Multi-Knowledge Electronic Comprehensive Journal For Education And Science Publications (MECSJ)
ISSUE (28), January (2020)
ISSN: 2616-9185

www.mecsj.com

Using the Behavioral Biometrics to Prevent SQL Injection Attack

Makera M. Aziz*
Computer Science Department, College of Computer Science and Mathematics,
University of Mosul, Mosul, Iraq
Email: makera.aziz@uomosul.edu.iq

Dina Rafaa Ahmed
Software Department, College of Computer Science and Mathematics,
University of Mosul, Mosul, Iraq
Email: dinasalimagha@uomosul.edu.iq

Abstract

Using the web has grown a lot in the last decades transferring the information between the nodes of the network, keep this information secret, and away from the attackers is very important. Classify the type of attacks and find ways to avoid them is a goal for many researchers. This paper deals with one type of attack and the most common attack that threatens the database is the SQL injection attack by using behavioral biometrics and specifically typing speed. It is suggested the system that measures the time and the duration that need for each user to complete his/her login duration and save this time to use when the user doing his login again. It compares the login duration time for the current user login with the duration for his last login that has been saved before. If this time is different that means this login is an attack and the system must prevent the attacker to access the database. The result showed that the proposed method can be used to detect the attacks that target the database.

Keywords: SQL injection Attack, Typing Speed, Behavioral Biometrics.

الملخص

إن إستخدام صفحات الويب نما بشكل كبير في العقود الاخيرة ولذلك فإن عملية نقل المعلومات بين أجزاء الشبكات وعملية الحفاظ على سرية هذه المعلومات وحفظها بعيداً عن المهاجمين مهمة جداً. عملية تصنيف الهجومات وإيجاد طرق لتجنب كل نوع أصبح هدف من أهداف الباحثين. يتعامل هذا البحث مع نوع من أنواع الهجومات الذي تتعرض له الشبكات والذي يعد الأكثر شيوعاً في تهديد قاعدة البيانات وهو SQL injection اي حقن ال SQL. حيث اقترح البحث إستخدام أحد المقاييس السلوكية وهو سرعة الطباعة لحماية الأنظمة من هجمات حقن ال SQL. فقد إقترح البحث قياس الوقت والفترة التي يستغرقها كل مستخدم للنظام لإكمال عملية تسجيل الدخول الى النظام (login) وحفظ الوقت المستغرق لإتمام هذه العملية في قاعدة بيانات وإستخدام هذا الوقت ومقارنته مع الوقت الذي يستغرقه المستخدم في عملية الدخول مرة اخرى. فإذا كان الفارق بين الوقتين كبير فإن هذا يشير الى أن عملية تسجيل الدخول الحالية هي عملية هجوم ويجب على النظام منع المستخدم الحالي من الحصول على تخويل الدخول الى النظام. بينت النتائج إن الطريقة المقترحة يمكن إستخدامها لتميز الهجوم الذي يستهدف قاعدة البيانات.

الكلمات المفتاحية: هجوم حقن الSQL، سرعة الطباعة، المقاييس السلوكية.

1. Introduction

The web application has a huge usage nowadays. The database of these webs includes sensitive and important information like (financial, military, etc.) This type of information makes these webs as a target for attacks. Provide a secure environment for these webs is one of the important aspects to protect this information from different types of attacks (Alwan & Younis, 2017). A lot of attacks are there that threaten the networks. The most common one is the SQL injection in this type of attack the attacker uses the SQL query statement and reform it in a way that makes him execute it to get unauthorized access to the database and get full access to the database (Jawanjal, et al., 2018). In SQL injection, the attacker tries to change the structure of the SQL query by injecting (or inserting) malicious code to the query statement (Sajjadi & Pour, 2013). This code changes the conditional of the query statement and for example put some conditions that always true. The hacker uses the user's input tools like a textbox to inject the SQL statement. Most of the SQL injection is used when the user inserts the user name, password, or both (Manmadhan & T, 2012).

