Vulnerability Detection with K-Nearest Neighbor and Naïve Bayes Method using Machine Learning

Herman [a,1],, Imam Riadi [b,2,] Yudi Kurniawan,[3*]

[a,c] Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta-55164, Indonesia
[b] Department of Information System, Universitas Ahmad Dahlan, Yogyakarta-55614, Indonesia
[1]hermankaha@mti.uad.ac.id, [2]imam.riadi@is.uad.ac.id, [3]yudi2007048008@webmail.uad.ac.id
* corresponding author

ARTICLE INFO

Article history
Received 27 Feb 2023
Revised 01 Apr 2023
Accepted 10 May 2023

Keywords
Security
Website
SQL Injection
Naïve Bayes
K-Nearest Neighbor

ABSTRACT

In this day and age, the use of the Internet has increased. SQL injection is a serious security threat on the Internet for various dynamic websites. As the use of the Internet for various online services increases, so make the security threats that exist on the Web. SQL injection attacks are one of the most serious security vulnerabilities on the Web. Most of these vulnerabilities are caused by a lack of input validation and the use of SQL parameters. SQLMap is an application from the Kali Linux operating system tha is useful for injecting data on a website by using the features available in this application. In this paper, author conducts a security assessment to detect attacks on a website, more precisely to detect SQL Injection attacks,using the K- Nearest Neighbor method and naïvbayes. The results obtained are that the website being tested has SQL Injection vulnerabilities, and the K-Nearest Neighbor method is the best method for this case because it has an accuracy of 94.2%. In comparison, the Naïve Bayes method has an accuracy of 80%.

1. Introduction

Databases as data storage media in an information system certainly have a very important role from the aspect of data privacy and usefulness in the completeness of the features of an information system, along with the development of technology, a database can no longer only be accessed via a server/localhost, but can also be accessed through a global computer network that is interconnected and can be accessed remotely with the use of internet services [1].

In its development, data security has become an important part that cannot be separated in the implementation of an information system. The database as a data storage medium in the information system is ensured to have good security in order to maintain data privacy and the usefulness of the information system [2]. Data must be protected from all forms of possible threats by hackers who do not have access legally by taking preventive measures, such as penetration testing, which can simply be interpreted as a method of evaluating and testing the security of a computer system and network, including those relating to security. data[3].

This study discusses the security testing technique with the SQL injection method, which is a hacking technique with a focus on testing the database as a data storage medium, with the aim that website developers are more aware of the dangers of website vulnerabilities, and prevent data leakage [4].

The security test is carried out using a computer that uses the Windows operating system, the vega program as a tool to find vulnerabilities in a website, and SQLmap as a tool for penetration testing, to ensure that the website is vulnerable or not. The purpose of this research is to look for

10.29099/ijair.v7i1.795

website vulnerabilities, and make a report to the website developer, that the website is vulnerable to SQL injection so that the data in the database can be exposed. Therefore, it is hoped that from this research the developer will be more aware of existing vulnerabilities, and improve the structure of the website so that it is more secure from SQL injection attacks.

The purpose of this study is to find the vulnerability of a website and use the vulnerability to get data on the website, and this is done so that the website knows that a big/small vulnerability can cause information leakage of website user data, an irresponsible party can misuse the website user information data, therefore, this research was conducted, so that the website developer knows the vulnerabilities that exist on their website.

2. Method

2.1. Previous Research

In 2020 Abdul Djalil Djayali conducted a similar study on a web server for filling out study plan cards, digital forensic analysis is an action that forensic investigators must take to find out the source of an attack that occurs on a server that is being handled. The increasing use of the internet in the era of the industrial revolution 4.0 in higher education online services, compared to the security threats faced. SQL Injection attack is one of the web exploitation techniques that are quite old, but until now, the technique is still quite capable and effective. Most of these vulnerabilities are caused by the lack of input validation and the use of parameters in SQL queries. In this paper, an example of a SQL Injection attack will be given on the online study plan card (KRS) charging server. The conclusion from the tests carried out is that there is a serious vulnerability in the login parameters for student accounts and study programs where SQL Injection attacks with the comments method are tested nine times on the login page and can be bypassed. The numerical method using SQL also obtained data from the entire database contained in the web server. An IP blocklist must be done on the attacker's IP to apply the Digital Forensic Readiness Index (DiFRI)[5].Andria and Ridho Pamungkas also conducted similar research in 2020. The difference is that the research was carried out on the Android operating system using Termux. The results obtained in this research are that there is a security hole or vulnerability that allows for exploitation by hackers/hackers. It can display and access the database structure contained in the web server [6].

