Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Volume 13, Issue 03 March 2024

Enhancing Web Application Security: A Comprehensive Approach with WVS (Web Vulnerability Scanner)

Punyaban Patel, Reddyvari Venkateswara Reddy, Devalla Sai Kiran, J. Sai Sree Harsha, A. Mahi Pavan Reddy
Professor, Department of CSE(Cybersecurity), CMR College Of Engineering & Technology, Hyderabad, Telangana, India
Associate Professor, Department of CSE(Cybersecurity), CMR College Of Engineering & Technology, Hyderabad, Telangana, India B. Tech Student, Department of CSE(Cybersecurity), CMR College Of Engineering & Technology, Hyderabad, Telangana, India

***Abstract*— Nowadays, the Internet's rapid development has led to the increasing popularity of web applications. Security remains among the major challenges for the software industry continue replacing the corresponding client applications. Because of increasing prevalence and the expanded influence of attackers, web applications are likely to become an obvious choice of attackers who have to quickly identify the vulnerabilities. For those reasons, it may be critical to pinpoint the problem quickly and accurately benefits of modern automated vulnerability scanning systems, which work better than the manual detection method used earlier. This paper aims at common vulnerability screening method for application securing in the software industry. The paper starts by presenting the working principle of the vulnerability detection tools. It then discusses vulnerability scanning technology and presents the implementation process of the technology additionally the functions of its key module. Finally, it displays the scanning results and verifying the technology study. Finally, it highlights the key functions of the vulnerability scanning system and demonstrates the feasibility of the technology research through the scanning results.**

***Keywords*— vulnerability scanning technology, detection, automated, internet, SQL Injection.**

1. INTRODUCTION

Web vulnerability scanners are tools employed in the cybersecurity sector to search and test web applications and websites for potential security vulnerabilities. These scanners probe various components belonging to a web system. That includes sniffing out weaknesses within the code, configurations, and inputs as they hunt for security weaknesses that attackers could exploit. They look for signs of a wide array of vulnerabilities, ranging from relatively commonplace issues, like cross-site scripting (XSS), SqI injection and insecure server configurations, to more exotic flaws, such as authentication bypasses and remote-code execution vulnerabilities.

Web vulnerability scanners are practical instruments as they allow assessing the security of web applications in a straightforward and rapid manner. The key advantage of these tools is that they automate scanning, saving hundreds of hours that a single manual assessment would require. Alternatively, it enables organizations to keep their applications safer by scanning them more frequently, which is especially applicable when referring to companies that update their we servies weekly or even daily. An additional advantage provided by most web vulnerability scanners is the prioritization of results. Put differently, they detected vulnerabilities and the probability of their use by hackers. Properly allocate resources to minimize risks.

2. LITERATURE REVIEW

The author [1] explained the common web application security vulnerability detection methods in the power industry.

The author [2] tells us about various vulnerability scanners and their related methodology various vulnerabilities available in the web applications or the remote host network

and tries to find fresh systems that are capable of be deployed to secure the network.

The author [3] derived to elaborate existing web vulnerability detecting approaches with their advantages and disadvantages. Clustering approach has approached to efficiently detect the SQL Injection, X-path Injection and Cross Site Script-Ing attacks ranked by OWASP (Open Web Application Security Project) community. The improvement in detection efficiency of vulnerability scanner while continuing to have low rates of false positives and false negatives

The author [5] say's about foundation for upcoming projects that will end with the elaboration of web scanning and security suggest better innovations.

