p-ISSN: 2252-3006
e-ISSN: 2685-2411

Penetration Testing on the SISAKTI Application at Udayana University Using the OWASP Testing Guide Version 4

Reyhan Todo Noer Yamin[a1], I Made Agus Dwi Suarjaya[a2], I Putu Agus Eka Pratama[a3]
[a]Information Technology Department, Faculty of Engineering, Udayana University, Bali, Indonesia
e-mail: [1]reyhantodo28@gmail.com, [2]agussuarjaya@it.unud.ac.id, [3]eka.pratama@unud.ac.id

Abstrak

Aplikasi SISAKTI merupakan Sistem Informasi untuk memudahkan proses administrasi satuan kredit partisipasi mahasiswa Universitas Udayana secara online. Sampai saat ini belum ada pengujian keamanan yang dilakukan pada aplikasi SISAKTI, oleh karena itu penelitian ini dilakukan bertujuan untuk menguji keamanan aplikasi SISAKTI menggunakan teknik Black Box penetration testing, melakukan penilaian terhadap kerentanan sistem dan memberikan rekomendasi perbaikan. Metode yang dilakukan adalah dengan mengikuti panduan dari OWASP Testing Guide versi 4 pada modul Information Gathering, Input Validation Testing, dan Authorization Testing dari tiga modul tersebut terdapat 28 sub uji yang berhasil dilakukan, hasilnya terdapat 15 pengujian positif, 6 pengujian negatif, dan 7 pengujian tidak dapat dilakukan, dari 28 sub uji tersebut ter-dapat 8 vulnerability yang memiliki efek langsung terhadap sistem dan dinilai menggunakan CVSS calculator hasilnya 6 vulnerability tersebut memiliki rentan nilai dari 6.4 (Medium) sampai 9.9 (Critical).

Kata kunci: *Black Box Testing, OWASP Testing Guide Versi 4, Penetration Testing, Sistem Informasi*

Abstract

SISAKTI application is an information system to facilitate online administration of Udayana University student participation credit units. Until now, there has been no security testing carried out on the SISAKTI application, therefore this study aimed to test the security of SISAKTI application using Black Box penetration testing technique, conduct an assessment of system vulnerabilities and provide recommendations for improvements. The method used is by following the guidelines from OWASP Testing Guide version 4 using Information Gathering, Input Validation Testing, and Authorization Testing modules. From these three modules, there were 28 sub-tests that were successfully carried out, the results were 15 positive tests, 6 negative tests, and 7 tests which cannot be done, from the 28 sub-tests there are 8 vulnerabilities that have a direct effect on the system and are assessed using CVSS calculator, the results are 6 vulnerabilities have a vulnerable value from 6.4 (Medium) to 9.9 (Critical).

Keywords : *Black Box Testing, OWASP Testing Guide Version 4, Penetration Testing, Information Testing*

1. Introduction

Udayana University is one of the State Universities that develops information systems that are utilized to facilitate campus-scale organizational processes. Udayana University has an IMISSU website (*Integrated Management Information System, the Strategic of UNUD*) as an information system portal that uses the single sign-on method, namely that within IMISSU there are various applications that help Udayana University's information system needs.

SISAKTI (Participation Credit Unit System) is one of the applications available at IMISSU, which is an information system to facilitate the administration process of student participation credit units online, such as information on participation credit unit points, collection of participation credit unit files and validation of student credit units. This system is quite important because the participation credit unit will be calculated at the end of the study (S1) as a condition for attending graduation in each faculty.

According to the location of the attack, there are 2 types of attacks on the web system, namely attacks on the client and server sides. Attacks on the client side cannot change existing data on the server side, but allow attackers to view data that is not supposed to go through certain actions on the client side. Attacks on the client side also make it possible to change the structure on the client side. Examples of attacks on the client side are Cross-Site Scripting (XSS), Cross-XML Entity (XXE), and misconfigurations that display inappropriate data/actions.

Attacks on the server side are usually more dangerous than attacks on the client side because it is more possible to view, modify, and execute certain data on the server. Examples of attacks on the server side are SQL Injection, File Upload Vulnerability, Command Injection, and many more Invalid sources specified [1]. Therefore, this study will conduct penetration testing using the OWASP Testing Guide Version 4 framework, especially in the Information Gathering, Authorization Testing, and Input Validation sections to determine the vulnerabilities found in the SISAKTI application so that later it can measure existing vulnerabilities to provide recommendations for improvements that can be made and implementation of improvements to the system.

Based on the background that has been explained, the formulation of the problem as well as the research objectives was obtained, which amounted to 5 in this study, namely, 1.) What is the method for performing penetration testing of the SISAKTI Application at Udayana University based on the OWASP Testing Guide Version 4? 2.) What are the results of the penetration testing of the SISAKTI Application at Udayana University using the OWASP Testing Guide Version 4? 3.) How risky is the vulnerability found in the Udayana University SISAKTI Application? 4.) What are the recommendations for improvements that can be made to overcome security holes in the Udayana University SISAKTI Application? 5.) How to review the results of the penetration testing of the Udayana University SISAKTI application using the OWASP Testing Guide version 4 after the system was repaired?

2. Research Method

The methodology and design of the system explain the place and time of the final assignment research, an overview of the system and methods used in conducting the final assignment research entitled "Penetration Testing on the SISAKTI Application at Udayana University Using the OWASP Testing Guide Version 4".