2.2. Research Steps

In this study, there are several research steps to be carried out. These steps can be seen in Fig. 3.

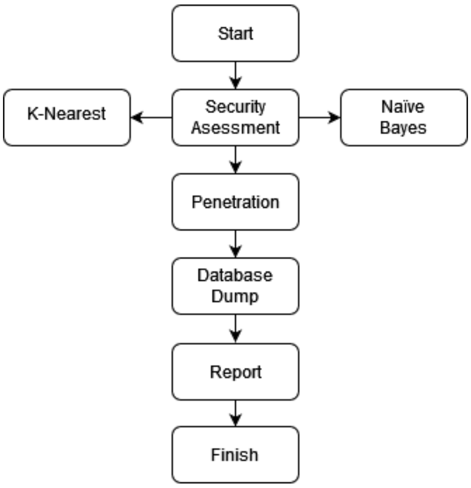


Fig. 1.SQL Injection Penetration Testing Research Steps

The first step is to conduct a vulnerability assessment of the website using the vega tools. The function of this stage is to find a website's vulnerability and exploit it to get sensitive data on a

website. Then if there are vulnerabilities, penetration testing will be carried out using SQL Map. 26] To confirm the vulnerability if the website successfully exploits the website, the expected result is that the author can view and access the website database. The next step is to extract data from the website database as evidence that the website has vulnerabilities that can be exploited. Attackers can access the information in the website database, the results of the dump information in the database are stored in Microsoft excel form, and the last step is to make reports and suggestions for website developers regarding website vulnerabilities. Yes, so there is a possibility that the data on the website database is scattered

2.3. SQL Injection

SQL injection is a technique for exploiting security vulnerabilities at the database layer of an application [7]. This cybercrime threat can occur because of inputs that are not properly filtered in its manufacture, so it creates a loophole that can be abused [8], an example of which can be seen in Fig. 1

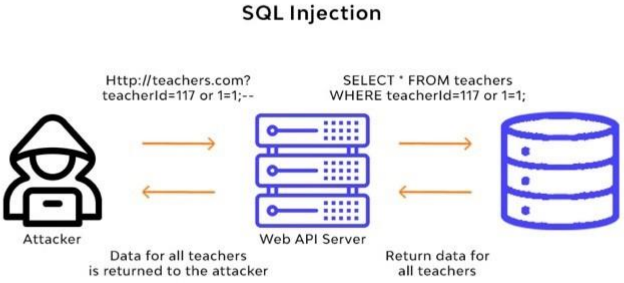


Fig. 2. How SQL Injection Works

In the absence of a filter on the website url, most likely, the website is vulnerable to SQL injection[9]. For example can be seen in Figure 1, the address www.example/id.php?=1, behind the url address, there is a GET method, meaning that the url is requesting access to the database with the syntax, "Select * from example where id =1, here the database returns a value, to display all tables with id = 1, therefore, if the attacker knows the structure of the website's database, then the attacker can retrieve all existing data in the database of the attacked website [10]. Every year the attack with the highest percentage is SQL injection[11], which can be seen in Fig. 2

