Securing Relational Databases with an Artificial Immunity Features

Ayman Mohamed Mostafa
Faculty of Computers and Informatics
Zagazig University- Egypt

Mohamed H. Abdel-Aziz
Vice-Dean of Education and Student Affairs
Ain Shams University- Egypt

Ibrahim M. El-Henawy
Professor of Computer Science
Zagazig University- Egypt

ABSTRACT

Database security is considered one of the major computer science research trends because of its importance in maintaining the privacy, integrity, and confidentiality of data. Human immune system is a set of defense mechanisms that can be used to defend the body against diseases caused by pathogens. Artificial immune system is the artificial simulation of human immunity that can be applied to computer security applications. The main goal of this paper is to develop a database security system based on danger theory. Danger theory is one of the most recent algorithms of artificial immunity that can provide interactive features for securing relational databases. By merging the developed features of artificial immunity to the security system, the secrecy of the database can be maintained.

Keywords

Database security, artificial immune system, danger theory

1. INTRODUCTION

Artificial immune system (AIS) is an umbrella term that covers all efforts to develop computational models inspired by biological immune systems [3]. It is an area of research that bridges the disciplines of immunology, computer science and engineering [1]. Artificial immune system can be regarded also as a set of algorithms generated from natural immune system [2, 6]. One of the major algorithms of artificial immune system is negative selection algorithm (NSA). Negative selection algorithm is a technique which can be used for anomaly detection, computer virus detection, and network security. Discrimination between self and non-self is considered one of the major mechanisms in the complex immune system. Artificial negative selection is a computational imitation of self/non-self discrimination.

The original negative selection algorithm (NSA) relies on the fact that data cannot corrupted when the detectors are generated [4]. This reflects the idea that self samples are at least considered correct regardless of whether they are complete or not. Even if, the self samples are complete as well as correct, negative selection algorithm (NSA) still probabilistic in most methods because it depends on data properties [3]. Negative selection algorithm is based on a set of detectors which can defend the system from any intruders. In order to have perfect detector coverage and achieve high detection rate, it is necessary to have too large number of detectors that are capable of detecting non-self samples. This will increase the time and space complexity.

The main goal of this paper is to open a new trend for applying the field of artificial immune system for securing relational databases. Based on our papers presented in [5, 7, and 8], this paper presents a database security system which merges the merits, features, and capabilities of the artificial immune system in order to provide an interactive security system which can be used in different real world applications.

2. RELATED WORK

2.1 Human Immune System

The main task of human immune system (HIS) is to defend the body against diseases caused be pathogens. Pathogens are foreign substances like viruses, fungi, parasites, and bacteria which attack the body. In order to detect and destroy pathogens, the immune system contains certain types of white blood cells called lymphocytes which can recognize pathogenic patterns [9]. Lymphocytes can generate B-cells and T-cells. B-cells are blood cells that have the ability to create antibodies to detect pathogens. T-cells are tissue cells that can regulate the generated antibodies. The infected cell by a virus is not directly detectable by antibodies because the cell carries no binding information on their surface. To solve this problem, all cells contain Major Histocompatibility Complex (MHC) molecules which are able to present intruded viral peptides on the cells surface.

2.2 Artificial Immune System

2.2.1 Negative Selection Algorithm (NSA)

In the generation stage of negative selection algorithm, the detectors are generated by some random process and censored by trying to match self samples. The matched candidates are eliminated, and the rest are kept as mature detectors [3]. In the detection stage, a collection of detectors is used to check whether an incoming data instance is self or non-self. If it matches any detector, it is claimed as non-self or an anomaly.

An up to date survey on negative selection algorithms was published in [3]. Though different variations of negative selection algorithms have been frequently proposed, the main characteristics of this method described in [4] still remain. Luo et al. [10] proposed a novel negative selection algorithm called r[]-NSA with binary representation. In r[]-NSA, each detector, called r[]-detector, has the corresponding array keeping multiple partial matching length. Two evolutionary Negative Selection Algorithms (ENSA) using binary representation were proposed in [11], which are simple ENSA and basic ENSA, respectively. In simple ENSA, if the detector matches the data, an abnormal change is identified. Otherwise, the initial detector set is evolved to the next generation through mutation, positive selection and negative selection. Such evolutionary generation loops continue until the abnormal change is detected. The steps in basic ENSA are similar to simple ENSA. However, an additional randomly generated detector set is added to the next generation detector set in basic ENSA. They claimed that the advantage of this change in basic ENSA is that it can search in the global space and prevent converging to local optima.