2.1. Research Methodology

The research methodology is the stages carried out by the testers to carry out Penetration Testing on the SISAKTI application at Udayana University.

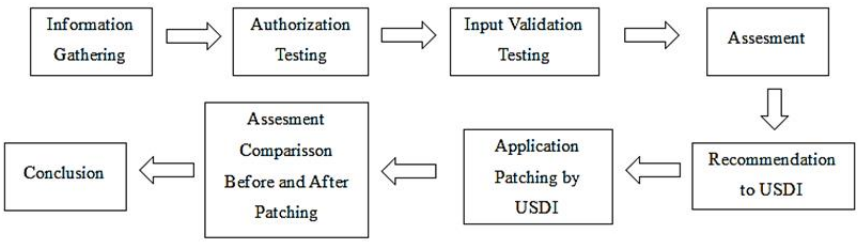

Figure 1. Research Scheme

Figure 1 is the research method used in this study, consisting of three stages of penetration testing, namely information gathering, authorization testing, and input validation testing using the OWASP testing guide version 4, then followed by an assessment, submission of recommendations to USDI for further improvement by USDI, then a re-assessment of the improvements made, and conclusions are made.

2.2. Information Gathering

The information gathering is the stage of gathering information regarding the specifications of the SISAKTI application system, the steps taken follow the OWASP Testing Guide Version 4. The Information Gathering testing phase is carried out from OTG-INFO-001 to OTG-INFO-010. The test starts from OTG-INFO-001 if sensitive information is found from a search engine, the status of OTG-INFO-001 changes to 1, then continues with testing OTG-INFO-002 if you get web server information the status of OTG-INFO-002 changes to 1, then

proceed to OTG-INFO-003, if you find a sensitive path then the OTG-INFO-003 status is changed to 1, then continue with OTG-INFO-004.

If the port used by the web server can be seen, the OTG-INFO-004 status is changed to 1. Based on the OTG-INFO-004 test results, if the database server port is not found, the OTG-INPVAL-005 status is changed to -1, if the Lightweight Directory port is not found Access, then OTG-INPVAL-006 changed to -1. OTG-INFO-005 test, if metadata/comments/sensitive information is found in the HTML code then the OTG-INFO-005 status is changed to 1. Based on the results of the OTG-INFO-005 test if no XML Document code is found then the status is OTG-INPVAL-008 changed to -1, if no server-side include is found then the OTG-INPVAL-009 status changes to -1, if no hidden input is found connected to the server, then the OTG-AUTHZ-003 status changes to -1.

Based on the OTG-INFO-006 test, if there is input to the application that can define the application entry point, the OTG-INFO-006 status is changed to 1, then proceed with the OTG-INFO-007 test, if it can define the execution path contained in the application, the status OTG-INFO-007 is changed to 1. Based on the results of the OTG-INFO-007 test, if an XPath URL is not found, the OTG-INPVAL-010 status is changed to -1, if no URL is found that connects to IMAP/SMTP, then the status is OTG-INPVAL- 011 is changed to -1, if no file inclusion is found in the URL then the OTG-INPVAL-012 status is changed to -1, if no path is found to execute the OS Shell then the OTG-INPVAL-013 status is changed to -1.

Based on the OTG-INFO-008 test, if you can find out the web application framework, the OTG-INFO-008 status is changed to 1, then proceed with the OTG-INFO-009 test. If a vulnerability is found in the service used, the OTG-INFO-009 status is changed to 1, then proceed with the OTG-INFO-010 test, if it can describe the application architecture in general, the OTG-INFO-010 status is changed to 1. After all, tests are completely followed by the authorization testing stage.

2.3. Authorization Testing

Tests carried out after the information gathering testing stage are authorization testing. The tests carried out were from OTG-AUTHZ-001 to OTG-AUTHZ-004. The test starts from OTG-AUTHZ-001, if there is an index of file directory that can be accessed, the OTG-AUTHZ-001 status changes to 1, then continues with OTG-AUTHZ-002 testing, which if there is a page that should not be accessed by users who is not authorized, the OTG-AUTHZ-002 status changes to 1.

Based on the results of the OTG-AUTHZ-003 test, if the status is -1 then the test is not carried out, if not then the test is still carried out also if you can change the access rights of certain users to higher privileges, then the OTG-AUTHZ-003 status is changed becomes 1. Based on the results of the OTG-AUTHZ-004 test, if a direct object is found that can access an unauthorized function, the OTG-AUTHZ-004 status changes to 1, then the test proceeds to the input validation testing stage.

2.4. Input Validation Testing

The next test is input validation testing, the test is carried out from OTG-INPVAL-001 to OTG-INPVAL-016, except for OTG-INPVAL-003 because it requires authentication from SSO from IMISSU, so the test cannot be carried out and the OTG-INPVAL-014 test also not done because the test is only done if the system uses another application that uses a desktop programming language.

Based on the results of the OTG-INPVAL-001 test, if a Reflected XSS attack can be carried out, the OTG-INPVAL-001 status will be changed to 1. Based on the results of the OTG-INPVAL-002 test, if a Stored XSS attack can be carried out, then the OTG-INPVAL-002 status will change to 1. Based on the results of the OTG-INPVAL-004 test, if a pollution parameter is found that is a particular vulnerability, the OTG-INPVAL-004 status changes to 1, then the OTG-INPVAL-005 status is checked if the status is 1 then the test is not carried out, if the status is 0 then the test is carried out.