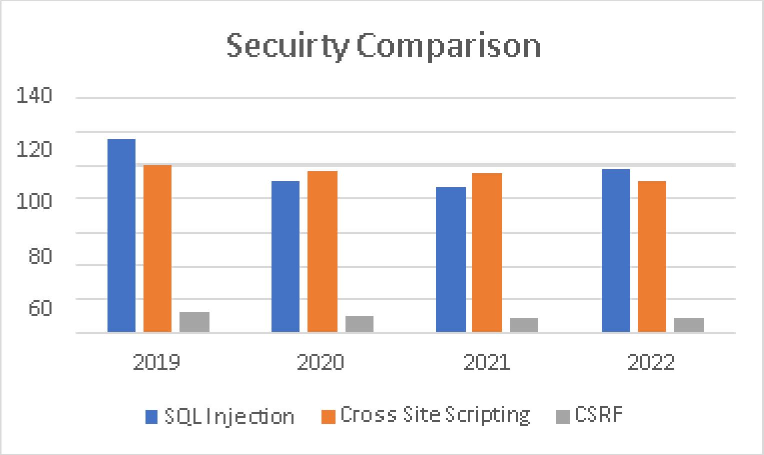


Fig. 3. Security Comparison Each year

From 2019 to 2022, the highest vulnerability on a website is SQL/SQL injection[12]. This vulnerability is very dangerous for website developers because attackers can retrieve data in the website database. In the second position, there is a vulnerability in Cross Site Scripting/XSS, and in the third, there is a vulnerability in CSRF tokens.

2.4. SQLMap

SQLmap is a tool used to perform penetration testing on a website by exploiting the variables contained on a website, either sent via POST or GET methods. This hacking technique that uses SQLmap is called SQLInjection[13]. POST method will send the variable value to the server separately, so the variable value is not visible[14], and GET it is a method of sending data using a query string, so all values in the form will be sent to the server/file side, and the values from you form will appear in the URL line/Address bar[15]. SQL Map is also an open-source tool which detects and exploits SQL Injection. By performing SQL Injection attacks, an attacker can take over and manipulate a database on the server.

2.5. Security Assessment / Testing

Security testing is a process to find security vulnerabilities in software or application. In it, there will be various types of tests to ensure that the system you are developing is completely safe from various threats of cyber attacks[16]. There are several types of security testing, first one is vulnerability Scanning is a security test carried out through automated software to scan a web application to look for vulnerabilities such as SQL Injection, Cross Site Scripting, and other vulnerabilities [17], second is security scanning is a scan used to find vulnerabilities or unwanted file modifications in web-based applications, websites, networks, or file systems, third is penetration testing is a testing process by simulating a cyber attack on the system to be tested. This test will be carried out manually by a professional and certified pentester using various pentest tools and techniques [8], fourth is risk assessment through risk assessment, security risks faced by applications, software, and networks will be identified and analyzed. The security risks will then be classified into several categories, namely high, medium, and low[18], fifth security auditing is a structured method for evaluating security measures within a company[19], next one is ethical hacking is a security test carried out using all hacking techniques and other related computer attack techniques[20]. This testing process is carried out by ethical hackers who have obtained permission to explore the company's IT infrastructure more broadly, and last is posture assessment refers to the methodology used to improve risk management capabilities in companies. Posture Assessment combines several types of security testing, namely Ethical Hacking, Security Scanning, and Risk Assessment[21].

2.6. Database

A database is a collection of data or information stored systematically. Databases are important tools to collect information, data, or files in an integrated manner [22]. There are several types of databases, the following types of database types is, first an operational database is also known as On- Line Transaction Processing. This type of database functions as a container for managing dynamic data in real-time or directly [23], second Database Warehouse This database is often used to perform data analysis and reporting. Database warehouse is considered a core component of business intelligence[24], third Distributed Database differs from a parallel system that is closely connected and has a single data system. This Database is not installed on a computer or similar device. This system is distributed through an incorporated site with no physical components [25], fourth is Relational Database organizes data based on a data relationship model. Many software tools use these relational databases to organize and maintain information through each data relationship, and last is End User Database This database is developed by end-users through their workstations. Various types of data files are created by themselves with a certain procedure. Examples include spreadsheets, word processing, and file downloads. Several attacks can be used on databases. First, namely DDoS attacks, DDoS attacks are the easiest to install and the most destructive, but over time these attacks have been overcome efficiently. Some cloud providers have overcome cloud infrastructure to prevent or reduce this attack. However, some solutions have yet to be able to detect all possible attacks perfectly. The purpose of this attack is to prevent users from enjoying the services provided by the server secondly, ransomware attacks. Ransomware is malware that can take systems hostage, most often by encrypting or stealing data and performing extortion. The observed ransomware attacks targeted vulnerabilities in the MongoDB database and, most recently, SQL Injection. This research uses penetration testing techniques to focus on SQL Injection attacks on the Operational Database.