Using new biometrics in addition to the password is a method that has been used to prevent SQL injection. Biometrics can be physiological like a fingerprint, face recognition; on the other hand, there are behavioral biometrics like keystroke dynamics, mouse movement dynamic, and voice recognition (Koong, Yang, & Tseng, 2014). Figure (1) shows the biometrics types, both types of biometrics can be used to recognize the user authorization and this biometrics cannot be repeated between two users. In this study, behavioral biometrics are used by measuring the time that the user needs to complete his login process. The login process asking the user to input his password and user name in the text box to get access to the database. The time that needs for this process depends on many factors. The most important one is the typing speed and the keystroke rhythm that is considered as behavioral biometrics and different users need different durations to complete the login process. Many types of SQL injection are there that using different ways to attack the database; on the other hand, many researchers suggested different methods to deal with this problem and to protect the database from unauthorized access (Gupta & Singhal, 2015). This study suggested using the typing speed to avoid the SQL injection attack.

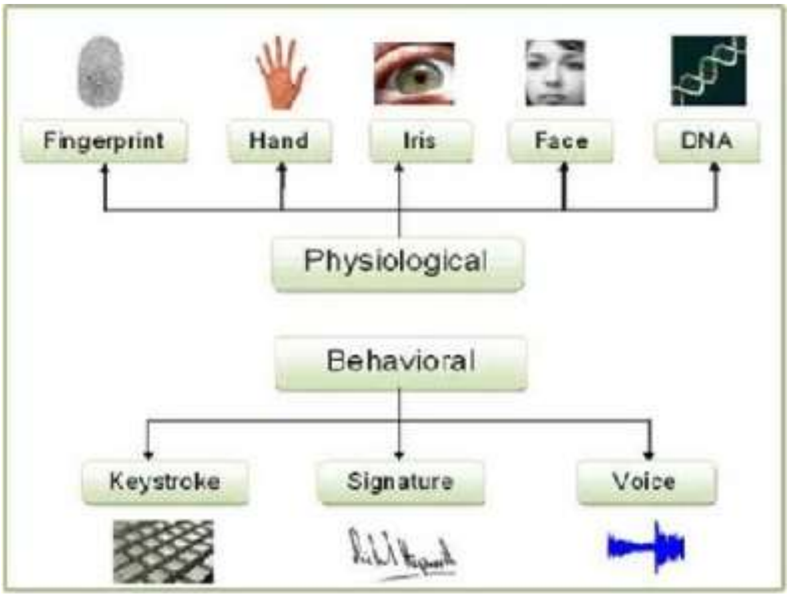

Figure (1): Biometrics Types (Mohammed, Jasim, & Mohammed, 2016)

2. Related Works

(Aziz & Ahmed, 2016) Suggested a method that keeps the input value used by the attacker away from the query statement and converts it to a static string that cannot be used to restructure the query statement. This static string will be compared with database values like username and password. This method prevents the special character to be used to restructure the query and it will be read as a single statement. (Soewito, E.Gunawan, & Hirzi, 2018) In this work, the researchers suggested saving the input query as token in a dynamic table on both front end and back end side and compare between them during runtime, if the two tables are matching each other the system will give the database access to the user; otherwise, the request will be rejected. (Namdev, Hasan, & Shrivastav, 2012) To provide the secure login to the system this work used the hashing to meet this goal. The hash value is computed for both inputs (user name and password) and saves this value in the database table.

The user has to put the correct user name and password that give the same value of hashing to access the database. (Jawanjal, et al., 2018) Apriori algorithm and AES algorithm both used in this work. A mechanism between client and server application is set and all the requests that go from client to server should pass through this mechanism to check before processing it by the server. If the request includes anything that refers to attack, the system will not give access to the database. (Balasundarama & Ramarajb, 2012) This method use multi-model approach: (1) text parser has been used to check the input statement, detect the input, and prevent attacker to use the special characters that used to restructure SQL query to target the database, (2) put some constructions to the user ID and Password to prevent attacker to use the character that let him\her inject the SQL query statement, (3) compute the ASCII for user name and password to detect the change in any character in the user name or password. (Lee, Jeong, Yeoc, & Moon, 2012) This method is comparing between a static method and a dynamic method. In the dynamic method, the attribute value of the SQL query has to be cleaned during real-time before comparing it with a static method.