Based on the results of the OTG-INPVAL-007 test, if an ORM Injection attack can be carried out, the OTG-INPVAL-007 status changes to 1, then the OTG-INPVAL-008 status is checked if the status is -1 then the test is not carried out if the status is 0 then the test is carried out. Based on the results of the OTG-INPVAL-008 test, if an XML Injection attack can be carried

out, the OTG-INPVAL-008 status changes to 1, then the OTG-INPVAL-009 status is checked if the status is -1 then the test is not carried out if the status is 0 then the test is carried out.

Based on the results of the OTG-INPVAL-009 test, if the SSI Injection attack can be carried out, the OTG-INPVAL-009 status changes to 1, then the OTG-INPVAL-010 status is checked if the status is -1 then the test is not carried out if the status is 0 then the test is carried out. Based on the results of the OTG-INPVAL-010 test, if an XPath Injection attack can be carried out, the OTG-INPVAL-010 status changes to 1, then the OTG-INPVAL-011 status is checked if the status is -1 then the test is not carried out if the status is 0 then the test is carried out.

Based on the results of the OTG-INPVAL-011 test, if an IMAP/SMTP Injection attack can be carried out, the OTG-INPVAL-011 status changes to 1, then the OTG-INPVAL-012 status is checked if the status is -1 then the test is not carried out if the status is 0 then the test is carried out. Based on the results of the OTG-INPVAL-012 test, if a code injection attack can be carried out, the OTG-INPVAL-012 status changes to 1, then the OTG-INPVAL-013 status is checked if the status is -1 then the test is not carried out if the status is 0 then the test is carried out. Based on the results of the OTG-INPVAL-013 test, if a command injection attack can be carried out, the OTG-INPVAL-013 status changes to 1. In the OTG-INPVAL-015 test, if an incubated vulnerability is found, the OTG-INPVAL-015 status changes to 1. In the OTG test - INPVAL-016 if a smuggling/splitting injection attack can be performed then the OTG-INPVAL-016 status changes to 1. After all the tests have been carried out, the status of each test code represents the test results, namely 0 (means the test was not successful), 1 (means the test was successfully carried out), and -1 (means the test cannot be carried out because there are prerequisites that are not met).

3. Literature Study

3.1 Penetration Testing

Penetration Testing is a method of evaluating the security of a computer system or network by simulating an attack from a malicious source and is part of a security audit. The simulation of attacks carried out is made like cases that can be made by black hat hackers, crackers, and so on [2].

3.2. Black Box Testing

Black Box Testing focuses on the functional specifications of the software. The tester can define a set of input conditions and perform tests on the program's functional specifications. Black Box Testing is not an alternative solution to White Box Testing but a complement to test things that are not covered by White Box Testing. [3]. Black box testing is done to observe the software's input and output results without knowing the system's code structure. This test serves to test whether the system can function properly. To carry out testing, the tester does not have to have the ability to write program code. This study uses minimum user rights.

3.3. Common Vulnerability Scoring System

The Common Vulnerability Scoring System (CVSS) is a measurement value used by various individuals or agencies to assess a vulnerability to a system. CVSS is a standard commonly used to measure a vulnerability which is then included as a value from a CVE (Common Vulnerability and Exposure) [4]. In 2014, CVSS updated its version to 3.0. CVSS will assess vulnerability from a score of 0.0 – 10.0 which is divided into 4 aspects, namely Low (0.0 – 3.9), Medium (4.0 – 6.9), High (7.0-8.9), and Critical (9.0 – 10.0). CVSS assesses several aspects in assessing the vulnerability of a system which is divided into 8 major sections: Attack Vector, Attack Complexity, Privilege Required, User Interaction, Scope, Confidentiality, Integrity, and Availability. Confidentiality, Integrity, and Availability spearhead the assessment of system vulnerability, if a vulnerability does not threaten these three aspects, then the vulnerability score will remain 0.

3.4. OWASP *Testing Guide* Version 4

The OWASP Testing Guide is a guide for conducting application testing, the OWASP Testing Guide has been compiled, reviewed, and edited by many experts in their fields, so that it can be used as an application testing guide. OWASP Testing Guide version 4 (OTGv4) is an update of the OWASP Testing Guide version 3 (OTGv3). The OTGv4 update has been

integrated with other OWASP documents, namely “The Developers Guide” and “The Code Review Guide”. This is implemented by adding testing categories and test numbering to OTGv4. Each part of OTGv4 has also been increased from 64 test points (OTGv3) to 87 test points, and 4 sections have been added, namely Identity Management Testing, Error Handling, Cryptography and Client-Side Testing. OTGv4 is also structured so that readers do not take what has been explained in this guide for granted, but instead encourage testers to integrate with other software testers to build test cases for the application. It is intended that application security studies can develop more broadly [5]. There are 3 sections tested in this study, namely Testing for Information Gathering, Authorization Testing and Input Validation Testing.

