International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

# Web Application Security Scanner for Prevention and Protection against Vulnerabilities

**Binny George[1], Jenu Maria Scaria[1], Jobin B[1], Praseetha VM[2]**

*[1]UG Student, Dept. of CSE, SJCET Palai, Kerala, India*
*[2]Associate Professor, Dept. of CSE, SJCET Palai, Kerala, India*

---

**Abstract -** *In today's world, Cyber security has become an important leap in the form of jobs, education. But the reality is that only a few are aware of the major web vulnerabilities. Some statistical studies show that small scale industries are directly and indirectly connected to the world of the internet, but they are not aware of the major web vulnerabilities of their web application. Since website hosting has become common nowadays, most of the web applications are prone to attacks and malicious attacks of web applications. Assessing and avoiding these vulnerabilities require deep knowledge of these vulnerabilities. There are numerous online scanners available on the Internet that provides only paid limited service. The tools are made in a way that it can only operate in command line interface or in any programming language. So it is a difficult task for a normal person to operate the scanners without previous knowledge. This paper presents a vulnerability scanner that scans the website and detects specific vulnerabilities, along with its location and solution. The vulnerabilities that the scanner considers are: SQL injection, cross site scripting.*

## 1. INTRODUCTION

Web applications have become an integral part of everyday life, but many of these applications are associated with vulnerabilities. In this era, where website hosting has become cheap and easy, the security has failed to keep up. Such vulnerabilities can risk small scale to large scale industries. Exploitation of vulnerability by an unauthorized person demands for quick recovery of these flaws so that reputation of the organization can be recovered. Therefore, vulnerability scanners can be widely used to evaluate the known weakness and vulnerabilities in a website. A large number of applications are becoming online, but how secure these applications are a matter of concern. Thus, it becomes necessary to find vulnerabilities that may cause severe risk to user's security. Vulnerability assessment means detecting the vulnerabilities before they could be used by an attacker. It is not only performed on a particular application but it can be run on any platform on which the application is run. This strategy only takes into consideration all the factors that can provide the correct answer for assessment of the vulnerability and security of the system. Therefore vulnerability scanners are used to scan the network and software application.

### 1.1 Types of Scanning

Scanning can be of two types [1], passive scanning and active scanning. In passive scanning, it is determined whether the tool can list out the vulnerabilities by considering the existing network. In active scanning, it is determined whether the queries can be made to the network for vulnerability. Scanners can be of different categories like port scanners, application scanners and vulnerability scanners [2]. Port scanners can be used to scan the ports for determining the open and closed ports, operating system, services offered. Application scanners are used to access a specific application in the network in order to track the weakness that can further cause a risk to the system. Vulnerability scanners find out the vulnerabilities in the system which when attacked by an attacker can exploit the system.

### 1.2 Mechanics of a scanner:

Mechanics of a scanner [3] is a three step process, crawling, simulation of attacks (fuzzing) and response analysis. In the first step, the scanner crawls into the web application that is part of the application and associated input pages and makes an index of all the visited pages. If the crawling mode is poor and the scanner has not reached the vulnerability, the scanner will surely miss the vulnerability. In the fuzzing step, the scanner sends some attacking patterns to the previously identified inputs. For each input and each vulnerability for which the scanner tests, the attacker module generates values that trigger vulnerability. In the response analysis phase the result of fuzzing phase is monitored to check if the web application is vulnerable and provide feedback to other modules. In the past, many of the popular websites have been hacked. Attackers are very active now and exploit the data without the user's knowledge. That is why security of web applications has become important these days. Web application security scanning is a software program which performs testing of web applications and identifies security vulnerabilities. Scanner does not scan the source code, but they perform only detection of the vulnerabilities [3]. Vulnerability management has many components like identifying the

vulnerabilities, evaluating risks, addressing risks and report security gaps. Identification of vulnerabilities involves the admins to identify the security holes in the network. It captures vulnerabilities as possible. While many of the top companies have got a special sector to identify the flaws in the network, automated tools have made these efforts more time saving. In the next phase, the risks are being evaluated. Not all vulnerabilities are crucial and urgent. Scanning tools identify the vulnerabilities and classify and categorize them that help to prioritize the vulnerabilities. Once the risks have been identified, the next task is to address these risks. The right tool helps you to automate the process of provisioning the devices. Once the vulnerabilities have been addressed, the scanning software can facilitate the creation of reports about whether a system is secure.