3. Proposed Method

Different users have different typing speeds. The login process is needed to type a full user name and full password to be complete. Most of the applications asked their user to input his/her user name with a minimum of (6-8) characters and asked to input his/her password with small letters mix with numbers, capital letters, and special characters. These increase the duration of the login process (login process: input the first character until click on the login button). On the other hand, the SQL injection is used a fewer number of characters and this needs less time to type it comparing with the time that needs for regular login. Sometimes, the attacker needs to fill one textbox only (like only username textbox, or only password’s textbox) to get his/her access to the database and this needs less time than the time that needs by an authorized user to do his regular login.

This study will compare the time that the user needs for his/her login with the time that he/she needs for his/her next login. If there is a big difference between them that means there is an unauthorized user try to access the database. The first login duration for the user will be stored from the registration process in the system and the login time will take the average of the last three logins of the user. Figure (2) shows how the proposed method is working: (when the user registered in the system, the system asked to do the login process three times to take the average of login duration and store in the database table to be compared with the next login duration.

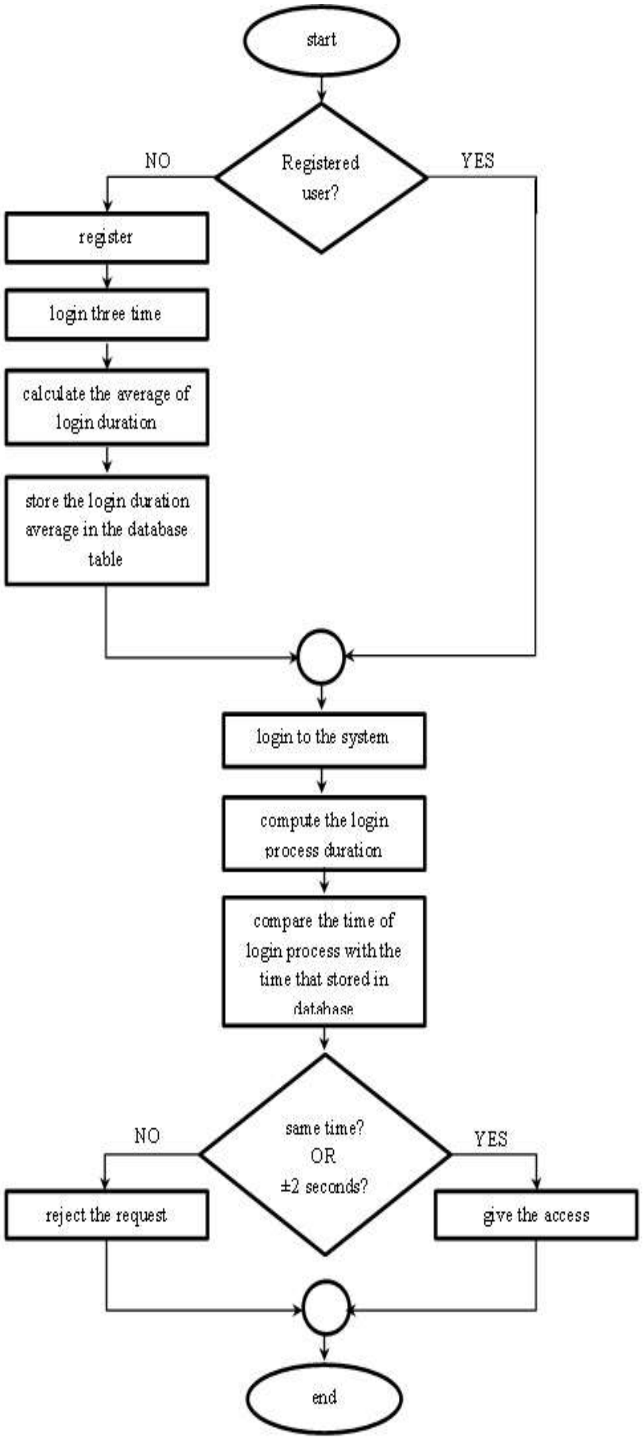
start
Registered user?
NO
YES
register
login three time
calculate the average of login duration
store the login duration average in the database table
login to the system
compute the login process duration
compare the time of login process with the time that stored in database
same time? OR ±2 seconds?
NO
YES
reject the request
give the access
end