3.5. State of the *Art*

There are 12 pieces of research that have been done before regarding Penetration Testing using the OWASP Testing Guide. The first research was conducted by Adetya Putra Dewanto based on the OWASP Top 10 Application Security Risk 2013 in a study entitled "Penetration Testing Pada Domain uii.ac.id Menggunakan OWASP 10" This study conducted penetration testing on the domain uii.ac.id. [6]. The second research was conducted by Andi Purnawan in a study entitled "Studi dan Implementasi Keamanan Website Menggunakan Open Web Application Security Project (OWASP) Case Study: PLN Batam." This study conducted a penetration test using the OWASP Testing Guide Version 4 on the PLN Batam Website. [7]. The third research was conducted by Mohammad Muhsin and Adi Fajaryanto in a study entitled " Penerapan Pengujian Keamanan Web Server Menggunakan Metode OWASP versi 4 (Studi Kasus Web Server Ujian Online)". Testing was carried out using the OWASP Testing Guide Version 4 in the Authentication Testing, Authorization Testing, and Session Management Testing sections of the Muhammadiyah University Ponorogo Online Examination system. [8]. The fourth study was conducted by Dr. Raden Teduh Dirgahayu, S.T., M.Sc., Yudi Prayudi S.Si., M.Kom., and Adi Fajaryanto. in a study entitled " Penerapan Metode ISSAF dan OWASP versi 4 Untuk Uji Kerentanan Web Server". This research tested the IKIP PGRI Madiun Website application which included Authentication Testing, Authorization Testing and Session Management Testing [9]. The fifth study was conducted by Adi Fajaryanto Cobantoro in a research journal entitled " Penerapan OWASP Versi 4 Untuk Uji Kerentanan Web Server (Studi Kasus E-Jurnal Server Kampus X Madiun)". Campus X Madiun server journal, the sections tested are the Authorization Testing, Authentication Testing, and Session Management Testing sections. [10]. The sixth study was conducted by I Putu Agus Eka Pratama and Anak Agung Bagus Arya Wiradarma in a study entitled "Open Source Intelligence Testing Using the OWASP Version 4 Framework at the Information Gathering Stage (Case Study: X Company)". intelligence testing of company X. [11]. The seventh study was conducted by Revo Daniswara, Gusti Made Arya Sasmita and I Putu Agus Eka Pratama in a study entitled "The Testing for Information Gathering Using OWASP Testing Guide v4 (Case Study: Udayana University SIMAK-NG Application)" this study conducted penetration testing to the SIMAK application on the Udayana University Campus using the OWASP Testing Guide Version 4. [12]. The eighth study was conducted by Ade Kurniawan in a study entitled "Penerapan Framework OWASP dan Network Forensics untuk Analisis, Deteksi, dan Pencegahan Serangan Injeksi di Sisi Host-Based". [13]. The ninth research was conducted by Haerudin and Hermanto in a study entitled "Improvement Website Security System Using the OWASP Method". This study aims to improve system security using the OWASP method. [14]. The tenth study was conducted by Yunanri. W, Imam Riadi and Anton Yudhana in a study entitled "Analisis Deteksi Vulnerability Pada web server open journal system Menggunakan Owasp Scanner" this study conducted vulnerability detection on an open journal system web server using the OWASP Scanner. [15]. The eleventh study was conducted by Tikaridha Hardiani in a study entitled "Data Security Analysis with OWASP framework on the XYZ website". This study analyzed security data on the XYZ website using the OWASP framework. [16]. The twelfth study was conducted by Syarif Hidayatullah and Desky Saptadiaji in a study entitled "Penetration Testing pada Website Universitas ARS Menggunakan Open Web Application Security Project (OWASP)". This study analyzed security vulnerability on the ARS University website using the OWASP framework. [17].

4. Results and Discussion

In this section, the results and discussion of the research that has been carried out will be explained, namely conducting Penetration Testing at SISAKTI using the OWASP Testing Guide version 4.

4.1. Implementation of the OWASP Testing Guide method version 4

The implementation of the OWASP Testing Guide version 4 method is focused on 3 categories of testing, namely Testing for Information Gathering, Authorization Testing and Input Validation Testing. There are 28 test points which will be explained in the following table.

Table 1. Penetration Testing Results

Sub-test	Testing Activity	Tools	Result
Conduct Search engine discovery/ reconnaissance for information leakage (OTG-INFO-001)	Search for information that can be obtained from search results using the Google search operator "site"	Mozilla Firefox	The test was not successful
Fingerprint Web Server (OTG-INFO-002)	See the response header from the SISAKTI application using the inspect element feature in the browser	Mozilla Firefox	The test was successfully performed
Review Webserver Metafiles for Information Leakage (OTG-INFO-003)	Search for information from the robots.txt file	Mozilla Firefox	The test was not successful
Enumerate Applications on Webserver (OTG-INFO-004)	Perform Port Scanning to find an open ports	NMAP	The test was successfully performed
Review webpage comments and metadata for information leakage (OTG-INFO-005)	Looking for sensitive information in the HTML source	Mozilla Firefox	The test was not successful
Identify application entry points (OTG-INFO-006)	Checking the entry point of the website	Mozilla Firefox	The test was successfully performed
Map execution paths through application (OTG-INFO-007)	Spidering the website	Burpsuite	The test was successfully performed
Fingerprint Web Application Framework (OTG-INFO-008)	Looking for an information on error handling and session	Mozilla Firefox	The test was successfully performed
Fingerprint Web Application (OTG-INFO-009)	Checking on the HTTP header	Mozilla Firefox	The test was successfully performed
Map Application Architecture (OTG-INFO-010)	Analyzing website architecture	Mozilla Firefox	The test was successfully performed
Testing Directory traversal/file include (OTG-AUTHZ-001)	Testing directory access on the include file	Mozilla Firefox	The test was not successful
Testing for bypassing authorization schema (OTG-AUTHZ-002)	Bypassing user authorization	Mozilla Firefox	The test was successfully performed
Testing for privilege escalation (OTG-AUTHZ-003)	Manipulating user privileges	-	Testing cannot be carried out because there are

