National College of Ireland

Comparative study of SAST and DAST tools with Manual penetration testing methods for Improving the identification of business logic vulnerabilities of authorization flaws and broken access control and false positive vulnerabilities for enhanced Web application Security

MSc Research Project
MSC CYBER SECURITY

Drisya Mohan Kondhankuzhiyil Radhamohanan
Student ID: 22210504

School of Computing
National College of Ireland

Supervisor: Kamil Mahajan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	DRISYA MOHAN KONDHANKUZHIYIL RADHAMOHANAN
Student ID:	22210504
Programme:	MSC CYBER SECURITY **Year:** 2024
Module:	INTERNSHIP
Supervisor:	KAMIL MAHAJAN
Submission Due Date:	02/09/2024
Project Title:	Comparative study of SAST and DAST tools with Manual penetration testing methods for Improving the identification of business logic vulnerabilities of authorization flaws and broken access control and false positive vulnerabilities for enhanced Web application Security.
Word Count:	6603 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	DRISYA MOHAN KONDHANKUZHIYIL RADHAMOHANAN
Date:	01/09/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Comparative study of SAST and DAST tools with Manual penetration testing methods for Improving the identification of business logic vulnerabilities of authorization flaws and broken access control and false positive vulnerabilities for enhanced Web application Security

DRISYA MOHAN KONDHANKUZHIYIL RADHAMOHANAN
22210504

Abstract

In the current scenario, as dependency on internet continue to grow, organizations are increasingly able to handle and share even critical information with ease. As a result, every organization will make use of necessary security measures for protecting their resources such as web applications, websites and other cloud services. At the same time malicious people always will be trying for finding loopholes and the ways to exploit it. In order to identify the weakness in the existing system security, experts will be performing various vulnerability assessments as well as penetration testing. Vulnerability assessments using automated tools such as SAST and DAST tools will be helpful for finding the security weaknesses in the system easily with low cost. But these tools are designed only for evaluating the specific issues it is programmed for. Hence such tools may not be successful always for the identification of logic errors or logic flaws in the system. But when the security experts go for manual penetration testing, they will test further and simulate the real-world attack scenarios to identify the weakness which can exploit by attackers. This paper will be investigating how efficient the manual penetration testing methods than automated tools for identifying the business logic errors occurs due to the application design flaws for enhancing the overall web application security.

Keywords: Business logic errors, SAST, DAST, Penetration testing

1 Introduction

As the organizations adapt the latest technologies and web platforms, its security becomes the important objective for them. Attackers will be always trying to find loopholes in the system security to access the critical informations from the organization. When the organizations give priority for their basic functionalities than provide security, it results for bigger cyber security impacts and affect the reputation of the company. Hence it is important for identify

the system weaknesses at its early stage before it will get exposed to the attackers. Security professionals are developing automated scan tools such as SAST and DAST tools for identifying the potential weaknesses present in the system. When SAST tool provides real time feedback during the code execution, application security can be identified at the developing stage itself. DAST test is considering as a blackbox testing since the automated tool does not have an access to the application source code.(Apoorva P, 2024) Even though automated scan tools can identify the flaws easier with less manual effort and time , manual penetration testing methods can take the steps further and provide an effective testing strategies by simulating real-world attack scenario such as SQL injection, Cross-site scripting, Brute forces and many more for identifying its true exploitability and resulting damages. Automated scan tests can be effective in terms of costs involved and through its reliability. But when consider multiple limitations involved in automated tool scan, these advantages cost benefit becomes pale. DAST and SAST tools get lack with the scan transparency hence the results verifications become tough. Another important fact that, automated scan tools will be fail to identify the business logic errors in the web application. Even though all logical errors are not vulnerability, through manual penetration testing broken structure in the application can be identified. Business logic errors are the vulnerabilities occurs due to the improper design and implementation of the web application. This weakness can be evoked by an attacker to make unintended activities. Attackers can try to manipulate the functionality for performing malicious goals. Automated scan tools may not be successful for identifying these logical errors. Since automated vulnerability scans tools will generally evaluate only the scenarios it designed to test. Logic flaws will not be exposed during normal use of the application. (Portswigger, 2023) However, the malicious attackers will try to exploit these errors by interacting with the application in a way that developers never intended. Business logic in web application are a set of rules which make sure how the application supposed to work by ensuring application operates in a sensible manner. When the security experts fail to identify this kind of errors that can cause the organization to undergo through serious security consequences such as financial losses, reputational damages as well as operational disturbances. Broken or non-existing input validation of inputs can results for allowing users as well as malicious people to make arbitrary changes in the transaction as well as for passing unexpected inputs to the server side logic. Detection of business logic vulnerability required a broad understanding of how different part of the application works. (Portswigger, 2023) Many enterprise's web applications will be having a custom logic and work flow. Since the automated scan tools following pre-defined scripts and patterns, which will be limiting them to provide a effective analysis for new or unforeseen vulnerabilities. Here comes the importance of manual penetration methods. It is important to make sure how effective the manual penetration testing methods than the automative testing tools in the organization. This understanding brings to the following research question.

Research Question:

How do the limitations of automated tools coming under SAST and DAST in detecting complex business logic issues like authorization flaws, broken access control and false positive vulnerabilities emphasize the need for manual testing efforts in web applications.

Objectives:

- Perform automated vulnerability analysis in the web application using both SAST and DAST tools
- Perform manual penetration testing to identify the existing vulnerabilities in the web application includes business logic error scenarios.
- Evaluate the results of both testing methods and identify its effectiveness to defend against cyber attacks

2 Related Work

As the world is digitizing every day, hackers will be evolving new attack methods for achieve unauthorized access to the system. Various research methods are introduced for analysing as well as to detect the weaknesses in the system that attackers might exploit. Research papers introduced by different researchers are helpful for analysing vulnerability analysis with automated tools and manual penetration testing methods for identifying its effectiveness for finding security issues as well as to get insight towards the techniques they used for the analysis. These papers will be also highlighting where each research works compromised for the identification of the weaknesses and how the other researcher could provide better analysis and defence mechanisms.