Figure (2): The Proposed Method

3.1.Data Collection

The data that has been used in this work collected from ten users. The users asked to type their user name and password to compute the time needed for each user to complete the login process and save the total time in the database table to be used for authorization detection with user name and password. Figure (3) illustrates the duration time for regular login.

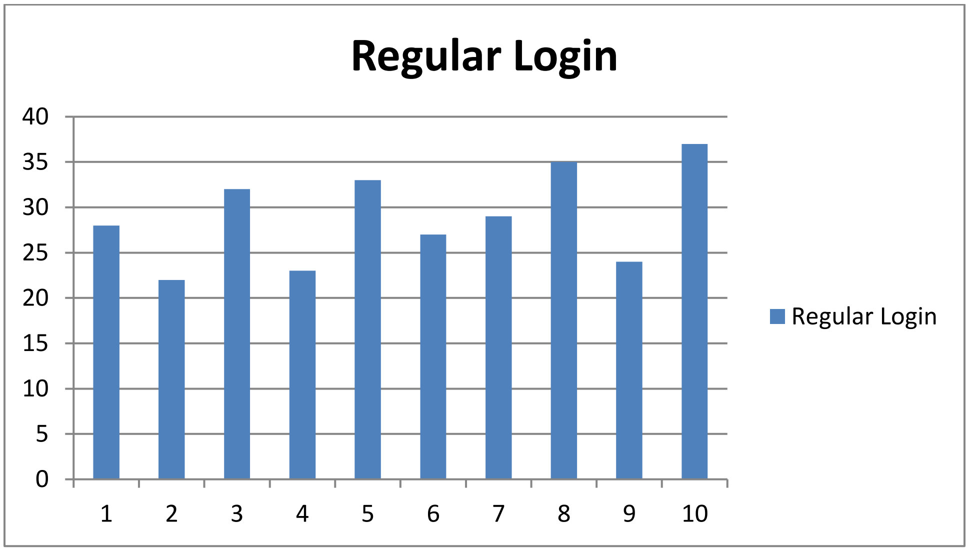

Figure (3): The Duration Time For Regular Login (for 10 users)

4. Results

The system is dealing with ten users and compute the time that needed for regular login and compare with the SQL injection statement that input in password textbox only uses the segment “**'or '1'='1**”. The result shows that the time that uses in SQL injection is less than the time that needs to do regular login. As shown in figure (4).