2.7. Naïve Bayes

Naive Bayes is a suitable method for binary and multiclass classification. This method, also known as Naive Bayes Classifier, applies supervised object classification techniques in the future by assigning class labels to instances/records using conditional probabilities. Conditional probability is a measure of the chance of an event occurring based on other events that have (assumed, presumed, asserted, or proven) occurred. The term supervised refers to the classification of training data that has been labeled with a class. For example, a fraudulent transaction has been flagged as transactional data. Then, if you want to classify future transactions as fraudulent/non-fraudulent, that classification will be referred to as supervised.The formula for the Naïve Bayes theorem can be seen in equation 1.

$$Probability = P(A|B) = P(B|A)P(A)P(B)$$

With the equation above, it is explained that P(A) is the probability that A will occur, P(B) is the probability that B will occur, and P(A|B) is the probability that A occurs with evidence that B has occurred, while the probability that B has occurred with evidence that A has occurred.K-Nearest Neighbor.

2.8. K-Nearest Neighbor

The Nearest Neighbor algorithm can classify the similarity of a test data with other test data based on the closeness value of an attribute. Table 1 is a SQL Injection attack that can be measured by its weight or value.

Table 1. Classification For K-Nearest

Attribute	Weight
Parameter	0,80
SQL Error	0,70
Filtering	0.60

This research evaluates SQL Injection security vulnerabilities in a web-based system or application. Security vulnerabilities in SQL Injection in each case of a cyber attack can be learned from previous cases. The pattern of security holes in SQL Injection can be calculated using the closeness between the previous attack cases and the current attack. The equation for calculating the proximity of the K- Nearest algorithm can be seen in equation 2.

$$K - Nearest(T,S)\frac{\sum i = 1\ F(Ti,Si) * Wi}{Wi}$$

Equation 2 is the formula for the distance between the two cases of SQL Injection attacks, T is the potential for the new attack, S is the previous attack, with W is the weight value of the K-Nearest attribute used. Based on Equation 2 and the SQL Injection attributes that have been formulated, a formula can be made or the possibility of an attack occurring. The formula can be seen in equation 3.

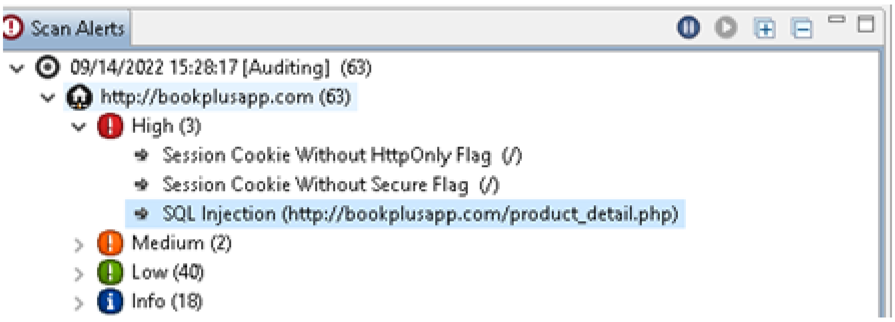
$$Accuracy = \frac{(A + B)F(C + D)F(E + F)}{BFDFF}$$

In equation 3, A is the closeness value of the parameter detection attribute, B is the weight of the parameter, C shows the closeness of the SQL Error, D is the weight of the SQL Error, E shows the closeness value of the filtering attribute and F is the weight of the filtering attribute.

3. Results and Discussion

Assessment Vulnerability Using Vega

In this trial, the website used is www.bookplusapp.com. The results of the vulnerability assessment using vega can be seen in Fig. 4.