Khera..., *et.al* analysing the life cycle of the vulnerability analysis and penetration testing process along with the tools for identifying weaknesses present in the system in their research paper. The researchers also provide importance of organization's requirement to update the security measures. The foundational idea on the VAPT and its evolution, methodologies its significant provides insight towards the effective cyber security practices. Through this paper the authors provide a broad insight towards the vulnerability assessment and penetration testing and its different stages such as scope, reconnaissance, vulnerability detection, information analysis, weakness exploitation, privilege escalation and reporting. This life cycle model highlights the importance of the structured process for addressing all potential weaknesses. But this paper is lack for discussing the different types of security issues and express how effective these stages to identify it as well as the advantages of using automated vulnerability scan tools during the various stages of testing methods.

Altaf ...*et.al* in their research paper explains the web application vulnerabilities by focussed particularly on SQL injection and its types. Along with the manual penetration methods this research paper identifies the automated testing strategies as well to improve the accuracy and effectiveness in the vulnerability analysis. SQL injection occurs when the web application fails to provide proper input validation, it can help attackers to send harmful codes and to create database queries instead of legitimate inputs. It can trick the database for sharing or revealing confidential informations. This research paper discusses about the static analysis in order to identify the areas in the PHP application, which can be vulnerable for the SQL injection. Bypass authentication, Union based SQL injection, Firewall bypassing are the

scenario tested during the research. The Web application tested for analysing the security weakness using an automated tool called Acunetix. This tool identified the common threats in the application by providing suggestions to fix it.

On further research it is identified that, Gautam A and Vijay T *et.al* introduced a research paper for improving the security in the websites and web applications especially the organizations from financial side. In order to improve the vulnerability assessments and penetration testing the researchers introduced a model framework for identifying the security weaknesses in the financial enterprises. They performed the penetration testing in the website in order to identify the weaknesses in the existing security mechanisms. Click Jacking, XSS scripting, SQL injection, Private IP disclosure were the vulnerabilities identified from manual penetration. Through the vulnerability test performed using automated tool such as Nessus and owasp zap also helped to identify the same vulnerabilities with very slight difference in their percentage statistics. After identifying these weaknesses, authors designed a new security framework for making the website more secure. Using this frame work when a user tries to login to the website, as part of authentication check they introduced, two factor authentication, verification of security questions as well as reference number along with login time pattern happens for verifying system Ip host names. In comparison with the previous research papers, these researchers provide the mitigation strategies as well for identified vulnerabilities. It will help the enterprise from exploitation of the vulnerabilities by the attacker. At the same time Gautam A and Vijay T *et.al* is not focused on identify the logical errors in the web application, it is required to check all possible errors or vulnerabilities present especially when consider a financial institutional safety.

In the research paper proposed by Alkofahi H..*et.al* proposing a new method for identifying the security flaws in the web application. The method specifically concentrates on resolving the flaws effects on authorization. Many web applications will not be provided with a rule about how the user supposed to interact with the system and makes it difficult to test for the security issues. The authors have suggested a black box testing mechanism in which they will be analysing the user interaction or behaviour with the application for identifying underlying authorization rules. User behaviour analysis will be what the users will do and what is the accessibility they have. Based on the similarity of the user actions groups will be generated using a technique called agglomerative hierarchical clustering. Automated role identification helped to detect users like admin, user, guest to the application access.

Application logic vulnerabilities are the security faults happens due to the incorrect decision-making or logic. These vulnerabilities are very difficult to define and detect. Un detected logic errors or vulnerability in the application can leads to potential security risks. Viktoria F *et.al* proposed research paper which introduces initial approach for automatically detecting the logic vulnerabilities in the web application. The process will be including various steps such as dynamic analysis to observe how the application normally works, based on the observations generating set of basic behavioural rules and specifications. By analysing the usual execution patterns the researchers were able to minimize the number of false positive alarms during web application testing. In order to verify the established behavioural rules.

Viktoria F *et.al* have tested different possible program paths using different inputs which represent for range of possible values. In order to implement these test strategies, the researchers implemented a tool called Waler and they were able to identify previously unknown logic vulnerabilities from several web applications. The main drawback for this tool was the logical error associated with the web application that the tool Waler can detect are restricted to some specific rules or methods it is currently has for detection. Another limitation was if the tool will not accurate there will be chance for missing some existing issues.

The Research work introduced by Fangqi S *et.al...*, proposes the initial static detection methods for identifying the logic vulnerabilities in the e- commerce web applications. Author comments that automating the detection of correct payment logic by ensuring integrity and authenticity is important. The researchers are created a tool for analyzing the PHP code. During testing the authors were successful for finding 12 new logic errors which was unknown earlier. The framework developed by the researchers provides combined symbolic execution and taint analysis for monitoring logic flaws during payment and different stages in the checkout process. For future analysis authors suggesting to include more strategies for exploring logic paths in the e-commerce website and recommends this analysis to popular e-commerce web services.

On further research, it is came to notice that Nikhil R.. *et.al* introduced a research paper for making comparative analysis of automated scanning and manual penetration testing for improved cyber security. The authors also performed automated scanning and manual penetration testing for identifying its effectiveness for identifying vulnerabilities in the system. In order to analyse this, the researchers were performed automated tool scan on a selected website using a tool called Netsparker and identified SQL injection, Reflected cross site scripting, Cross site request forgery and Clickjacking as vulnerabilities. When on the same website they produced manual penetration testing except SQL injection vulnerability rest of the vulnerabilities could able to identify as same as the automated tool. SQL injection weakness identified by the tool became a false positive result here. In addition to these result, manual testing approach helped to identify URL re-direction, privilege escalation, brute force, indirect object reference vulnerabilities as well. From this research the authors finding a conclusion that manual penetration testing is more effective than automated vulnerability assessment tools.

