", In Proceedings of the Network and Distributed System Security Symposium, pages 157–170, February 2000.

[19]Karen Mercedes Goertzel and Theodore Winograd, "Information Assurance Tools Report-Vulnerability Assessment", IATAC, Herndon, 6th edition May 2 2011.

[20]Stefan Kals, Engin Kirda, Christopher kruegel and Nenad jovanovic, "SecuBat: A Web Vulnerability Scanner", 15th International Conference on World Wide Web, ACM New York, USA, pp.247-256, 2006.

[21] William G.J. Halfond, Jeremy Viegas and Alessandro Orso, "A Classification of SQL Injection Attacks and Countermeasures", College of Computing Georgia Institute of Technology, IEEE 2006.

[22] Zoran Djuric, "A Black-box Testing Tool for Detecting SQL Injection Vulnerabilities", ISBN: 978-1-4673-5256-7/13/$31.00, 2013 IEEE.

[23] Jose, Fonseca, Nuno Seixas, Marco Vieira, and Henrique Madeira, "Analysis of Field Data on Web Security Vulnerabilities", IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 11, NO. 2, MARCH/APRIL 2014.

[24] José, Fonseca and Marco Vieira, "A Practical Experience on the Impact of Plugins in Web Security", IEEE 33rd International Symposium on Reliable Distributed Systems, 2014.

[25] Mukesh Kumar Gupta, M.e. Govil and Girdhari, "Static analysis approach to detect SOL Injection & Cross Site Scripting vulnerabilities in web application", IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), May 09-11, 2014, Jaipur, India.

[26] Ibéria Medeiros, Nuno Neves, "Detecting and Removing Web application vulnerability with static analysis and data mining", IEEE, Transaction on reliability, 2015.

[27] Dali, Loubna, Ahmed Bentajer, Elmoutaoukkil Abdelmajid, Karim Abouelmehdi, Hoda Elsayed, Eladnani Fatiha, and Benihssane Abderahim. "A survey of intrusion detection system", 2015 2nd World Symposium on Web Applications and Networking (WSWAN), 2015.

[28] Hazel, J.Jemi, P. Valarmathie, and R. Saravanan. "Guarding web application with multi - Angled attack detection", 2015 International Conference on Soft-Computing and Networks Security (ICSNS), 2015.

[29] Gupta, Mukesh Kumar, Mahesh Chand Govil, and Girdhari Singh. "An approach to minimize false positive in SQLI vulnerabilities detection techniques through data mining", 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014), 2014.

[30] Marco Cova. "Vulnerability Analysis of Web-based Applications", Test and Analysis of Web Services, 2007.

[31] Scholte, T., "Have things changed now? An empirical study on input validation vulnerabilities in web applications", Computers & Security, 05, 2012.

