IJOICT
ISSN 2460-9056
http://socj.telkomuniversity.ac.id/ijoict/

Intl. Journal on ICT
Vol. 5, Issue. 2, December 2019. pp. 1-10
doi: 10.21108/IJOICT.2019.52.268

Study the Best PenTest Algorithm for Blind SQL Injection Attacks

Aldebaran Bayu Nugroho[1], Satria Mandala[1*]

[1]*School of Computing, TelkomUniversity*

Bandung, Indonesia

[*] satriamandala@telkomuniversity.ac.id

Abstract

There are several types of SQL injection attacks. One of the most popular SQL Injection Attacks is Blind SQL. This attack is performed by exploiting a gap in the database server when executing query words. If the server responds to an invalid query, the attacker will then reverse the engineering part of the SQL query, which is obtained from the error message of the server. The process of generating a blind SQL injection attack is complicated. As a result, a Pentester often requires a long time to penetrate the database server. This research provides solutions to the problems above by developing the automation of a blind SQL injection attack. The method used in this research is to generate keywords, such as the database name and table name so that the attacker can retrieve information about the user name and password. This research also compares several search algorithms, such as linear search, binary search, and interpolation search for generating the keywords of the attack. Automation of the Blind SQL Injection was successfully developed, and the performance of the keywords generation for each algorithm was also successfully measured, i.e., 1.7852 seconds for Binary Search, 1.789 seconds for interpolation and 1.902 seconds for Linear Search.

Keywords: Blind SQL Injection, Linear Search, Binary Search, Interpolation Search

I. INTRODUCTION

Technology like the internet is very important. Especially internet technology such as websites. The growth of the website every year is increasing. According to a survey conducted by Netcraft in July 2018, website growth in the world has reached 1.6 billion hosts (Netcraft, 2018). This growth is in line with the number of cases of attacks on the website.

Criminal cases in cyberspace every year always grow bigger and more types. Based on OWASP Top 10 data, the widest attacks in the last two surveys that have been carried out by the organization are injection-based attacks such as SQL, NoSQL, OS, and LDAP (OWASP, 10AD). Data sourced from WhiteHat Security declares that there are at least 83% of websites that have at least one vulnerable critical gap (Grossman,2011). A system that is not developed with security in it will have a fatal impact if attacked. There are so many system breakers who take advantage of the damaged system, such as asking for ransom for data taken or breaking into the system just for the sake of pleasure. The main cause of poor website security is that developers do not implement effective security and program systems with safer codes (Acar et al., 2017).

The automation of SQL injection rarely causes Pentester to take a long time to carry out attacks and not know about the best test algorithms. The used method to penetrate the website is Blind SQL Injection. The use of the Blind SQL Injection method is based on its complexity when inputting the syntax one by one to

Received on Aug 2018. Accepted on May 2020

find the existing vulnerable gaps. With the research carried out, the syntax injection process is generated automatically. This method provides questions that produce true or false answers so that the injection process produces definite answers. The purpose of this research is to design and implement automation of Blind SQL injection using linear, binary, and interpolation search algorithms and analyze in terms of time performance for each algorithm that implemented.

II. Literature Review

A. *Penetration Testing*

Penetration Testing is a method used to test the security of a system by giving an attack to the system. Penetration testing itself has three stages that must be performed, namely Information Gathering, Attack Generation, and Response Analysis (W. G. J. Halfond, Choudhary, & Orso, 2009). Information Gathering phase is the process of analyzing targets that aim to identify information that is useful to determine the right attack of a vulnerable gap. Attack Generation is a process of attack carried out by using specific packages in accordance with the results of the information obtained in the previous phase. The next phase, Response Analysis, is useful for checking the attacks carried out, whether it is successful, and then making information records about the attacks that have been successfully carried out. In this study, the Information Gathering phase has been carried out because it uses the localhost website that has been created by the author, and the position to inject the attack is known. The focus of this research is on the Attack Generation phase.