## 2. EXISTING SOLUTIONS

For testing and evaluating vulnerability using scanners, a vulnerable web environment has to be formulated. This is fulfilled by vulnerable web applications that are specially designed to provide users, the environment to identify the attacks and the way to rectify it. This section deals with some of the scanners that can evaluate the vulnerabilities of a web application [4].

### 2.1 Nmap

Nmap [2] is a port scanner that is used to scan the ports. It takes an IP address and captures all the information related to it. If an IP address is provided, then it finds the host to which it belongs to. It also finds the number of ports that are running on that particular host, number of ports that are opened, number of closed ports, services provided by these ports, which may be TCP or FTP oriented. It predicts the type of operating system that is being connected to the system. The topology of the host is recorded in the form of graphical format which shows the various gateways through which the local machine accesses that host. If the ports are opened, then the attacker can easily make unauthorized access to the host. A number of various ports can be scanned using Nmap.

### 2.2 Nessus

Nessus [2] is a vulnerability scanner that lists out the vulnerabilities in the remote host. It provides both internal and external scan. Internal scan is related to the host within a particular router. External scan involves the host outside a particular router. Web application tests can also be performed using the scanner. There are two ways in which scanning can be performed, either it can be done at first instance or a template can be formulated for a host and launch this to scan the host. Multiple scanning of the host can be done at once. Vulnerability can be evaluated by Nessus using four types of severity-high, medium, low and informal. Results are saved automatically as soon as the scan of the particular host is completed. The results are provided in two ways-vulnerabilities by plug-in and vulnerabilities by host. The first one classifies all the vulnerabilities during the scan and lists out the outs affected by these vulnerabilities. It generates a report that can be used to fix the vulnerability. The latter addresses the issues related to host, follow up scans and assessment is done accordingly. The results can be exported in any desired format. Nessus is based on client-server architecture. Each session is controlled by the client and the test is done on the server side. More than 100 websites can be scanned using Nessus.

### 2.3 Acunetix

Acunetix [4] is an automated web application security testing tool that checks the web application by checking for web vulnerabilities like SQL injection, cross-site scripting and exploited vulnerabilities. Acunetix scans a web site or web application that is accessible via a web browser. Acunetix offers a strong solution for custom based web applications utilising Javascript, AJAX. Acunetix has an advanced crawler that finds any file. The scanning is performed in three steps-target specification, site crawling and structure mapping and pattern analysis. In target identification, the target is checked with an active web server and hosts any web application. Information is collected regarding web technologies, web server type and responsiveness for appropriate filtering tests. In structure mapping and site crawling, the index file of a web application is fetched first, determined by the URL. Received responses are taken to get inks, input fields that create a list of directories and files inside the web application. Pattern analysis is executed against the web application.

### 2.4 Nikto

Nikto [5] is a command based tool that is used to scan specific targets. It uses Perl language to scan the web application. It performs security checks against dangerous files. Attackers look for web application vulnerabilities so that they can gain

access to outdated apache servers. It is a free and open source scanning tool therefore IT enterprises can easily identify the security flaws in the organization and take necessary steps to shield and upgrade the system. The tool is able to find servers that were not developed by the enterprise.