			prerequisites that are not met
Testing for insecure Direct Object References (OTG-AUTHZ-004)	Accessing unauthorized user data	Mozilla Firefox	The test was successfully performed
Testing for Reflected Cross-site scripting (OTG-INPVAL-001)	Inject the <button> tag that has the onclick attribute	Mozilla Firefox	The test was successfully performed
Testing for Stored Cross site scripting (OTG-INPVAL-002)	Testing url injection with the aim of accessing or changing other user data	Mozilla Firefox	The test was successfully performed
Testing for HTTP Parameter pollution (OTG-INPVAL-004)	Exploits vulnerability on the input parameters	Mozilla Firefox	The test was successfully performed
Testing for SQL Injection (OTG-INPVAL-005)	Testing vulnerabilities against SQL Injection	Mozilla Firefox dan Burpsuite	The test was successfully performed
Testing for LDAP Injection (OTG-INPVAL-006)	Testing the Lightweight Directory Access Protocol (LDAP)	-	Testing cannot be carried out because there are prerequisites that are not met
Testing for ORM Injection (OTG-INPVAL-007)	ORM Layer Testing	Mozilla Firefox	The test was successfully performed
Testing for XML Injection (OTG-INPVAL-008)	Testing vulnerabilities against XML Injection	-	Testing cannot be carried out because there are prerequisites that are not met
Testing for SSI Injection (OTG-INPVAL-009)	Testing vulnerabilities against SSI Injection	-	Testing cannot be carried out because there are prerequisites that are not met
Testing for XPath Injection (OTG-INPVAL-010)	Testing the vulnerability to XPATH Injtction	-	Testing cannot be carried out because there are prerequisites that are not met
Testing for IMAP/SMTP Injection (OTG-INPVAL-011)	Testing mail server vulnerabilities	-	Testing cannot be carried out because there are prerequisites that are not met
Testing for Code Injection (OTG-INPVAL-012)	Testing Local File Inclusion and Remote File Inclusion vulnerabilities	-	Testing cannot be carried out because there are prerequisites that are not met
Command Injection (OTG-INPVAL-013)	Inject OS Commands via HTTP requests	Mozilla Firefox dan Burpsuite	The test was not successful
Testing for Incubated Vulnerability (OTG-INPVAL-015)	Testing the file upload feature loophole	Mozilla Firefox dan Burpsuite	The test was successfully performed

Testing for HTTP Splitting/Smuggling (OTG-INPVAL-016)	Exploiting HTTP Headers	Mozilla Firefox	The test was not successful

Table 1 is the points that were 3 section of the test (Testing fo Information Gathering, authorization Testing and Input Validation Testing) successfully tested in this study and produces server information and details of the vulnerabilities found on the server. There were 15 tests that were successfully carried out and resulted in information on the vulnerabilities that existed in the system, there were 6 tests that were not successful because no vulnerabilities were found and there were 7 tests that could not be carried out because there were prerequisites that were not met.

4.2. Assessment using the CVSS Calculator

Based on the successful testing, an assessment will be carried out using CVSS to measure how risky the vulnerability is. Based on the 15 tests that were successfully carried out, 7 of them were the Testing for Information Gathering sub-test so that in this sub-test, the data obtained is only a reference for conducting deeper tests on Authorization Testing and Input Validation Testing. Based on the 8 points that were successfully tested and found vulnerabilities, there were 2 points from the Authorization Testing sub-test, namely OTG-AUTHZ-002 and OTG-AUTHZ-004, while the other 6 points were obtained from the Input Validation Testing sub-test, namely OTG-INPVAL 001, OTG-INPVAL-002, OTG-INPVAL-004, OTG-INPVAL-005, OTG-INPVAL-007 and OTG-INPVAL-015.

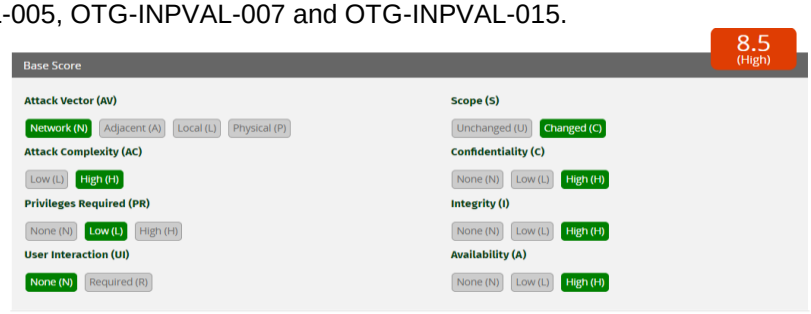

Figure 2. CVSS OTG-AUTGZ-002

Figure 2 is the result of the CVSS calculator calculation for OTG-AUTHZ-002. The attack vector chosen is Network because it is through a web application, Attack Complexity is chosen High because it requires in-depth information regarding the system to control attacks, Privileges Required is chosen Low because it requires login via SSO IMISSU as general privileges, User Interaction is selected None because it does not require user interaction to carry out the attack, Scope is chosen changed because the attack affects other parts besides the attack part, namely the page of another user, Confidentiality is selected High because there is information related to the system that can be seen and has a direct impact on the system, Integrity is selected High because it can modify files belonging to other users and Availability is selected High because it can damage the "availability" of other users. The final result of the OTG-AUTHZ-002 assessment is 8.5 (High).