Each paper provides an insight into different researchers approach and ideas for finding effective solution for vulnerability assessments and system protection form attackers. Some authors suggests that automated scan tools can provide better results and analysis reports in lesser time and with high reliability, while others suggesting for manual pen testing methods instead. On the light of these findings, this research focusses on the limitations of automated tools coming under SAST and DAST in detecting complex business logic issues like authorization flaws, broken access control and false positive vulnerabilities to emphasize the need for manual testing efforts in web applications.

3 Research Methodology

Security of the web platform becomes huge concern in almost every organization in the world. Security experts will be always look for better defence techniques and technologies for early identification of the system weaknesses and provide better security. Vulnerability analysis and penetration testing performing an important role for the identification and mitigation of these weaknesses. When vulnerability assessments reveal the weakness the system have, penetration testing will be performing a step additional by simulating real world attack strategies to exploit these errors. (Nikhil R, 2024) In this project SAST and DAST tool used for identifying the security flaws existing in the web application. In addition to that manual penetration testing methods introduced for identifying business logic vulnerabilities. This stage contains the details and steps followed during the research such as website hosting, testing with SAST tool and its analysis, testing with DAST tool and its analysis, performed manual penetration testing use cases and final analysis of the test results.

3.1 Web Site Hosting:

Firstly, an e-commerce website which has developed by PHP programming language selected and the source code selected from the Code Astro website. Selected website is developed by incorporating all the functionalities required for an e-commerce purchase with the list of products having respective price and containing own description, features and photos. The project divided into two different categories with admin privileges and user privileges. When a new user wants to access the functionalities in the website they have to register. Customer can find the products, add required one in to cart, add in to Wishlist. Customer can follow For the checkout procedure for buying one product. Admin have all the privileges to monitor the orders made by the customer and the corresponding order details. (Adminastro ,2021)

In order to test the security feature added in the website, firstly the website hosted in the kali Linux installed in the virtual machine using XAMPP for running the project from local host services without the requirement of the operating system. Apache and MYSQL were used as the servers. After downloading the project file from the Code Astro website, unzipped it and copied into the XAMPP directory in the machine. Inside the htd docs folder in the XAMPP directory the extracted project folder pasted. From the browser followed the URL path http://localhost/phpmyadmin for creating the database as per the 01 LOGIN DETAILS & PROJECT INFO.txt. After importing the .sql database file, it was able to access the website through the URL http://localhost/ecommerce-website-php(1).

3.2 SAST Testing using SonarQube:

Static Application Security Testing or SAST tool will be used for analysing source code to identify the security issues. These tools will be helpful for the continuous integration and continuous development during application development to analyse the code and report vulnerabilities. But it is not assured that all types of vulnerabilities get accounted by SAST tools. Increased level of identified false positives are yet another issue expecting from SAST

tool. In this project SonarQube used for performing SAST vulnerability testing. SonarQube is one powerful static code security testing tool and after performing scan it will be providing an output with identified vulnerabilities, duplications, security hotpots etc. From these results the developer can analyse and modify the program code of the application in order to avoid the risk of attack during the time of the deployment. Firstly, sonar scanner is required for scan the projects from Sonarqube. After downloading the sonar scanner sonar scanner properties file edited and replaced the URL with URL of Sonarqube server http://localhost:9000 after this sonar scanner added to the environment path variable. After this system got rebooted to verify the updated path for the sonar scanner. Next step was the installation of SonarQube. Downloaded the community edition of SonarQube accessed the unzipped SonarQube directory to start. Once Sonarqube is up and running it will be accessible from the http://127.0.0.1:9000 URL. For begin the test, firstly it is required to create a local project locally, then the tool will provide a token for authentication. After generating the token, the provided command can be used for executing the sonar scanner. (OxNehru ,2024)

3.3 DAST Testing using Burpsuite:

DAST tools or Dynamic Application Security Testing tools will be helpful monitoring the security vulnerabilities present in the system while the application is in running state. DAST tools are not specifically focused on the software but focuses on the application layer. DAST tools will be able to identify the common vulnerabilities such as SQL injections, Cross site Scripting, broken authentication and many more. In this research Burp suite is taken for performing the scan. Burpsuite is the popular DAST tool used by the security people for performing vulnerability analysis. (S.Ihor,2023) After Downloading the Burp suite, in order to begin the scan, from the dashboard new scan selected and scan type selected with crawl and audit. Website URL provided to scan. Using both HTTP and HTTPS protocol setting Burp scan performed in the website.

3.4 Manual penetration Testing:

Penetration testing will be evaluating the security in the website through identifying vulnerabilities and exploiting it. From this testing approach the security experts will able to evaluate the impact and exploitability of the existing vulnerabilities. In this project besides of identifying the usual vulnerabilities such as SQL injection, Cross site Scripting, Cross site request forgery and all, aim was focussing more on business logic vulnerabilities and identify how attackers might going to exploit it for their gain. In this research Excessive trust in client-side controls, failing to handle un-conventional inputs, making flawed assumptions on user behaviour are selected as the business logic error use cases for performing manual testing. In addition to that SQL injection also verified. For verifying these vulnerabilities while operating the web application traffic interrupted from burp suite and performed exploitation. Business logic vulnerabilities are the flaws happens and un noticed during the phase of application design and implementation. This fault will be misused by the attacker for their malicious goals. It will be invisible for the security professionals while not giving

attention to the logic error in the application. These errors will be unique for each application, for their identification that particular domain knowledge is essential for the security professionals. Primarily such issues will affect for the functionality of the application, but when it affects to the authentication mechanism it can lead to serious security issues. Normally there is an assumption that a user will interact with the application only with the web interface. And that input will be check for integrity before forwarding to the server-side logic. Here in the research web application lack for providing the server-side logic security.