Scan Alerts

09/14/2022 15:28:17 [Auditing] (63)
http://bookplusapp.com (63)
High (3)
Session Cookie Without HttpOnly Flag (/)
Session Cookie Without Secure Flag (/)
SQL Injection (http://bookplusapp.com/product_detail.php)
Medium (2)
Low (40)
Info (18)

Fig. 4. Vulnerability Assessment Results Using Vega

Vulnerability analysis on the website www.bookplusapp.com there are three vulnerabilities at a high level. Here the author only takes the vulnerability to SQL injection. The vulnerability that exists on the website is at the url www.bookplusapp.com/product_detail.php?id=9 with the GET method, meaning that the url is requesting a request to the database, The attacker may think that there is an id column in the product_detail PHP file. For example, there is a book product about SQL Injection, which is stored with id = 9, along with other information about the book. The book information is stored in the product_detail.php file, where this file will be stored and forwarded to the database. In this case, the attacker does not know the structure of the website's database. Therefore the attacker can take advantage of the PHP file parameters because this PHP file functions to pass on the information that is then stored in the database. To find the name of the database, along with the table where the information is stored, the attacker uses the id parameter in the product_detail.php file, where this parameter is the primary key in a database table. To find out the database structure of the website, use the SQL Map tool, where this tool functions to find the database structure of a website, by utilizing the SQL Injection vulnerability parameter on a website.

Detection Using K-Nearest

The results of the K-Nearest algorithm use 50 datasets, where later, this data will be trained with the help of machine learning. The results of the training data can be seen in table 2.

Table 2. Detection K-Nearest

Value	K-Nearest(website)
Parameter	1,00
Parameter Weight	0,80
SQL Error	1,00
SQL Error Weight	0,70
Filtering Detection	0,80
Filtering Detection Weight	0,60

The results of table 2 get the Nearest or predetermined weight, the results of the training data get the weight in the parameter category of 0.80, the weight in SQL Error is 0.70, and the weight in Filtering detection is 0.60. From the results of table 2, it can be calculated the potential for attack/accuracy in the K-nearest Neighbor method. To calculate the potential for SQL Injection, equation 3 can be used.

$$Accuracy = \frac{\mathrm{K, LM}}{N, K} * 100\%$$

Accuracy = 94,2%

The accuracy obtained using the K-Nearest Neighbor method or algorithm is 94.2%

Detection Using Naïve Bayes

Detection using naïve Bayes using machine learning k = 2 with SQL injection data set can be seen in Table 3.

Table 3. Classification Naive Bayes

No	CommentSQL	Operator	LogicalSQL	Keyword SQL
1	--, #, -+,	<, >, ==,	OR,	UNION,
	++, --, -“,	!=, <=,	AND,	SELECT,
	/*, */, /**/	>=, <<,	NOR,	ORDER,
		>>, &&	XOR	INFORMATION
				SHCEME,
				INSERT,
				UPDATE,
				DELETE,
				FROM
				DATABASE,
				WHERE ,COLUMN,NAME

Table 3 is the category used to create data sets, which will later be used to train data using machine learning. The data sets used are 50, and k = 2. The results of the data training using Naïve Bayes can be seen in Table 4

Table 4. Result Classification Naive Bayes

No	Data set	Constant	Classification
1	50	K >= 3	40
2	50	K < 3	10

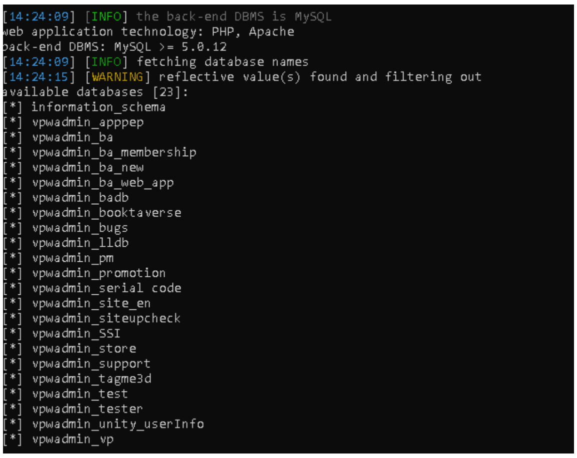
The results in table 4 can be obtained by using a dataset of 50, the SQL Injection classification that successfully detects SQL Injection vulnerabilities is 40, and those that do not include SQL Injectionare 10. Furthermore, to find naïve Bayes accuracy, you can use equation 4

$$Probability = \frac{Number\ of\ success\ attacks\,(prob)}{number\ of\ attacks\,(atempt)}$$

Equation 4 is used to find the accuracy/probability of SQL Injection using naïve Bayes, with 50 trials with naïve Bayes classification K >= 3, and the result is 80%