B. *Blind SQL Injection*

Blind SQL injection attack scheme is an attack that uses a type of query that produces true or false answers. If the question that is injected into the server produces the correct answer, then the website will function normally. If it is wrong, then the website page has a very significant different characteristic (W. G. Halfond, Viegas, & Orso, 2006). Research on Blind SQL injection was carried out by Ali et al. with an application called MySQLInjector that scans website servers to detect hidden SQL vulnerable gaps that are efficient in penetrating their targets (Ali, Shakhatreh, Abdullah, & Alostad, 2011). MySQLInjector has three Blind SQL features, including Blind SQL Injection based on True/False response, True/Error response and Order by. Then Appelt et al. research mutation testing on SQL injection input values so that it can generate random input to detect the existing vulnerable SQL on the website (Appelt, Nguyen, Briand, & Alshahwan, 2014). Patil et al. do research to improve efficiency when scanning vulnerabilities while maintaining low false positives and false negatives (Patil, Marathe, & Padiya, 2016). To analyze the target, Patil's research uses a black box testing approach.

C. *Searching Algorithm*

The searching algorithm is an algorithm used to solve problems by retrieving specific stored information in a data set. Research on searching algorithms was used to measure performance from the Small Adaptive Intersection algorithm that outperformed other intersection algorithms in the context of search engines with optimum results obtained using binary search (Barbay, López-Ortiz, & Lu, 2006). Then, in other studies that are applying a searching algorithm on B-tree indexes by using interpolation search (Graefe, 2006).

SQL injection automation is done using three search algorithms. Each algorithm used in this research is explained in table 1 below:

Table 1
Types of Algorithms

No.	Types of Algorithms	Information
1.	Linear Search (Brute Force)	Results are searched in sequence based on the sequence of data created.
2.	Binary Search	The data will be divided into two segments then compared whetherthe results are at the top or bottom of the segment that has been done before and then will continue to divide it until the data sought is the same as the results.
3	Interpolation Search	Search for data by estimating the location of the data using key values.

III. RESEARCH METHOD

A. Data

The data used in this study is a localhost website that is connected to a MySQL database. The data for searching uses hex numbers that can be printed that range from 0x20 to 0x7f.

B. Test Matrix

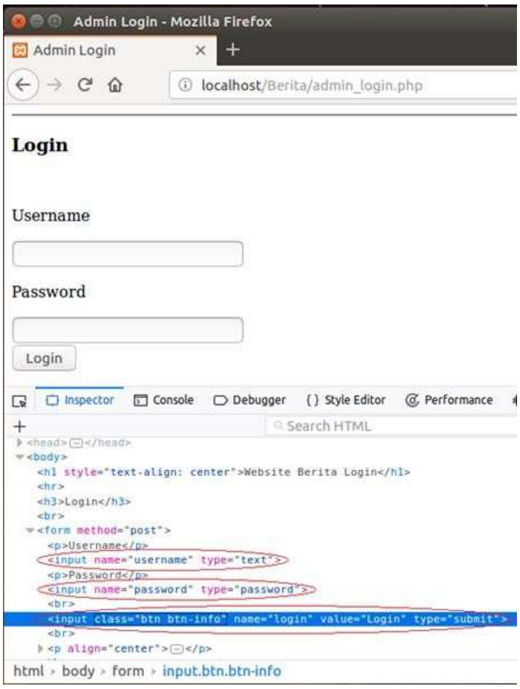
To measure the performance of the system being built, we compare the results of the data obtained from each algorithm in the form of time performance when searching for each correct letter from each algorithm used. In the formula to find the time (1) below, T_0 is the initial time when the letter search process is started while T_1 is the end time when the letter is obtained. Time performance is derived from the following formula:

$$\Delta T = T_1 - T_0 \tag{1}$$

C. Design of Algorithms

1. Blind SQL Injection Attack Scheme

In this research, we used the Illegal / Logically Incorrect Queries attack type. The attack is to utilize syntax errors or logical errors so that the website application returns an error page. The injection is done using a vulnerable gap from the website page taken from the login form. On this form, if you use the inspect element function in the browser, you can see form elements that can be used to automate. There are three elements in the <form method = "post"> that must be used, namely "username", "password" and class button.