3. METHODOLOGY

This method approach delves into vulnerability detection technology, which includes simulating hacking to uncover application vulnerabilities. It assesses various vulnerability types and suggests that integrates fuzzy testing to automate the vulnerabilities in web applications.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Volume 13, Issue 03 March 2024

1. XSS Detection:

The most attackable web security vulnerabilities are the XSS vulnerability. This vulnerability allows attacker injecting some malicious script/ code into the web pages. This injected code will execute on the client side of the victim user. With this at his disposal, the attacker can perform many session hijacking and data theft through web server logs viewable in client browsers only by malicious guy who oughtn't be doing anything like that at all. The user input fields on forms that accept unsanitized data are points of attack for this vulnerability. However which other user controlled data contains it depends on IE: search bars, comment sections and contact us forms; query strings in URLs. The most attackable web security vulnerabilities are the XSS vulnerability. This vulnerability allows attacker injecting some malicious script/ code into the web pages. This injected code will run execute on the client side of the victim user. With this at his disposal, the attacker can perform many session hijacking and data theft through web server logs viewable in client browsers only by malicious guy who oughtn't be doing anything like that at all. The user input fields on forms that accept unsanitized data are points of attack for this vulnerability. However which other user controlled data contains it depends on IE: search bars, comment sections and contact us forms; query strings in URLs.

Reflected XSS:

XSS (Cross-Site Scripting) refers to a web security vulnerability that attackers to compromise a vulnerable application. This occurs when an application includes un validated and unescaped user input within the generated output and the user's web browser interprets the input as active content reflected.If an application does not validate or escape user input and then includes it within its out, an attacker can construct a carefully-structured HTTP request to the application that contains malicious script tags included a payload that is then sent to a victim. This script is then executed within the victim's web browser when the injected page is returned from the web application, and the attacker can steal session tokens, redirect the victim to another page that they control or any number of other professional mischief. To help protect against reflected XSS, it ensure that robust input validation and output encoding practices are in place. User input is thoroughly sanitised as a first step and also that any output is properly encoded to prevent script execution.

DOM-based XSS:

To rectify a DOM-based Cross-Site Scripting vulnerability, sections in the code of the website examined must be found where user input manipulates JavaScript to alter the display and function of the site. Now, the data must be checked as to whether user input undesirable changes how the website works.And finally, a clear understanding of how changes within the Document Object Model will affect user input and how this is then woven go the web page. Write some dirty code that can affect the appearance of the area under your control, e.g., run scripts not sanctioned by those who own and shee the site.

Then, this code must be injected into the selected input and changes observed. After that, specific evidence of vulnerability is positioned in the documentation; this can take the shape of a section of the report. And then, additionally, it is a good idea to recommend the owner of the website fix that problem. Tell them some information about why you could manually reproduce what caused it and possible suggestions on how to fix it. This includes suggesting for example that the website implement some client-side security mechanisms, like a Content Security Policy. However, it matters in mind that any vulnerability assessment carried out only with the appropriate authorization and strictures of ethical hacking. vulnerabilities is not ethical.

2. SQL Injection:

Finding SQL injection vulnerabilities starts from the methodical testing of exactly possible SQL injection payloads, which start from simple ones and every next one becomes more complicated. Additionally, it is necessary to control the server response for vulnerability indicators. Error-based detection requires the following steps. The sample URLs are obtained with the addition of quotes ‘ or semicolon ; to the value. Obtain the URL details are along with the parameters. Parse the data for injection, and make a URL that the application of SQL injection to it, with the formation of data on HTTP, into the parameters of which it is more correct to place the following attack payload. The last steps are to submit the adderess of the web application server, perform the detection of vulnerability, get the information on the response, and obtain the feature value.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Volume 13, Issue 03 March 2024

3. Remote Code Execution:

a) The scanner begins its search by attempting to identify sections in the web application where a user can place their data, such as forms, or URL parameters.

b) Next, it includes several specially designed payloads in these fields, enabling it to determine whether the application properly interprets them as an operation code.

c) Subsequently, it thoroughly analyzes every single responses that the application has provided and verifies whether the code that was included has been loaded. This stage could entail figuring out signs of peculiar behaviour or error messages.

d) If any of the previously described signs are noted, the scanner may decide to carry out a series of additional tests that would make possible it to confirm the presence of a particular RCE problem.

e) Finally, the scanner provides a report and an inventory of all identified issues it could be used by attackers.