3.4 Penetration

After getting the results using Naïve Bayes and K-nearest, then doing penetration testing to validate the vulnerabilities of the results of the methods used, penetration testing using SQL map tools, the results can be seen in Figure 5



```
[14:24:09] [INFO] the back-end DBMS is MySQL
web application technology: PHP, Apache
back-end DBMS: MySQL >= 5.0.12
[14:24:09] [INFO] fetching database names
[14:24:15] [WARNING] reflective value(s) found and filtering out
available databases [23]:
[*] information_schema
[*] vpwadmin_apppep
[*] vpwadmin_ba
[*] vpwadmin_ba_membership
[*] vpwadmin_ba_new
[*] vpwadmin_ba_web_app
[*] vpwadmin_badb
[*] vpwadmin_booktaverse
[*] vpwadmin_bugs
[*] vpwadmin_lldb
[*] vpwadmin_pm
[*] vpwadmin_promotion
[*] vpwadmin_serial code
[*] vpwadmin_site_en
[*] vpwadmin_siteupcheck
[*] vpwadmin_SSI
[*] vpwadmin_store
[*] vpwadmin_support
[*] vpwadmin_tagme3d
[*] vpwadmin_test
[*] vpwadmin_tester
[*] vpwadmin_unity_userInfo
[*] vpwadmin_vp
```

Fig. 5. Penetration Testing Using SQLMap

From the penetration testing results, we get database information on the website being tested. Thereare a total of 23 databases on the website.

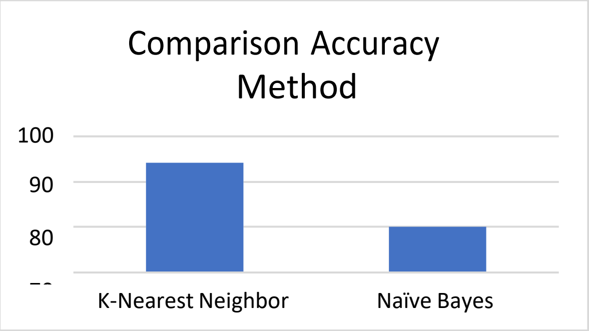


Fig. 6. Comparison Accuracy

Can be seen in Figure 6 the comparison of accuracy between the K-Nearest Neighbor and Naïve Bayes methods. K-Nearest Neighbor has an accuracy of 94.2%, while Naïve Bayes has an accuracy of 80%

4. Conclusion

From the results of experiments conducted to find SQL Injection vulnerabilities in the URL of a website, two algorithms are used to detect url parameters that may have vulnerabilities. The first uses the K-Nearest Neighbor algorithm, which has an accuracy of 94.2%, and the second method, the naïve Bayes method, is used, with the help of machine learning to facilitate research. The naïve Bayes method detects SQL injection vulnerabilities with an accuracy of 80%. In this study, the best method is the K-Nearest Neighbor method because it has a higher accuracy of 94.2%. If other researchers want to do the same trial, the authors suggest increasing the test data and data training used so that accuracy is better.

Acknowledgment

The Author want to aknowldge the faculty of Industrial Enggineering , Universitas Ahmad Dahlan, Yogtakarta, Indonesia.

Declarations

Author contribution. The contribution or credit of the author must be stated in this section.
Conflict of interest. The authors declare no conflict of interest.

Data and Software Availability Statements

Data and Software availability where data and software supporting the results reported in a published article can be found, including hyperlinks to publicly archived datasets and software analyzed and generated during the study/experiments.

References

[1] B Kusnandar, "Pengguna Internet Indonesia Peringkat ke-3 Terbanyak di Asia," https://databoks.katadata.co.id/datapublish/2021/10/14/pengguna-internet-indonesia peringkat-ke-3-terbanyak-di-asia, 2021.

[2] S. Mirdula and D Manivannan, "Security Vulnerabilities in Web Application - An attack prepective," 2013.

[3] Worang and E. Sutanta, Sistem Basis Data. Yogyakarta: Graha Ilmu, 2004.

[4] W. G. J. Halfond and A. Orso, "Detection and Prevention of SQL Injection Attacks," 2013.