Figure 3. CVSS OTG-AUTHZ-004

Figure 3 is the result of the CVSS calculator calculation for OTG-AUTHZ-004. The attack vector chosen is Network because it is through a web application, Attack Complexity is chosen High because it requires in-depth information regarding the system to control attacks,

p-ISSN: 2252-3006
e-ISSN: 2685-2411

Privileges Required is chosen Low because it requires login via SSO IMISSU as general privileges, User Interaction is selected None because it does not require user interaction to carry out the attack, Scope is chosen changed because the attack affects other parts besides the attack part, namely the page of another user, Confidentiality is selected High because there is information related to the system that can be seen and has a direct impact on the system, Integrity is selected High because it can modify files belonging to other users and Availability is selected High because it can damage the "availability" of other users. The final result of the OTG-AUTHZ-002 assessment is 8.5 (High).

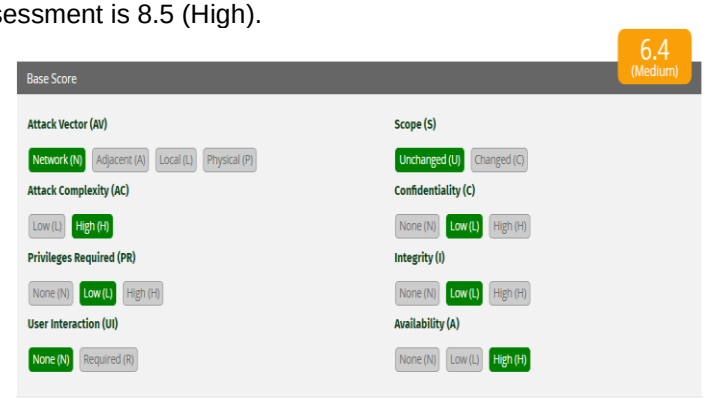

Figure 4. CVSS OTG-INPVAL-001, OTG-INPVAL-002 and OTG-INPVAL-004

Figure 4 is the result of the CVSS calculator calculation for OTG-INPVAL-001, OTG-INPVAL-002 and OTG-INPVAL-004. CVSS produces the same value because the results of the 3 sub-tests have the same impact on the system. The attack vector chosen is Network because it is through a web application, Attack Complexity is chosen High because it requires in-depth information regarding the system to control attacks, Privileges Required is chosen Low because it requires login via SSO IMISSU as general privileges, User Interaction is selected None because it does not require user interaction others to carry out the attack, Scope is chosen Unchanged because the attack does not affect other parts besides the attack part, Confidentiality is chosen Low because there is no information related to the system that can be seen but has a direct impact on the system, Integrity is chosen High because it can modify files belonging to other users and Availability High is chosen because it can damage the "availability" of both personal and other users' pages. The final result of the OTG-INPVAL -001, OTG-INPVAL-002 and OTG-INPVAL-004 was 6.4 (Medium).

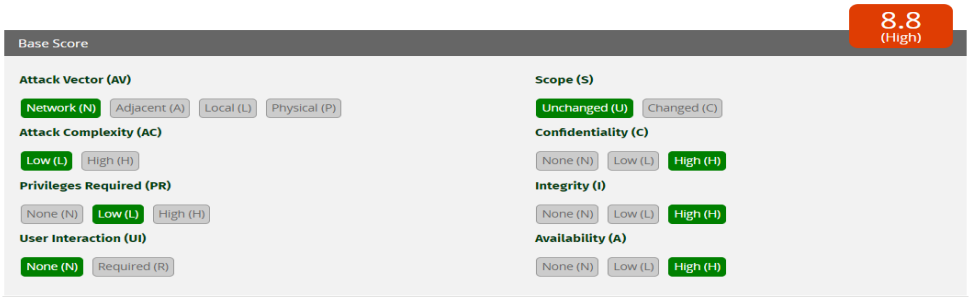

Figure 5. CVSS OTG-INPVAL-005

Figure 5 is the result of the CVSS calculator calculation for OTG-INPVAL-005. The attack vector chosen is Network because it is through a web application, Attack Complexity is chosen Low because it does not require in-depth information regarding the system to control attacks, Privileges Required is chosen Low because it requires logging in via SSO IMISSU as general privileges, User Interaction is chosen None because it does not require interaction another user to carry out the attack, Scope is chosen Unchanged because the attack does not affect other parts other than the part of the attack, Confidentiality is chosen High because there is information related to the system that can be seen and has a direct impact on the system, Integrity is chosen High because it can modify files belonging to other users, and Availability was chosen High because it can damage the "availability" of both personal and other users' pages. The final result of the OTG-INPVAL-005 assessment was 8.8 (High).

p-ISSN: 2252-3006
e-ISSN: 2685-2411

Figure 6. CVSS OTG-INPVAL-007

Figure 6 is the result of the CVSS calculator calculation for OTG-INPVAL-007. The attack vector chosen is Network because it is through a web application, Attack Complexity is chosen High because it requires in-depth information regarding the system to control attacks, Privileges Required is chosen Low because it requires login via SSO IMISSU as general privileges, User Interaction is selected None because it does not require user interaction to carry out the attack, Scope is chosen changed because the attack affects other parts besides the attack part, namely the page of another user, Confidentiality is selected High because there is information related to the system that can be seen and has a direct impact on the system, Integrity is selected High because it can modify files belonging to other users, and Availability is selected High because it can break the "availability" of other users. The final result of the OTG-AUTHZ-002 assessment is 8.5 (High).