3.4.1 Use Case 1: Excessive trust in client-side control:

In order to verify the Excessive trust in the client-side control weakness in the application, from the Burp Proxy the data in the quantity of purchase tampered by providing negative value. Proper integrity check of the traffic sent to the server-side logic fails here. Application takes the negative quantity value. This vulnerability can make use by an attacker to perform malicious activity from the burp proxy and make changes to the input data. From this scenario it is clear that client-side verification is not reliable for ensuring security of an application security.

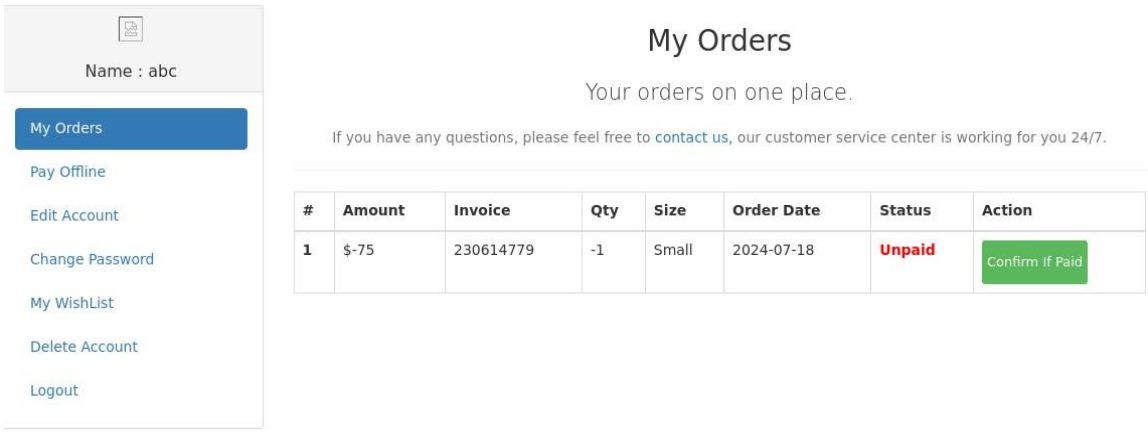

Figure 1. Excessive trust in client-side control

3.4.2 Use case 2: Failing to handle un conventional input vulnerability:

Failing to handle un conventional input vulnerability was the next logic error identified in continuation with the previous error. When application is designed in such a way that to accept certain data type of input value, the application logic has to determine that is an acceptable value or not. In this research the application is designed for providing e-commerce, customers will be ordering products by specifying quantity and shipping details. The business logic will be preventing the quantities out of stock but at the same time it allows to takes quantity values as negative with no logic. Since the application does not verifying the server-side logic the negative values altered through intercepting traffic from burp suite get passed and takes input and proceeds for checkout of the product.

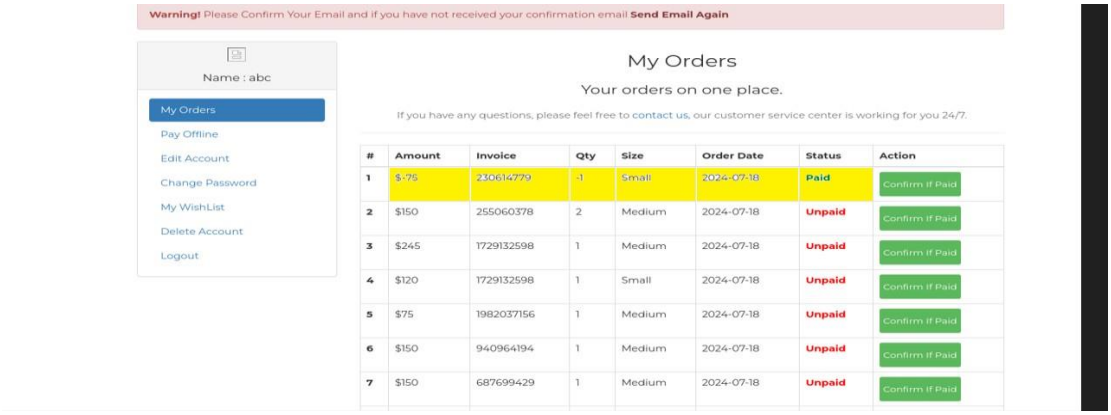

Figure 2. Failing to handle un conventional input vulnerability

3.4.3 Use case 3: Inconsistent security Control:

Yet another logic error identified is that the application makes the flawed assumption about the user behavior and results for inconsistent security control resulting privilege escalation. From the website when the admin login page is intercept from burp suite and even the session cookie of admin gets altered with the user cookie, the user get access to the admin login page. This results for the admin privilege escalation. When the regular user login to the application, they will be assigned with a session cookie with a session ID. Through the burp suite it is able to intercept the traffic and modify the admin session cookie with user's cookie, it does not deny the access. It affects the role-based access control and results for privilege escalation.

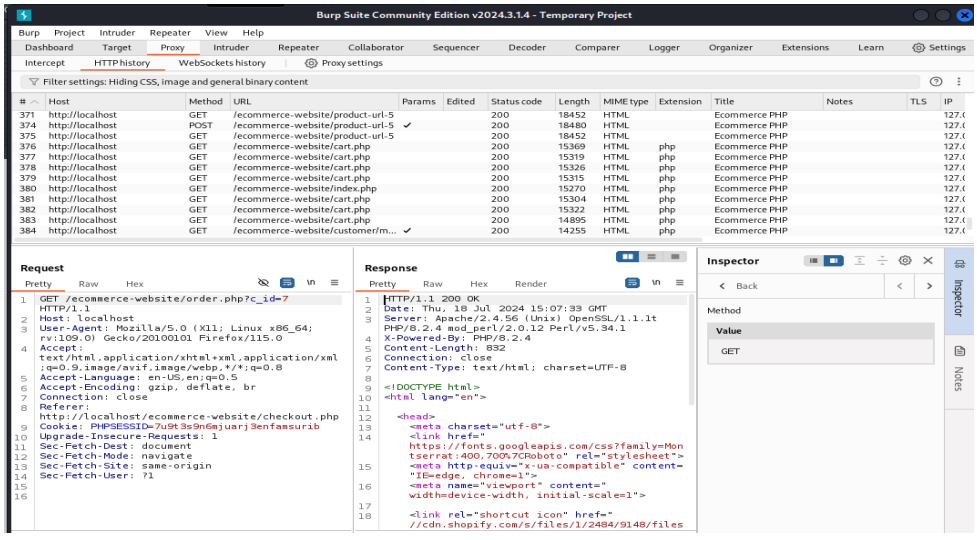

Figure 3. Privilege Escalation to admin page with user login cookie.