Admin Login - Mozilla Firefox
Admin Login
localhost/Berita/admin_login.php
Login
Username
Password
Login
Inspector
Console
Debugger
Style Editor
Performance
Search HTML
html › body › form › input.btn.btn-info

```
<head>…</head>
<body>
  <h1 style="text-align: center">Website Berita Login</h1>
  <hr>
  <h3>Login</h3>
  <br>
  <form method="post">
    <p>Username</p>
    <input name="username" type="text">
    <p>Password</p>
    <input name="password" type="password">
    <br>
    <input class="btn btn-info" name="login" value="Login" type="submit">
    <br>
  <p align="center">…</p>
```

Fig. 1. Login Form

After obtaining these elements, the automation application will call it using the library requests contained in the Python library. The function is post type because the query that is designed is sent through the function then gives penetration to the website through the login page. The query generated by the application aims to finally find the username and password in the database.

The injection is done using the substring () function, which consists of three parameters, namely column_name, first_index, and a number of characters. The injection query is in the main query of the gap in the login form so that the main query cannot work and replace it with the injection query. Queries that made will be automated by the algorithms in table 1 above. The first thing to look for is the name of the database. Then from that name, the application will continue to search for a list of existing tables. After the table name is obtained, the user must enter the name of the table displayed by the indicated application containing a list of usernames and passwords. After that, the application provides an output in the form of a user list along with the password registered on the website.

2. Linear Search Algorithm

Attack automation schemes using linear search algorithms are explained through the flowchart as follows:

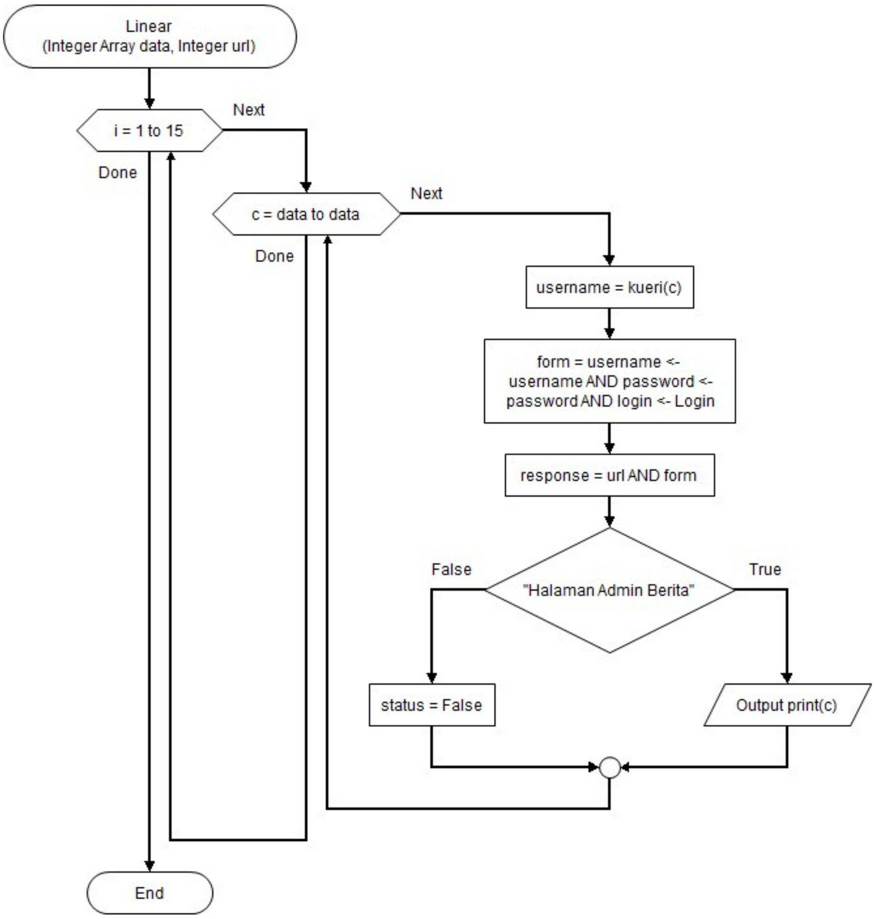


Fig. 2. Linear Search Flowchart

The algorithm looks for each correct letter in the database and then does it repeatedly for fifteen times. Repetition fifteen times is a prediction of the length of the char arranged into a sentence so that every time the program repeats and the answer is correct. Then it produces a letter indicating that the letter is the data that is sought. The search uses hex data sequentially from 0x20 to 0x7f. Form and Response function is to connect applications with the system on the web so that applications can provide query results that are made to be sent to the login form contained on the website then penetrate the database.

3. Binary Search Algorithm

The attack automation scheme using binary search algorithms is explained through the flowchart as follows:

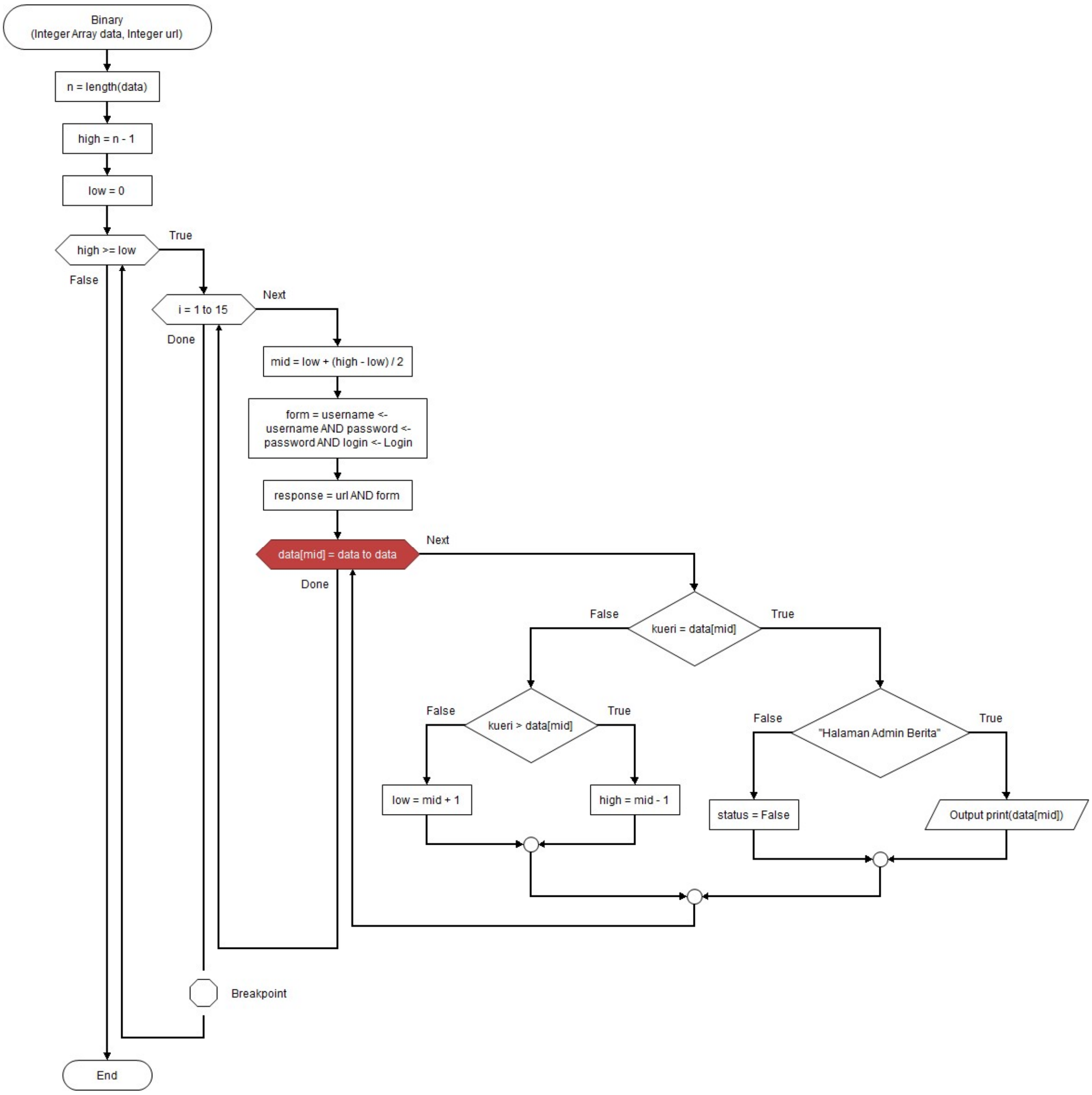


Fig. 3. Binary Search Flowchart

The algorithm looks for each letter in the data by dividing the data into two. Then the data is compared whether the results are above or below until it gets all the results of each letter arranged into a word. The Form and Response have the same function in the linear search algorithm.

4. Interpolation Search Algorithm

The attack automation scheme using the interpolation search algorithm is explained through flowchart as follows:

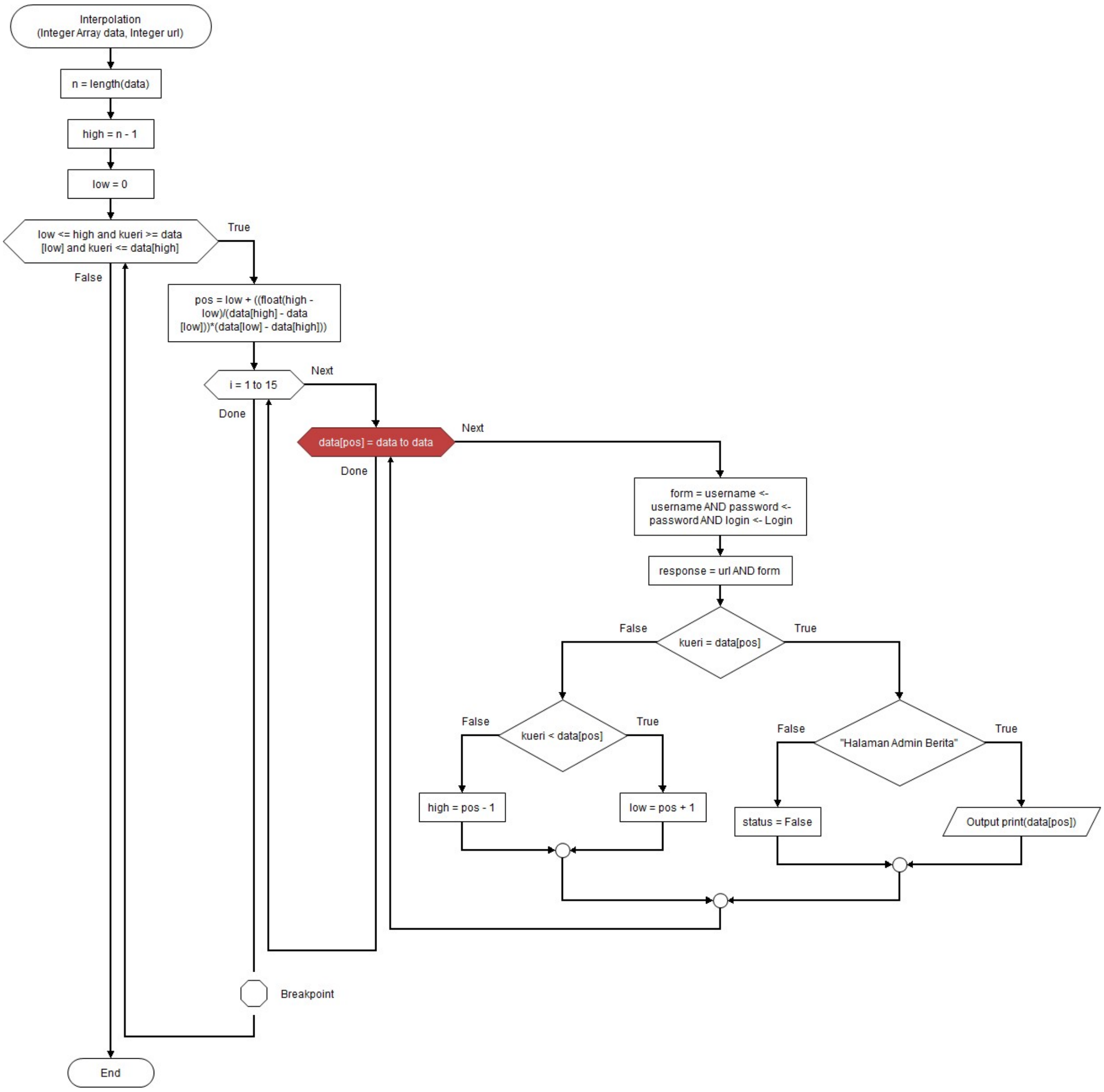


Fig. 4. Interpolation Search Flowchart

The algorithm looks for results by estimating the location of the data so that the process of starting a search is not like a binary search that is always from the middle but starts from data that is close to the result. The heading is used to find out the location of the data that is searched, whether it has given the right or wrong results.

IV. RESULTS AND DISCUSSION

A. *Test Result*

Tests were carried out as many as ten trials on each algorithm. Testing is done based on the search time for each letter. For the search to be balanced, each algorithm looks for the same target, namely the name of the database from the website. For search data, each letter can be seen in the attachment section. The test was carried out with two formations, the first test formation was Linear-Binary-Interpolation, and the second was Linear-Interpolation-Binary. The purpose of doing these two formations are to be able to explore the strength of each algorithm. The following is the average time of each algorithm looking for each letter arranged into a sentence called "webberita" which is the name for each experiment that is carried out:

Table 2.
TEST RESULTS OF LINEAR-BINARY-INTERPOLATION

Experimental Data	Linear Search (s)	Binary Search (s)	Interpolation Search (s)
1	2.188714889	1.421164444	1.587965444
2	2.033020444	1.574856556	1.799804111
3	1.540954556	1.571406	1.608142333
4	1.407849667	1.773449667	1.451089778
5	1.821896222	1.525930444	1.552824444
6	1.645822667	1.436014889	1.626720111
7	1.775803556	1.708488222	1.511631667
8	1.457512222	1.530847222	1.551522778
9	1.444242222	1.394215778	1.487922778
10	1.424257	1.511222778	1.612802444
Average	**1.674007344**	**1.5447596**	**1.579042589**

Based on Table 2 above, testing with the Linear-Binary-Interpolation formation shows the best time performance results, namely Binary Search with an average time of 1.54 seconds, then Interpolation Search 1.57 seconds and the last position of Linear Search 1.67 seconds.