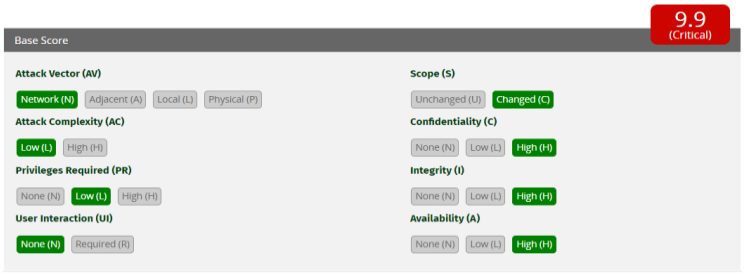

Figure 7. CVSS OTG-INPVAL-015

Figure 7 is the result of the CVSS calculator calculation for OTG-INPVAL-015. The attack vector chosen is Network because it is through a web application, Attack Complexity is chosen Low because it does not require in-depth information regarding the system to control attacks, Privileges Required is chosen Low because it requires logging in via SSO IMISSU as general privileges, User Interaction is chosen None because it does not require interaction another user to carry out the attack, Scope is chosen changed because the attack affects other parts besides the part of the attack, namely another user's page, Confidentiality is selected High because there is information related to the system that can be seen and has a direct impact on the system, Integrity is selected High because it can modify the user's files others, and Availability is chosen High because it can break the "availability" of other users. The final result of the OTG-INPVAL-015 assessment is 9.9 (Critical).

4.3. Improvement Recommendations

The information known in the Information Gathering is not a vulnerability but to determine the scope of testing at a later stage, while tests whose results are related to configuration are found on OTG-INFO-001, OTG-INFO-003, and OTG-INFO-005 but the results are negative, because Therefore, this section does not provide recommendations on system configuration. Recommendations for the Authorization Testing OTG-AUTHZ-002 and OTG-AUTHZ-004 sub-tests, OWASP Testing Guide version 4 does not provide specific recommendations to overcome this weakness, but provides suggestions for creating authorization scenarios with predefined access privileges. From the test, it is known that the SKP delete function that uses the id_n parameter can be modified with other user data, a suggested solution is to encrypt the URL.

Recommendations for the Validation Testing sub-test, namely OTG-INPVAL-001 and OTG-INPVAL-002, OWASP Testing Guide version 4 provides recommendations for using XSS

character validation. too big but it is possible to steal session information and cookies or burden application performance. The reflected XSS gap is found in the onclick attribute that is used, therefore it is better not to use the onclick attribute to execute a certain command, using javascript is better. OTG-INPVAL-004 test results, OWASP Testing Guide version 4 does not provide specific recommendations to overcome this weakness but provides advice not to use the same variable value repeatedly in parameters.

OTG-INPVAL-007 test results, OWASP Testing Guide version 4 provides recommendations not to use values that are directly connected to the database or have the risk of being modified in the Object Relational Model (ORM). Based on the test, it is known that the delete SKP function has an id_n parameter which, if changed, will delete another user / SKP data that has that id_n. This means that it is necessary to build an authorization system that can validate this, for example, there is a validation of authorization via data cookies before a command is executed, so that even if a value is modified, if it is not the value of the authorized user's rights then the function will not be executed, other than that Encrypting value scripts is also an option that can be implemented.

For attacks like OTG-INPVAL-007, OWASP Testing Guide version 4 provides recommendations not to use values that are directly connected to the database or have the risk of being modified in the Object Relational Model (ORM), but for SQL Injection cases, OWASP Testing Guide version 4 provides recommendations to make validation in PHP, for example using the function myphp_real_escape_string($value); which is a PHP function that will convert the value of a variable into a string that the query can accept as an actual string. There are problems with this method, namely that it does not solve the problem if the SQL Injection attack is carried out by utilizing the file upload vulnerability (OTG-INPVAL-015), therefore OTG-INPVAL-015 must be repaired. OWASP Testing Guide version 4 provides recommendations for verifying the type of uploaded content, generally programmers will use the mime type checking function, as in the SISAKTI application, but of course, there are still gaps that can be exploited. The author recommends creating your own verification logic, validation is made by encrypting the file name and changing the content type (file extension) before saving it.

4.4. Review Recommendations for Improvement

The improvements that have been made by the Udayana University USDI team are in the form of adding input validation to the SKP file upload function and encryption at the file storage location. The author re-examines the results of implementing the recommendations for improvement, namely OTG-AUTHZ-002, OTG-AUTHZ-004, OTG-INPVAL-001, OTG-INPVAL-002, OTG-INPVAL-004, OTG-INPVAL-005, OTG-INPVAL-007, and OTG-INPVAL-015..