3.4.4.Use case 4: SQL Injection:

SQL injection is the common attack the malicious people will follow for exploiting the vulnerabilities present in the web application by providing malicious SQL code as a query. In order to verify this, during navigating to the user login page in the website traffic intercepted and added SQL query in the field of password as ‘ OR ‘1’=’1.After modifying the request it forwarded to the server from burp suite. The website bypassed the authentication and gained access to the website.

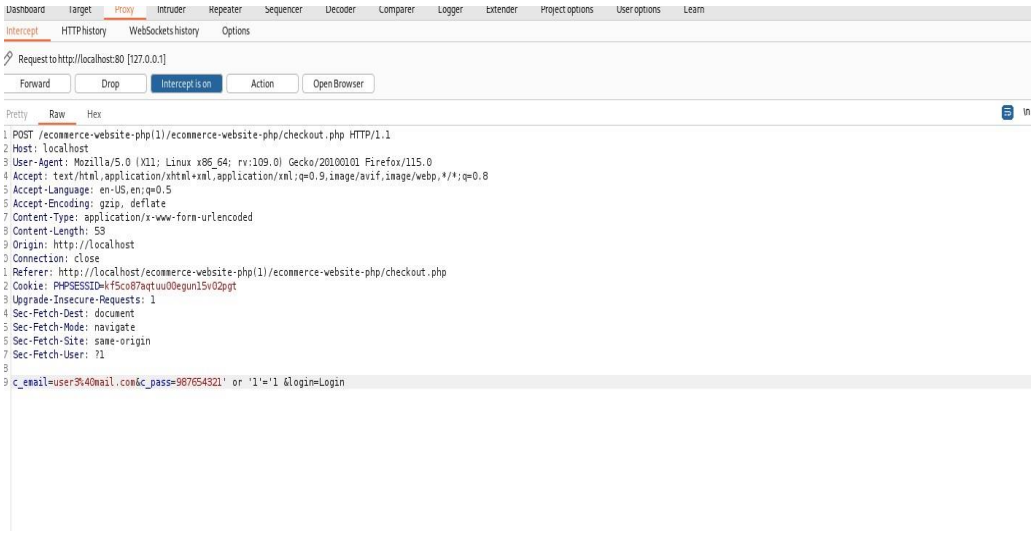

Figure 4. SQL injection

4 Design Specification

4.1 Environment Setup:

Oracle VM Virtual box :

In order to conduct the research work the web application testing setup on a virtualized environment for that Oracle VM virtual box is used. Hence it was able to create a isolated environment for testing the web application without affecting the primary OS. In this virtualized platform multiple operating systems can run and perform testing. It is an open-source virtualization software and it can be available for Windows, Linux as well as macOS. It will be supporting different networking modes such as NAT, bridged, Host-only and internal networking for simulating testing over various network environments. NAT network mode is selected for this research.

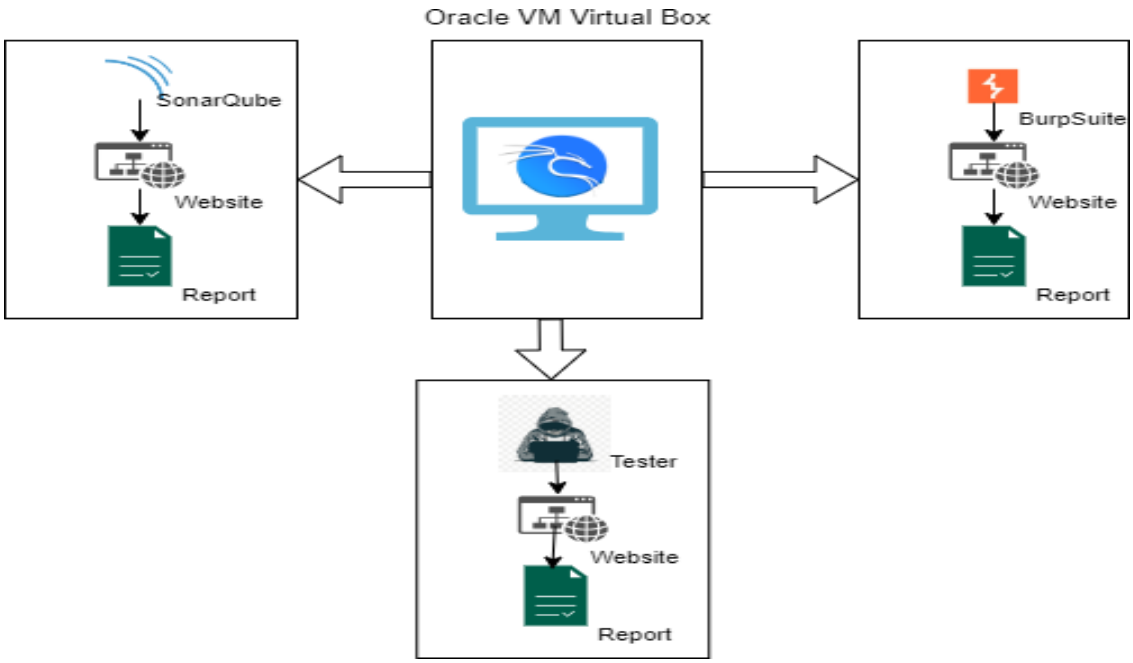

Figure 5. Design Specification.