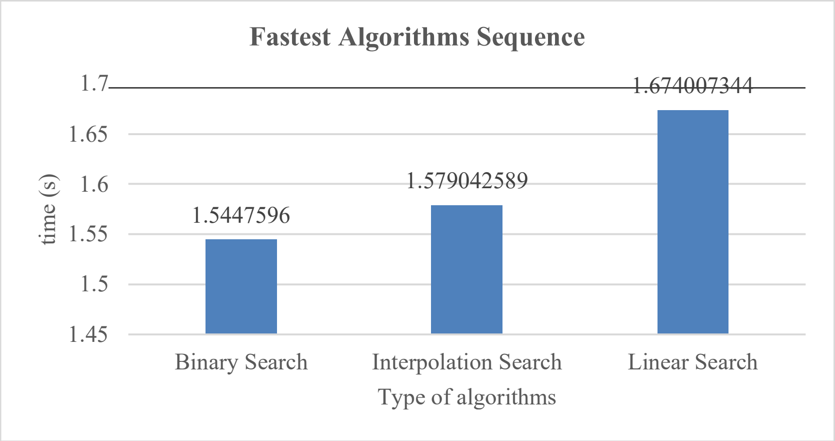


Fig. 5. Test results of Linear-Binary-Interpolation formation

Table 3.
Test Results of Linear-Interpolation-Binary

Experimental Data	**Linear Search (s)**	**Interpolation Search (s)**	**Binary Search (s)**
1	4.186308778	2.192088222	1.998454889
2	2.101781111	2.003216111	2.113540556
3	1.925351	2.041105444	1.937580889
4	1.638294333	1.830556222	2.105688222
5	1.888069111	2.062408778	1.864982444
6	2.255281889	1.776017222	2.142145111
7	1.813413333	2.207248222	1.933683111
8	1.810432556	1.790293889	2.126194889
9	1.885558	2.022441444	2.013260667
10	1.804168111	2.061836556	2.020775
Average	**2.130865822**	**1.998721211**	**2.025630578**

Based on table 3 above, testing with the Linear-Interpolation-Binary formation shows the best time performance results, namely Interpolation Search with an average time of 1.99 seconds, then Binary Search 2.02 seconds and the last position Linear Search 2.13 seconds.

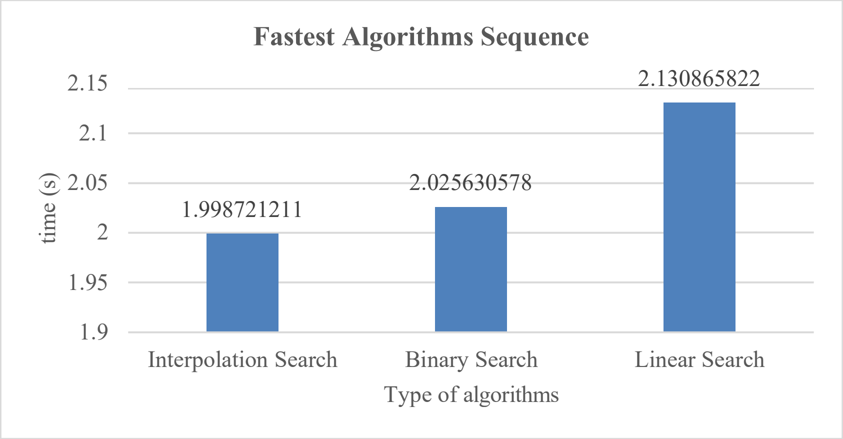


Fig. 6. Test results of Linear-Interpolation-Binary formation

After doing the two formations above, based on Figures 5 and 6, the average time performance of the two formations is the fastest Binary Search with 1,785 seconds, then Interpolation Search with 1,789 seconds and Linear Search with 1,902 seconds.

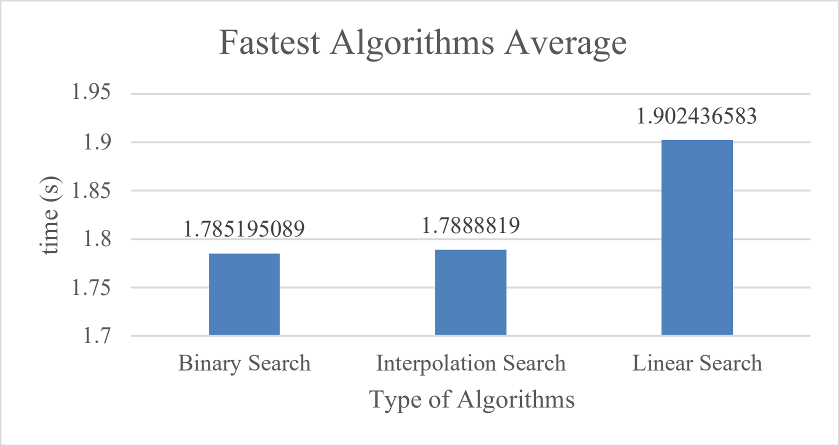


Fig. 7. Average test results

B. Test Results Analysis

Based on the data that has been obtained, the algorithm that has the best time performance is binary search followed by interpolation search and the last linear search. Based on research conducted by Rahim et al, an interpolation search has better performance than the other two algorithms (Rahim, Nurarif, Ramadhan, Aisyah, & Purba, 2017). But in this study get different results because it is affected by the processes that exist in the operating system so that the response time on localhost can be different. Furthermore, the data used is data in the order corresponding to hex numbers so that the formula in the interpolation algorithm cannot give the best results.