Table 2. Comparisson Penetration Testing Results

Sub-Test	Penetration Testing Before Improvement	Penetration Testing After Improvement
OTG-AUTHZ-002	The test was successfully performed	The test was not successful
OTG-AUTHZ-004	The test was successfully performed	The test was not successful
OTG-AUTHZ-001	The test was successfully performed	The test was successfully performed
OTG-AUTHZ-002	The test was successfully performed	The test was successfully performed
OTG-INPVAL-004	The test was successfully performed	The test was not successful
OTG-INPVAL-005	The test was successfully performed	The test was not successful
OTG-INPVAL-007	The test was successfully performed	The test was not successful
OTG-INPVAL-015	The test was successfully performed	The test was not successful

Based on the results of retesting the 8 points on the Table 2, leaving only 2 points, namely OTG-INPVAL-001 and OTG-INPVAL-002 which still have vulnerabilities and have not been repaired according to the recommendations given

5. Conclusion

Based on the tests that have been carried out, all the objectives of this research have been achieved, namely applying the OWASP Testing Guide version 4 to carry out Penetration Testing on the SISAKTI system, knowing the results of penetration testing using the OWASP Testing Guide version 4 on the SISAKTI system, assessing how risky the vulnerabilities found on the system are. SISAKTI, provided recommendations to USDI Udayana University to make improvements to the system and also retested the SISAKTI system after the recommendations were implemented so that it only resulted in minor risks after the system was repaired and was the main objective of this research.

References

[1] Dirgahayu, R. T., Prayudi, Y. & Fajaryanto, A., 2015. Penerapan Metode ISSAF dan OWASP versi 4 Untuk Uji Kerentanan Web Server. Jurnal Ilmiah Nero, Volume 190-197.

[2] EC-Council, 2018. Certified Ethical Hacker Module 01. London: IPSpecialist LTD.

[3] Mustaqbal, M. S., 2015. Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus: Aplikasi Prediksi Kelulusan SNMPTN). *Jurnal Ilmiah Teknologi Informasi Terapan (JITTER),* Volume I(3), pp. 31-36.

[4] Juhad, Hilal Afrih and Isnanto, R. Rizal and Widianto, Eko Didik (2016) *Analisis Keamanan pada Aplikasi Her-registrasi Online Mahasiswa Universitas Diponegoro.* Jurnal Teknologi dan Sistem Komputer, 4 (3). pp. 479-484. ISSN 2338-0403

[5] Meuuci, et al., OWASP Testing Guide v4, 2014).

[6] Dewanto, A. P., 2013. Penetration Testing Pada Domain uii.ac.id Menggunakan OWASP 10, Yogyakarta: Universitas Islam Indonesia.

[7] Purnawan, A., 2014. Studi dan Implementasi Keamanan Website Menggunakan Open Web Application Security Project (OWASP) Studi Kasus : PLN Batam, Bandung: Universitas Pasundan.

[8] Fajaryanto, A. & Muhsin, M., 2015. Penerapan Pengujian Keamanan Web Server Menggunakan Metode OWASP Versi 4 (Studi Kasus Web Server Ujian Online). *Multitek Indonesia,* pp. 31-41.

[9] Fajaryanto, Prayudi & Dirgahayu, 2015. Penerapan Metode ISSAF dan OWASP versi 4. *Jurnal Ilmiah NERO,* Volume 1, p. 191."

[10] Cobantoro, A. F., n.d, 2016. Penerapan OWASP Versi 4 Untuk Uji Kerentanan Web Server (Studi Kasus E-Jurnal Server Kampus X Madiun). *Seminar Nasional Telekomunikasi dan Informatika (SELISIK),* Volume 74-79.

[11] I Putu Agus Eka Pratama, Anak Agung Bagus Arya Wiradarma, "Open Source Intelligence Testing Using the OWASP Version 4 Framework at the Information Gathering Stage (Case Study: X Company)", International Journal of Computer Network and Information Security(IJCNIS), Vol.11, No.7, pp.8-12, 2019.

[12] Daniswara, R., Sasmita, G., & Pratama, I. (2020). The Testing for Information Gathering Using OWASP Testing Guide v4 (Case Study : Udayana University SIMAK-NG Application). JITTER : Jurnal Ilmiah Teknologi Dan Komputer, 1(1), 23-33.

[13] Kurniawan, A. (2020). Penerapan Framework OWASP dan Network Forensics untuk Analisis, Deteksi, dan Pencegahan Serangan Injeksi di Sisi Host-Based. Jurnal Telematika, 14(1), 9-18.

[14] Haeruddin, H., & Hermanto, H. (2022). IMPROVEMENT WEBSITE SECURITY SYSTEM USING OWASP METHOD. CoMBInES - Conference On Management, Business, Innovation, Education And Social Sciences, 2(1), 199.

[15] W, Y., Riadi, I., & Yudhana, A. (2018). "Analisis Deteksi Vulnerability Pada web server open journal system Menggunakan Owasp Scanner", Jurnal Rekayasa Teknologi Informasi (JURTI), 2(1), 1.

[16] Hardiani, T. (2022) Data Security Analysis with OWASP framework on website XYZ, CYBERNETICS. Vol.6, No.1.

[17] Hidayatulloh, S., & Saptadiaji, D. (2021). Penetration Testing pada Website Universitas ARS Menggunakan Open Web Application Security Project (OWASP). Jurnal Algoritma, 18(1), 77-86.