4. Server Side Template Injection:

a) The scanner checks if the internet-based program uses any server-side template engines.

b) The worm throws special inputs into the parts where template engine collide.

c) The scanner formulates payloads exploiting the template engine's syntax, that permit the running of code.

d) Requests which have been injected are forwarded a software to see how it responds.

e) The scanner looks for outputs that were not supposed to occur or error messages such as those in blue-- which might suggest a STTI vulnerability. The scanner produces a report with its findings on SSTI vulnerabilities identified.

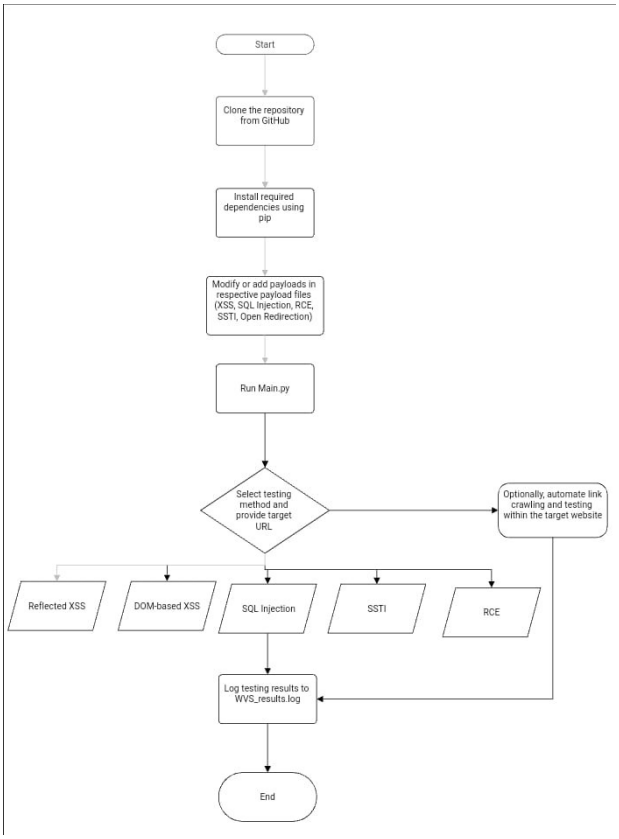

Fig 3: Flow chart

4. RESULT & DISCUSSIONS

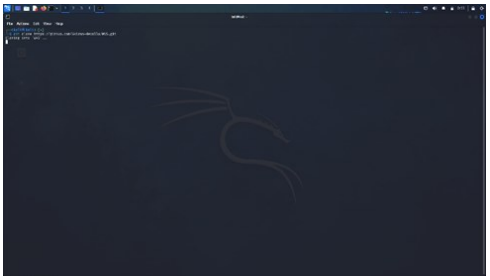

Fig 4.1: Cloning the respository from github

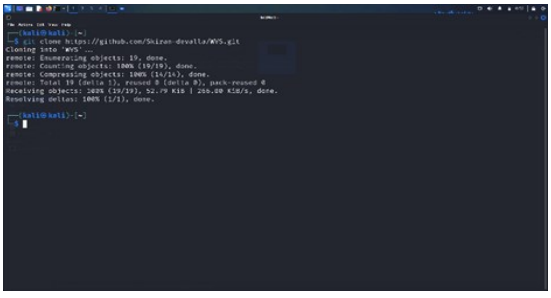

Fig 4.2: Installing the requeriments

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Volume 13, Issue 03 March 2024