### 2.5 Burp Suite

Burp scanner [6] is a tool for automatically finding vulnerabilities in a web application. It is designed to be used by security testers. It is a proxy based tool package. It consists of various functional specifications. The tools offered by burp suite are spider, proxy, intruder, repeater, sequencer, decoder, extender and scanner. In the first step, a proxy is set in the browser. After the proxy is set, the burp suite is about to begin. The burp window [7] has many tab specifications like proxy, intruder, spider, repeater, sequencer, and scanner, where each has got its own sub-tabs. For instance, proxy tab has three sub tabs -intercept, proxy and options. Proxy tab is used to fix the proxy and configure it. At this time, the intercept tab remains with it. A xampp server is installed in the system which is developed to scan the system. Through this, the username and password of the user can be identified, provided that the intercept tab remains off. Intruder tab is used to automate customized attacks against web applications. Spider tab provides the crawler feature for the application test. Repeater tab is used to modify the HTTP request and analyze their responses. Scanning performs scanning of the hosts. Scanning involves testing of hosts for the vulnerabilities inside it. Burp suite identifies the type of vulnerability and its severity.

**Table -1:** Comparison of Scanners

| Vulnerabilities | Nmap | Nessus | Acunetix | Nikto | Burp Suite |
|---|---|---|---|---|---|
| SQL Injection | ✓ | ✓ | ✓ | | ✓ |
| Improper Error Management | ✓ | ✓ | ✓ | | ✓ |
| Cross Site Scripting | ✓ | ✓ | ✓ | ✓ | ✓ |
| Insecure Cookies | | | ✓ | | ✓ |
| Session Token in URL | | | ✓ | | ✓ |
| Password Auto-Enabled | | | | | ✓ |

Table -1 shows comparative view of tools mentioned above on the basis of the vulnerabilities they detect. From the comparative analysis of various scanners it is clear and evident that scanners perform differently in different categories. They may vary differently in a number of aspects like the type of vulnerabilities they evaluate, number of vulnerabilities, the speed with which they detect the vulnerabilities, cost effectiveness and efficiency. However, being cost effective and efficient is not enough, they should be secure, reliable and identify the vulnerabilities accurately. Most of the online scanners are paid and provide only limited services. The complex steps may not be easily understood by small scale users. In addition to this, two of the most widely spread and dangerous vulnerabilities are SQL injection and cross site scripting. Exploitation of these vulnerabilities can risk the web application environment.

## 3. PROPOSED METHOD

With the increasing development of the internet, web applications have become increasingly vulnerable and are exposed to unauthorized attacks. To deal with this problem, many online scanners are available in the market. But most of them are not able to detect all the vulnerabilities available. If a situation arrives where the scanner we use cannot detect the vulnerability, then the attacker can easily crawl into the system and exploit the data and resources. Our proposed method is a vulnerability scanner which detects the vulnerabilities like SQL injection, cross site scripting, broken authentication [8], payload, email disclosure. The vulnerability scanner scans the website and checks whether the above vulnerabilities are identified while scanning. Fig-1 shows the overall design of the vulnerability scanner. The scanner is available as either mobile or web application. The user can submit the URL in the application where the scanner will crawl into the URL to check for the sub-URLs. The scanner then identifies if the above mentioned vulnerabilities are present in the URL or not. If

present, it will list out the vulnerabilities and some other additional information regarding the URL like sever information, technology information, certification information, etc.