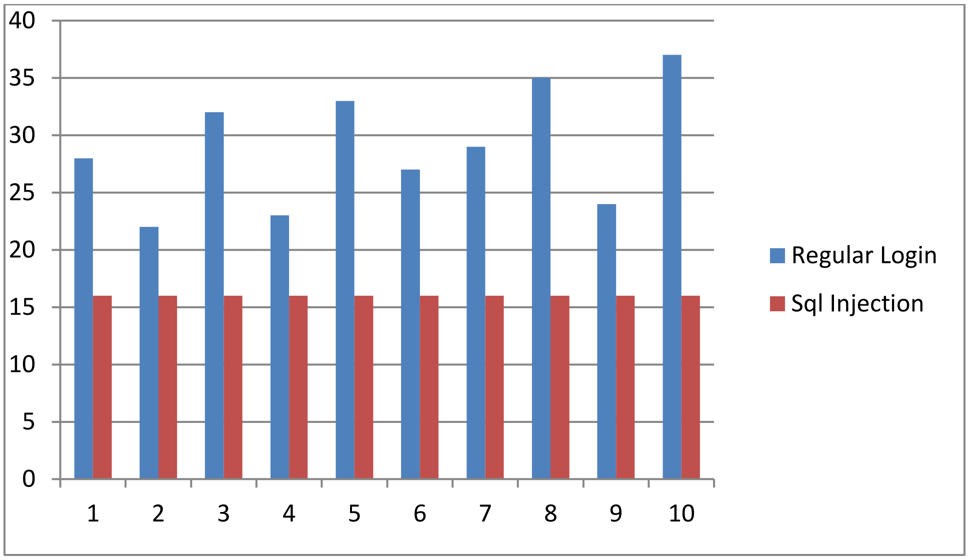

Figure (4): The SQL Duration and Login Duration

5. Conclusion

The proposed method showed that there is a noticeable difference between the time that the user needs to do the regular login and the time that need for typing the SQL injection statement. Using keystroke as behavioral biometric can recognize the SQL injection attack and can be used to protect the system by computing the time that needs (for both process regular login and SQL injection) and compare between them. If the difference is more than 2 seconds (plus/minus) that means this is the unauthorized user and this is an attack. The proposed method successfully worked to recognize the attack.

References

- Alwan, Z. S., & Younis, M. F. (2017). Detection and Prevention of SQL Injection Attack: A Survey. *International Journal of Computer Science and Mobile Computing, 6*(8), 5-17.
- Aziz, M. M., & Ahmed, D. R. (2016). Proposed Method to Prevent SQL injection Attack. *Iraqi Journal for Computers and Informatics, 42*(1), 59-63.
- Balasundarama, I., & Ramarajb, E. (2012). An Efficient Technique for Detection and Prevention of SQL Injection Attack using ASCII Based String Matching. *Procedia Engineering*, 183-190.
- Gupta, J., & Singhal, R. (2015). SQL Injection A threads to web application. *IJRCS, 2*(1), 34-37.
- Jawanjal, S., Shegokar, S., Nandurkar, V., Ardak, R., Chaudhari, S., Rithe, S., et al. (2018). An Efficient Technique for Detection and Prevention of SQL Injection Attack in cloud. *International Journal for Research in Applied Science & Engineering Technology, 8*(4), 2669 - 2673.
- Koong, C.-S., Yang, T.-I., & Tseng, C.-C. (2014). A User Authentication Scheme Using Physiological and Behavioral Biometrics for Multitouch Devices. *The Scientific World Journal*, 1-12.
- Lee, I., Jeong, S., Yeoc, S., & Moon, J. (2012). A novel method for SQL injection attack detection based on removing SQL query attribute values. *Mathematical and Computer Modelling*, 58-68.
- Manmadhan, S., & T, M. (2012). A METHOD OF DETECTING SQL INJECTION ATTACK TO SECURE WEB APPLICATIONS. *International Journal of Distributed and Parallel Systems, 3*(6), 1-8.
- Mohammed, R. S., Jasim, H. N., & Mohammed, A. Z. (2016). Iris Matching Using SURF Algorithm. *International Journal of Signal Processing, Image Processing and Pattern Recognition, 9*(12), 91-102.

- Namdev, M., Hasan, F., & Shrivastav, G. (2012). A Novel Approach for SQL Injection Prevention Using Hashing & Encryption (SQL-ENCP). *International Journal of Computer Science and Information Technologies, 5*(3), 4981 - 4987.
- Sajjadi, S. M., & Pour, B. T. (2013). Study of SQL Injection Attacks and Countermeasures. *International Journal of Computer and Communication Engineering, 2*(5), 539 - 542.
- Soewito, B., E.Gunawan, F., & Hirzi. (2018). Prevention Structured Query Language Injection Using Regular Expression and Escape String. *Procedia Computer Science, 135*, 678-687.