Kali Linux:

A Debian Linux based Kali operating system chosen for this project. It is also an open source and containing multiple built-in functionalities for performing vulnerability analysis and penetration testing. Kali Linux version 2024.2 installed in the virtual machine and configured with basic settings such as 4096 MB memory, 16 MB video memory.

XAMPP:

XAMPP is the free and open source platform for setup a local webserver especially useful for web development as well as testing. XAMPP will be consists of a cross platform in order to support in all operating systems such as Windows, Linux and macOS. Apache server used for handling the HTTP requests and provide the web content. MYSQL will be helpful for storing and managing the data. PHP and Perl are the programming languages used for web development. In this project the website developed using PHP code ,XAMPP can able to host it.

4.2. Tool Setup:

SAST Tool:

In order to perform static code analysis and identification of vulnerabilities in the application code SonarQube is used for this project. It provide thorough understanding for the management and measure code quality through code analysis. It helps for identifying the

potential security issues or vulnerabilities in the code to secure from threats before code deployment.

DAST Tool:

In this project Burp suite used for analyzing the security vulnerabilities in the web application during its normal operations. Burp suite containing automatic vulnerability scanner for detecting the common cyber security issues such as SQL injection, Cross site Scripting, misconfigurations. Burp suite also used for intercepting the proxy and modify the HTTP requests while performing manual penetration testing. Repeater tool in the burpsuite is helpful for manually modify and resend the request to verify the application responds to that.

5 Implementation

Identification vulnerabilities including business logic errors present in the web application was the primary aim of this research. The research was performed mainly in two stages including vulnerability analysis using automated tools and through manual penetration testing. The vulnerabilities mainly identified from the DAST tool was SQL injection and clear text submission of password. These are the two vulnerabilities identified with high severity.

		Confidence			
		Certain	Firm	Tentative	Total
Severity	High	2	1	0	3
	Medium	0	0	0	0
	Low	2	0	2	4
	Information	3	7	11	21

Figure 6. Severity Chart generated on Burpscan report.

Figure 7. SQL injection reported on Burp scan

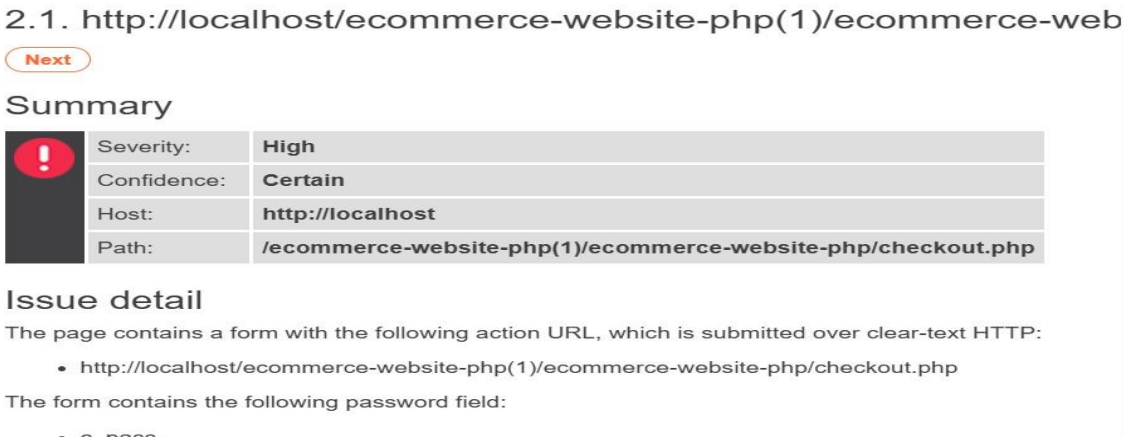

Figure 8. Clear text submission reported on Burpscan

Along with these results lot of false positive issues were also notified in the generated report. Whereas the SAST tool could able to identify the vulnerabilities of not adding password protection to the database and vulnerability associated with recapta key revoke, and regeneration.