[5] Abdul Djalil Djayali, "Analisa Serangan SQL Injection pada server pengisian Kartu Rencana Studi(KRS) Online," 2020.

[6] Andria and Pamungkas Ridho, "Penetration Testing Database Menggunakan Metode SQL Injection Termux," 2020.

[7] A Raharja, "Analisis Kerentanan pada Aplikasi E-Voting Menggunakan OWASP Framework," 2019.

[8] S. Mohammad, S. Sajjadi, and B. T. Pour, "Study of SQL Injection Attacks and Countermeasures," vol. 2, 2013.

[9] K Pertiwi, "Analisa Keamanan Website Dari Serangan Cross Site - Scripting (XXS) Menggunakan Framework OWASP," 2019.

[10] Hermawan Rudi, "Teknik Uji Penetrasi Web Server Menggunakan SQL Injection dengan SQLMap di Kali Linux," vol. 6, 2021.

[11] Ramansyah, Prayudi Yudi, and Riadi Imam, "Deteksi Bukti Digital Game Online Pada Platform Skyegrid Menggunakan Framework FRED," JATISI, vol. 8.

[12] Yunanri, Riadi Imam, and Yudhana Anton, "Analisis Keamanan Webserver Menggunakan Metode Penetrasi Testing (PENTEST)," vol. 2.

[13] Ula Muhammad, "Evaluasi Kinerja Software Web Penetration Testing," vol. 11, 2019.

[14] Sunyoto A and Pramono Edi, "Deteksi Serangan SQL Injection Menggunakan Hidden Markov Model," vol. 5, 2021.

[15] P Sitorus and A Habibi, "Teknik Pencegahan Penetrasi SQL Injeksi Dengan Pengaturan Input Type Number dan Batasan Input Pada Form Login Website," vol. 4, 2020.

[16] "Cyber Security Assessment," www.itgid.org, Apr. 19, 2022.

[17] M. Ign, S. H. Muhammad, S. Hoga, and R. Abd Aedah, "Web Vulnerability Asessment and Maturity Model Analysis on Indonesia Higher Education," Procedia Computer Science 161 (2019) 1165-1172

[18] V. Syamsudha, A. R. Syed, and E. Gayatri, "The Solution of SQL Injection Vulnerability in Web Application Security," 2019.

[19] Al Azhar Muhammad, "Digital Forensic Panduan Praktis Investigasi Komputer, Jakarta: Salemba Infotek," 2012.

[20] C Palmer, "Ethical Hacking," vol. 40, 2001.

[21] G Mahendra, "Penetration Testing Menggunakan Framework ISSAF Dan OWASP Pada Aplikasi Desa Digital Diskominfo Kabupaten Gianyar," vol. 4, 2021.

[22] S. S. Ardiansyah, S. Raharjo, and J. Triyono, "Analisis Keamanan Serangan Sql InjectionBerdasarkan Metode Koneksi Database," vol. 4, 2016.

[23] Warman, Indra, and Ramdaniansyah Rizki, "Analisis Perbandingan Kinerja Query Database Management System (DMS) antara Mysql 6.7.16 dan MariaDB 10.1," vol. 6, 2018.

[24] Halib, bin Badaruddin, Edy Budirman, and Hario Jati Setyadi, "Teknik Hacking Web Server Dengan SQLMap di Kali Linux," Jurti, vol. 1, 2017.

[25] Lika Sudirhayanto, Halim Putra Dwi Roy, and Verdian Ihsan, "Analisa Serangan SQL Injeksi Menggunakan SqlMap," 2018.

[26] [26] A. Rico Agarta, "Analisa Keamanan Website Pada Universitas Gunadarma Terhadap Serangan SQL Injection," 2021.

[27] P. Singh, K. Thevar, and B. Shaikh, "Detection of SQL Injection and XSS Vulnerability in Web Application," 2015.

[28] Riadi Imam, "Log Analysis Techniques using Clustering in Network Forensic," vol. 10, 2012.

[29] Ariyus Doni, Computer Security. Yogyakarta: Andi, 2006.

[30] Z.S. Alwan and M. F. Younis, "Detection and Prevention of SQL Injection Attack : A Survey," vol. 6, 2017