Fig 4.3: Select the process

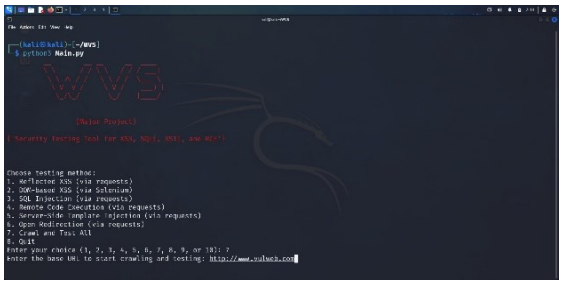

Fig 4.4: Enter URL

Fig 4.5: Scrutinize

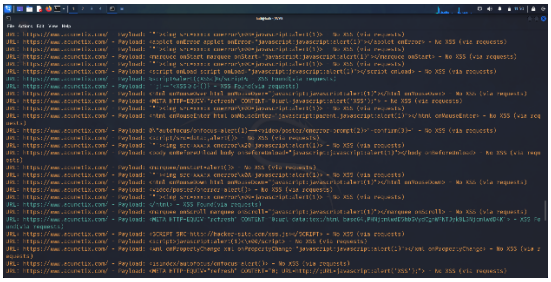

Fig 4.6: WVS_result.log

6. BENEFITS

Efficiency: Rapid scanning applications
Consistency: Consistent scanning can prevent human error
Coverage: Comprehensive inspection over every piece of an application, including authentication and session management.Timeliness: Regular scans mean a constant watch for new vulnerabilities.

Scalability: Capable of dealing with the growing needs for scanning larger applications
Cost-Effectiveness: New contract, old price. This long-term saving compared to manual testing means Your investment is money well spent eating of the good food for
Risk Reduction: Testing early in the life cycle and continuing throughout development assists in avoiding security breaches by discovering vulnerabilities before they are exploited. Also vital to maintain the security of delicate data is what this kind of testing does.When the the capacity to execute the sentences has gone up.

7. CONCLUSION

We conclude that this paper explores the operational principles of a vulnerability detection software and evaluates widely used application vulnerability detection technologies. It creates and deploys a specialized Web vulnerability application scanning system for a Power Company, capable of concurrently scanning multiple target websites with a degree of scalability. Future plans involve refining the web crawler algorithm to enable more thorough data collection from internal websites and improving the identification of vulnerabilities technology to expand coverage.

8. REFERENCES

[1] Bin Wang, Lu Liu, Feng Li, Jianye Zhang, Tao Chen & Zhenwan Zou (2020)

[2] Sheetal Bairwa, Bhawna Mewara & Jyoti Gajrani (2015). Vulnerability scanners: a proactive Approach to assess security.

[3] Rathod, S.K. Jagtap, J.R. Satpute, A. P. Shikhare4, K.A. Pujari, A.S. Pandit (2021). An Automatic scanner for vulnerabilitis WebApplications with Firewall Techniques

[4] Deeptha R, K.Sujatha, D.Sasireka, R.Neelaveni , R.Pavithra Guru (2022).

[5] H Yulimanton 1,2*, H L H S Warnars1, B Soewtiol, F L Gaol1 And E Abdurachaman (2021). Websecurity And Vunerabilitys

[6] Suliman Alazmi (Member, IEEE), and daniel conte de leon (Member, IEEE) (2018). An organized knowledge on the features and performance of web application vulnerability scanners

[7] Emre erturk, Angel rajan Year of publishing (2016). Vulnerabilities on the web, case study.

[8] Asra Kalim, C K Jha, Deepak Singh Tomar, Divyia Rishe Sahu Year of publishing (2015). A structure for identifying vulnerabilities in web Applications Author name

[9] W3af. http://w3af.sourceforger.net/.

[10] web Application vulnerability Scanner Evaluation Project (Vulnerability Scanner Evaluation Project) (2014) http://code.google.com/p/wavsep/

[11] Michael Schmidt, Thomas Röthlisberger, HTML5 web, security [EB] Compass Security AG, December 6th, 2010

[12] Gordon Lyon.Top 125 Network Security Tools [EB] (2012). http:// sectools.org/

[13] Evaluation Project [EB] (2013) http://www.sectoolmarket.com/ Shay-Chen the Web Application vulnerability Scanner