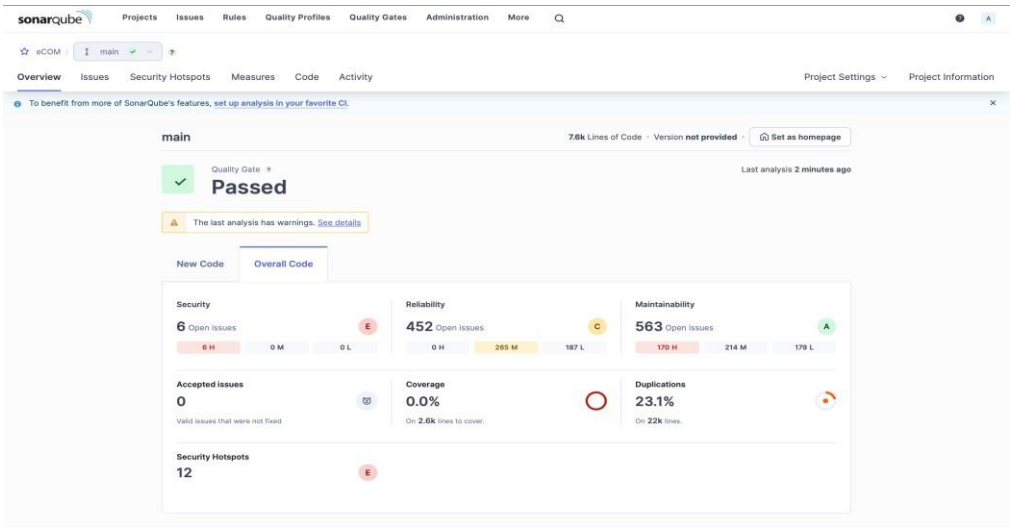

Figure 9 SonarQube test result

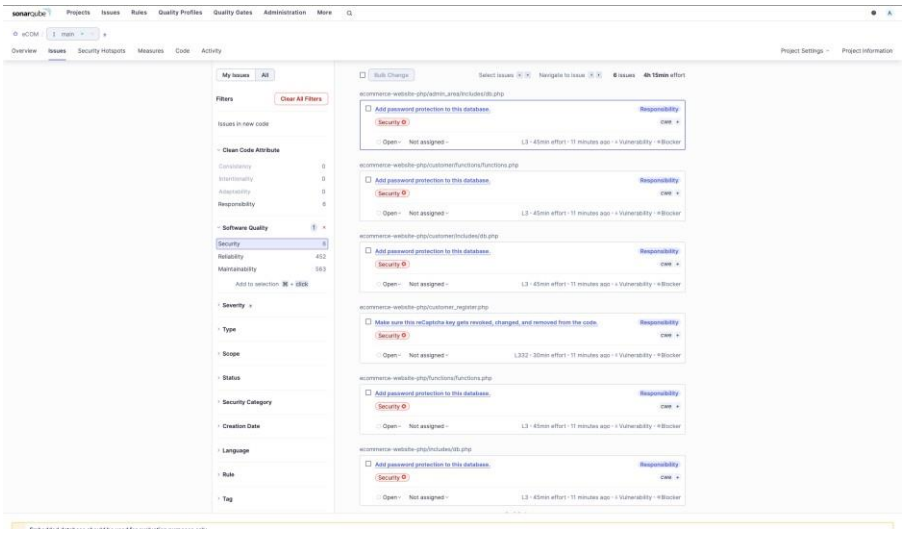

Figure 10 SonarQube test result

Even though there are vulnerability associated with input validation it was not notified during the automation testing. Manual penetration testing use cases were able to identify the web application associated business logic errors. Excessive trust in client-side control, failing to handle unconventional inputs and making flawed assumptions about user behaviour are the use cases tested as part of manual penetration testing along with SQL injection. From the attacker point of view from the burp proxy the input data tampered and it sent to the web browser. In such scenarios ideally the data will be sent to the server-side logic for checking proper integrity check and validates. Here in this webpage that integrity validation not happening due to the trust in the client-side server, hence it makes an attacker to perform all required damages in the service with minimal effort. Also, the application has to be designed in such a way that to accept the arbitrary values of a particular data type. But the application logic has to be decided whether the value given is acceptable or not. When consider the e-commerce website when a product purchases the provided quantity should be logical. Negative values should not be taken for quantities as well as for the product price. Since the application is not provide an adequate server-side validation for reject these inputs the malicious attacker can pass the negative values and results for unwanted application behaviour. Inconsistent security controls result from the flawed assumption of the user behaviour SQL injection was another vulnerability identified in the application.

6 Evaluation

This research was proposed to identify the security gaps exists between the automated vulnerability analysis tools and manual penetration testing methods while performing security testing on the web application. Most of the organizations and enterprises will prefer automated tools for the vulnerability analysis on their system. Definitely, Automated tools have certain advantages. Regular monitoring of the system by running automated scan tool

such as Nessus, Burp suite can be helpful for identifying the security risk associated with the web application. It will help to understand the efficiency of the current security measures taken by the organization. Time management is the biggest advantage by the automated scan tools for the security professional (Keshav M,2024). Since it is an automated procedure, human work load and working hours can get reduced. However, there are few disadvantages as well when organizations depend only on automated scan tools for the identification security weaknesses. As discussed earlier in the research paper, automated scan tools can work only according to how they are programmed. Attackers will be always trying to find new methods for finding loopholes in the existing security measures. In such scenario, when the tool work only as per they programmed for finding common weaknesses such as SQL injection, Cross site scripting and all , dependency only on the tools will not be effective defensive mechanism that a security professional can prefer. In this research as well it was clear that, there were some logic errors also associated with the web service. It couldn't highlight or notified during the test carried out by SAST and DAST tool mechanisms. Instead, when the tools flag the non-issues as vulnerabilities such as false positive errors, it becomes a waste of time by verifying those issues. Here comes the importance of manual penetration testing strategies. Manual penetration testing can make sure zero false positive results after analysing the application. During the research, manual testing was effective to identify the logic errors and weakness present in the application. Even though the manual penetration testing methods could uncover these issues and vulnerabilities present in the system, security professionals cannot solemnly depend on manual testing methods as well. Time consumption by this testing approach is the biggest disadvantages. For the research when automated scan took few hours to complete, manual testing took more time in days to complete. It is not feasible to test every part of the complex as well as large part of the application with in a limited period of time. According to the tester's skill, experience and knowledge the quality of the tests scenarios also can be verified. Human errors become the barrier for here. The testers can make wrong analysis and mistakes while recognizing a new attack scenario occurs. When the automated tool performs vulnerability analysis repeatedly with in a duration of time, manual testing can not be possible to handle the repetitive testing efficiently. From the analysis it is clear that identification of system weaknesses using automated tools can not provide in depth analysis and effective identification of security issues as manual testing. But the property of speed of analysis and scalability provided by automated tools are remarkable. Importance of balancing both automated tools and manual efforts are occurring here. Combination of these methods can improve the over all security of the application especially when consider the business logic vulnerabilities.

7 Conclusion and Future Work

In the current scenario there are different types of attacks and data breaches happening in the web applications. Security professionals are taking different approaches to identify the security weaknesses in the system such as vulnerability analysis and penetration testing using automated tools as well as manual approaches. Comprehensive approach during testing will help to make sure the vulnerabilities and other system weaknesses are getting identified in

timely manner using multiple methods and testing tools. It will be ensuring thorough security testing and the identification of mitigation strategies for robust protection against the application threats. From the analysis it is clear that security analysis using both automation testing tools and manual penetration tastings methods have certain advantages as well as disadvantages. During the chosen web application testing using Sonarqube, the tool examined the source code before the application in its running state and analysed the code paths and vulnerabilities exists. Similarly for verifying the security issues associated with the application in its running state Burp suite tool is used. It could also identify the issues such as SQL injection, cross site scripting along with false positive errors. The manual penetration testing provided insight towards the business logic flaws present in the web application. From this research came to the conclusion that, combination of both manual testing efforts as well as automated tools are required for effective identification of security weaknesses present in the system. Most of the organizations will be depending on the automated scan tools only for vulnerability analysis. When the logic errors become specific and unique to a particular application or webservice, automated scan tools cannot address them. It can identify only through manual testing approaches. When organizations can adopt a balanced approach through combining SAST, DAST and manual penetration testing methodologies, they can enhance the cyber security and defensive mechanisms.