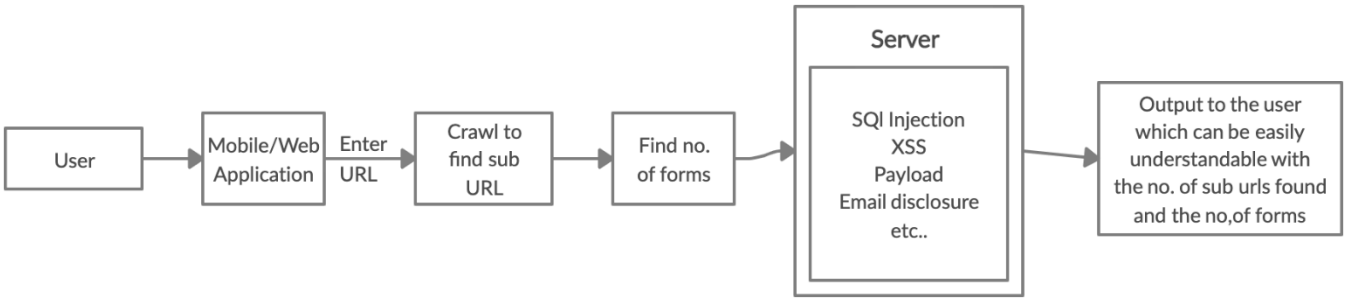


**Fig-1:** Overall Design

### 3.1 SQL Injection

User gives his username and password through a web application. Web application has stored the given details to the SQL server. An attacker gives HTTP requests that are sent to the web server to inject commands to the SQL server in order to gain system level access. The vulnerable web application allows this malicious code to be placed on an SQL server, thus making it possible for the attacker to use SQL commands to get user account credentials.

How SQLI vulnerability can be exploited?

During an SQLI attack, a malicious code is given as an input to a function that calls an SQL query, which is called immediately. Injection allows the user to add malicious code via a web application to another system. These attacks include calls to an operating system through system calls, the use of external programs via shell commands, and also calls to backend databases. Injection is a weak point where the user can insert a malicious code of his interest. Malicious code is embedded with the user input data and passed to the application. If user input data is not properly filtered in the system, in that case, the interpreter processes the malicious code as a normal legitimate user input and the system outputs accordingly. The level of risk associated with SQL injection is high. An attacker can exploit the data and steal necessary information from the web application. Fig-2 shows some of the SQL errors that the vulnerability scanner considers while scanning.

```
sql_errors = {
    "MySQL": (r"SQL syntax.*MySQL", r"Warning.*mysql_.*", r"MySQL Query fail.*",
        r"SQL syntax.*MariaDB server"),
    "PostgreSQL": (r"PostgreSQL.*ERROR", r"Warning.*\Wpg_.*",
        r"Warning.*PostgreSQL"),
    "Microsoft SQL Server": (r"OLE DB.* SQL Server", r"(\W|\A)SQL Server.*Driver",
        r"Warning.*odbc_.*", r"Warning.*mssql_", r"Msg \d+, Level \d+, State \d+",
        r"Unclosed quotation mark after the character string", r"Microsoft OLE DB
        Provider for ODBC Drivers"),
    "Microsoft Access": (r"Microsoft Access Driver", r"Access Database Engine",
        r"Microsoft JET Database Engine", r".*Syntax error.*query expression"),
    "Oracle": (r"\bORA-[0-9][0-9][0-9][0-9]", r"Oracle error", r"Warning.*oci_.*",
        "Microsoft OLE DB Provider for Oracle"),
    "IBM DB2": (r"CLI Driver.*DB2", r"DB2 SQL error"),
    "SQLite": (r"SQLite/JDBCDriver", r"System.Data.SQLite.SQLiteException"),
    "Informix": (r"Warning.*ibase_.*", r"com.informix.jdbc"),
    "Sybase": (r"Warning.*sybase.*", r"Sybase message")
}
```

**Fig -2**: Output of SQLI

### 3.2 Cross Site Scripting

Cross site scripting (XSS) vulnerability occurs when there is a possibility of inserting a malicious code by an unauthorized user. Thus, the XSS flaw is as a result of not validated or sanitized input parameters. There are three types of XSS: Non-Persistent, called Reflected XSS; Persistent or Stored XSS; and Document Object Model (DOM)-based. Non-persistent XSS

vulnerability occurs when a web application accepts the user's malicious request. This is then echoed to the application's response in an unsafe way. Persistent XSS vulnerability occurs when a web application accepts the malicious request, stores in a data source and which then displays the information from the request to a wide range of users. DOM-based XSS vulnerability does not involve server validation. The attack works on a web browser, avoiding the server side. The DOM environment in the victims browser is modified by the original client-side script, and as a result of that, the payload is executed. In Cross-Site Scripting, attackers exploit the user's trust over a vulnerable web application. In this attack, malicious JavaScript or html codes are inserted through user input fields to a page. It generally occurs when the application sends user input data as a part of a webpage, without properly validating to the user's browser. The risks associated with Cross-Site Scripting include a hijacking session, an unauthorized changing of the contents of application, redirecting the application to another website, and insertion of some malicious codes or links. The level of risks is high. By the hijacking session, attackers can get secret and important information.