In the experiments were carried out ten times, there were anomalies such as sometimes linear algorithms faster than binary or interpolation and vice versa. This is because the process of the localhost has different response times that affect the results when retrieving data. To reduce this effect, the data collection process is carried out when the new operating system lights up without any other process other than running an experimental algorithm then for subsequent experiments carried out with that method.

V. Conclusion

Based on the testing and analysis that has been done, it can be concluded that:

1. The Binary Search algorithm is the fastest algorithm based on test results with a time of 1,7852 seconds, then Interpolation Search with 1,789 seconds and Linear Search 1,902 seconds.
2. Based on the data sequence of hexadecimal numbers used, the Binary Search algorithm is the most optimal algorithm.
3. The process of automating the injection carried out can make it easier for a Pentester to carry out an attack on the website.

REFERENCES

Acar, Y., Stransky, C., Wermke, D., Weir, C., Mazurek, M. L., & Fahl, S. (2017). Developers Need Support, Too: A Survey of Security Advice for Software Developers. *Proceedings - 2017 IEEE Cybersecurity Development Conference, SecDev 2017*, 22–26. https://doi.org/10.1109/SecDev.2017.17

Ali, A. B. M., Shakhatreh, A. Y. I., Abdullah, M. S., & Alostad, J. (2011). SQL-injection vulnerability scanning tool for automatic creation of SQL-injection attacks. In *Procedia Computer Science* (Vol. 3, pp. 453–458). https://doi.org/10.1016/j.procs.2010.12.076

Appelt, D., Nguyen, C. D., Briand, L. C., & Alshahwan, N. (2014). Automated testing for SQL injection vulnerabilities: an input mutation approach. *Proceedings of the International Symposium on Software Testing and Analysis*. https://doi.org/10.1145/2610384.2610403

Barbay, J., López-Ortiz, A., & Lu, T. (2006). Faster adaptive set intersections for text searching. In *International Workshop on Experimental and Efficient Algorithms* (pp. 146–157). Springer.

Graefe, G. (2006). B-tree indexes, interpolation search, and skew. In *Proceedings of the 2nd international workshop on Data management on new hardware* (p. 5). ACM.

Grossman, J. (2011). 10 important facts about website security and how they impact your enterprise. *WhiteHat Security*, 3.

Halfond, W. G. J., Choudhary, S. R., & Orso, A. (2009). Penetration testing with improved input vector identification. In *Software Testing Verification and Validation, 2009. ICST'09. International Conference on* (pp. 346–355).

Halfond, W. G., Viegas, J., & Orso, A. (2006). A classification of SQL-injection attacks and countermeasures. In *Proceedings of the IEEE International Symposium on Secure Software Engineering* (Vol. 1, pp. 13–15). IEEE.

Netcraft. (2018). Web Server Survey. Retrieved from https://news.netcraft.com/archives/2018/07/19/july-2018-web-server-survey.html#more-26592

OWASP, T. (10AD). Application Security Risks-2017, Open Web Application Security Project (OWASP).

Patil, S., Marathe, N., & Padiya, P. (2016). Design of efficient web vulnerability scanner. In *Inventive Computation Technologies (ICICT), International Conference on* (Vol. 2, pp. 1–6).

Rahim, R., Nurarif, S., Ramadhan, M., Aisyah, S., & Purba, W. (2017). Comparison Searching Process of Linear, Binary and Interpolation Algorithm. In *Journal of Physics: Conference Series* (Vol. 930, p. 12007). IOP Publishing.