As artificial intelligence and machine learning provides advanced technologies and strategies towards cyber security. When integrate artificial intelligence in the vulnerability assessment automated tools, these tools will be capable for detecting the complex security threats associated with the system and threat patterns that might not be noticeable during traditional detection methods. Learning from the previous attack scenarios organizations can implement machine learning methods for predicting the potential vulnerabilities present in the system. These mechanisms will be also helpful for reducing the false positive error detection. With the continuous improvement take place in the AI / ML technologies enterprises can make sure their system security with accurate assessments and building proactive security mechanisms. Hence it is recommended to integrate these technologies in automated scan tools for improvised vulnerability assessments in the system and to strengthen the defence mechanisms.

References

Apoorva P ,'SAST vs. DAST: What's the best method for application security testing?' (2024), Available at: https://www.synopsys.com/blogs/software-security/sast-vs-dast-difference.html [Accessed on Aug 12 ,2024]

Portswigger,(2023)'Business logic vulnerabilities', [Online] Available at : https://portswigger.net/web-security/logic-flaws [Accessed on Aug 12 ,2024]

Y. Khera, D. Kumar, Sujay and N. Garg,(2019) 'Analysis and Impact of Vulnerability Assessment and Penetration Testing', *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Faridabad, India, 2019, pp. 525-530, doi: 10.1109/COMITCon.2019.8862224. Available at: Analysis and Impact of Vulnerability Assessment and Penetration Testing | IEEE Conference Publication | IEEE Xplore [Accessed Aug 12,2024]

I. Altaf, F. u. Rashid, J. A. Dar and M. Rafiq,(2015) 'Vulnerability assessment and patching management,' *International Conference on Soft Computing Techniques and Implementations (ICSCTI)*, Faridabad, India, 2015, pp. 16-21, doi: 10.1109/ICSCTI.2015.7489631Available at: Vulnerability assessment and patching management | IEEE Conference Publication | IEEE Xplore [Accessed Aug 12,2024]

A Goutam and V. Tiwari,(2019),'Vulnerability Assessment and Penetration Testing to Enhance the Security of Web Application', *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, Mathura, India, 2019, pp. 601-605, doi: 10.1109/ISCON47742.2019.9036175,Available at: Vulnerability Assessment and Penetration Testing to Enhance the Security of Web Application | IEEE Conference Publication | IEEE Xplore [Accessed Aug 13,2024]

H. Alkofahi, D. Umphress and H. Alawneh, "Discovering Authorization Business Rules toward Detecting Web Applications Logic Flaws," 2022 International Arab Conference on Information Technology (ACIT), Abu Dhabi, United Arab Emirates, 2022, pp. 1-7, doi: 10.1109/ACIT57182.2022.9994086. Available at: Discovering Conditional Business Rules in Web Applications Using Process Mining | Information Integration and Web Intelligence (acm.org) [Accessed Aug 13,2024]

S. Fangqi ,L. Xu, Z.Su, 'Detecting Logic Vulnerabilities in E-Commerce Applications', [Online] Available at: ndss14.pdf (ucdavis.edu) [Accessed Aug 13,2024]

N. Rane and A. Qureshi, "Comparative Analysis of Automated Scanning and Manual Penetration Testing for Enhanced Cybersecurity," 2024 12th International Symposium on Digital Forensics and Security (ISDFS), San Antonio, TX, USA, 2024, pp. 1-6, doi: 10.1109/ISDFS60797.2024.10527240. Available at : Comparative Analysis of Automated Scanning and Manual Penetration Testing for Enhanced Cybersecurity | IEEE Conference Publication | IEEE Xplore [Accessed Aug 13,2024]

Felmetsger, V., Cavedon, L., Kruegel, C., & Vigna, G**.** (2010). 'Toward automated detection of logic vulnerabilities in web applications'. In *Proceedings of the 19th USENIX Security Symposium*, Available at: https://www.usenix.org/conference/usenixsecurity10/toward-automated-detection-logic-vulnerabilities-web-applications [Accessed Aug 14,2024]

G. Deepa, P. S Thilagam, A Praseed, R.Alwyn. ,' DetLogic: A black-box approach for detecting logic vulnerabilities in web applications', (2018), Journal of Network and Computer Applications , Available at: DetLogic: A black-box approach for detecting logic vulnerabilities in web applications (muhammetbaykara.com) [Accessed Aug 14,2024]

OxNehru,(2024),'Step-by-Step Guide to Installing and Setting Up SonarQube on Kali Linux' Medium, Available at:https://0xnehru.medium.com/step-by-step-guide-to-installing-and-setting-up-sonarqube-on-kali-linux-a73fd793e9b8 [Accessed Aug 15,2024]

Adminastro ,2021,' Ecommerce Website in PHP MySQL with Source Code' [Online] Available at: https://codeastro.com/ecommerce-website-in-php-mysql-with-source-code/ [Accessed Aug 15,2024]

S.Ihor,(2023),' Dynamic Application Security Testing: The Ultimate Guide' ,[Online],Available at: https://www.techmagic.co/blog/dast/ [Accessed Aug 16,2024]

Keshav M ,(2024),' A Complete Guide to Automated Vulnerability Scanning'[Online],Available at: A Complete Guide to Automated Vulnerability Scanning - Astra Security Blog (getastra.com) [Accessed Aug 16,2024]