### 3.3 Broken Authentication

The user authentication typically involves the username and password of the user. When the authentication process weakens, the attacker can get the credentials of the user. Authentication is an act of verifying the identity of a user, allowing access to resources in an information system. It refers to the process of verifying either a user, which requests to communicate with either the whole application or with a part of it in order to make sure that only the intended user can get access to the application and its resources. A session is a sequence of all the activities between a client and a web server for a particular login and logout period. The activities are generally associated within the login and log out period of the same user. So, there is a different session for each different user. For a number of different reasons, a web application requires to hold session information. For example, there may be different contents to deal with according to the preference, or the type of user; or there may be security issues. Effective authentication and proper session management are important for a web application to be secure. The level of risk associated with the authentication and session management is high. Attackers usually gain access to the system through hijacking a username and a password or session IDs. They can access secret information and data while pretending that they are the legitimate user of the application.

How Broken Authentication is exploited?

The password recovery mechanism is based on a secret question and answer [10]. A user provides the name of the city, when he/she was born and his/her password is immediately displayed on a web page without further verifications. Using social engineering, an attacker can guess the credentials of the user. Brute force attack is widely used to obtain log-in credentials, session identifiers, and credit card information with the help of brute force tools. Attackers can use these tools and proxy applications such as Burp Suite to access a user's private information. Brute force attack is very simple
1. The intercepted request is sent to the intruder application.
2. The parameter is selected
3. Payloads are formed and configured to be used in the task.
4. Attack begins.

Additional information we get while scanning are,
- Who is
- SQL
- XSS
- Crawl
- E-mail disclosure
- Credit Card disclosure
- Command integration get method
- Directory Traversal
- Server information
- Technology information
- X-content type check
- X-SSS protection check
- Tep-port Scanner
- URL encode
- Certification Information
- Available methods
- Cyber thread indicate
- IP2 location
- File input available check

## 4. CONCLUSION

The results of our comparative evaluation of the scanners confirmed again that scanners perform differently in different categories. Therefore, no scanner can be considered an all- rounder in scanning web vulnerabilities. The above proposed scanner is best suited for beginners who are not aware of the complex steps of scanning. Vulnerability scanning identifies the security vulnerabilities in an organization. Vulnerability assessment provides the organization with the awareness and risk associated with the organizations working environment and work accordingly. The advantage of using vulnerability scanner is that it identifies known security exposures before attackers find them.

## REFERENCES

[1] Mansour Alsaleh, Noura Alomar, Monirah Alshreef, Abdulrahman Alari and AbdulMalik Al-Salman, "Performance-Based Comparative Assessment of Open Source Web Vulnerability Scanners", (2017) .

[2] Sheetal Bairwa, Bhawna Mewara and Jyoti Gajrani, "Vulnerability Scanners: A Proactive Approach to Assess Web Application Security", (2014)

[3] S. El Idrissi, N. Berbiche, F. Guerouate and M. Sbihi, "Performance Evaluation of Web Application Security Scanners for Prevention and Protection against Vulnerabilities", (2017).

[4] Acunetix https://www.acunetix.com/vulnerability-scanner/

[5] Nikto https://hackertarget.com/nikto-website-scanner/

[6] Burp Suite https://portswigger.net/burp

[7] YU Shiyuan, WANG Yutian, LIU Xin, "Burp Suite Extender Apply in Vulnerability Scanning", (2018).

[8] Balume Mburano, "Evaluation of web vulnerability based on OWASP Benchmark", (2017).

[9] Deepika Sagar, Sahil Kukreja, Jwngfu Brahma, Shobha Tyagi, Prateek Jain, "Studying Open Source Vulnerability Scanners For Vulnerabilities In Web Applications", (2017).

[10] Kinnaird McQuade, "Open Source Web Vulnerability Scanners", (2014).

[11] Y. Makino and V. Klyuev, "Evaluation of web vulnerability scanners," IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), (2015).