2.2.2 Artificial Immune Network (AINE)

An immune network theory suggests that immune system is capable of achieving immunological memory by the existence of a mutually reinforcing network of B-cells. The B-cells not only stimulate each other but also suppress connected cells to regulate the over stimulation of B cells in order to maintain a stable memory.

Artificial immune networks were defined and implemented by Timmis et al. [12]. As presented in [16], Omni-aiNet was developed to solve single and multi-objective optimization problems, either with single and multi-global solutions. Omni-aiNet united the concepts of Omni-optimization with distinctive procedures associated with immune-inspired concepts. Zhang et al [13] proposed a Tree Structured Artificial Immune Network (TSAIN) for data clustering and classification. In this model, a topological link is setup between two antibodies immediately after one has reproduced by another with no need to set a threshold for this connection. As presented in [15] most applications of artificial immune network are applied to clustering, classification, optimization, and robotics.

2.2.3 Clonal Selection Algorithm (CSA)

Clonal selection theory states that a clonal expansion of the original lymphocyte occurs when the original lymphocyte is activated by binding to the antigen; however, any clone of the activated lymphocyte with antigen receptors specific to molecules of the organism's own body (self-reactive receptors) is eliminated during the development of the lymphocyte [1]. Castro et al [14] proposed a clonal selection algorithm named CLONALG for learning and optimization, CLONALG generates a population of N antibodies, each specifying a random solution for the optimization process. As presented in [15] most applications of clonal selection algorithm are applied to system modeling, optimization, and data mining.

2.2.4 Danger Theory

Danger theory (DT) postulates that the human immune systems respond to the presence of molecules known as danger signals, which are released as results of unnatural cell deaths. Immune system will only respond when damage is indicated, and be actively suppressed otherwise. The danger theory (DT) proposes that antigen presenting cells (APCs) have danger signal receptors (DSR) which recognize signals sent out by distressed or damaged cells. These signals inform the immune systems to initiate immune responses [2]. According to the danger theory presented in Figure 1, a cell that dies unnaturally sends outs the danger/alarm signal.

The danger signal establishes a danger zone around itself. On the other hand, the antigens near the cell that emits the danger signal are captured by APCs such as macrophages, and then travel to the local lymph node and present the antigens to lymphocytes. The antibodies secreted by B cells match the antigens, but only those that match the antigens in the danger zone will be activated. Those that do not match or are not in the danger zone will not become stimulated [1].

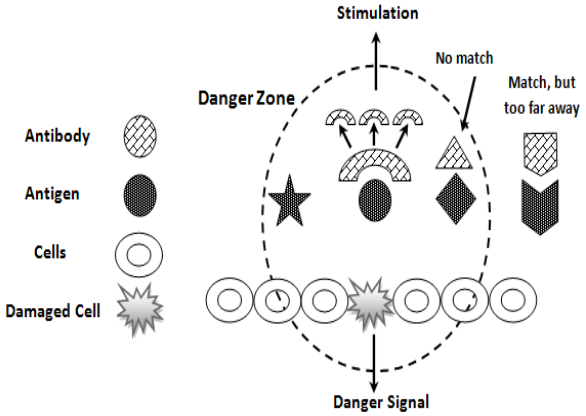

Figure 1. Danger Theory Mechanism

3. DATABASE SECURITY LEARNING MECHANISM

Based on our papers [5, 7, and 8] that present a package of interactive database security policies and architectures, the main goal of this paper is to provide an artificial immune security algorithms to prevent database administrators and users from performing any hostile act. The main target is to build a set of immune-based algorithms that work as countermeasures for securing relational databases. Figure 1 presents the security architecture hierarchy based on our paper [8].

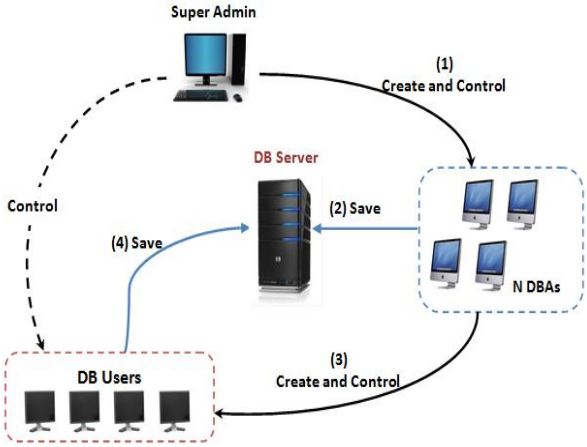

Figure 2. Database Security Architecture

By deploying the architecture presented in Figure 1, three levels of security are developed to provide a harmonious interaction between super administrator (**SA**), database administrators (**DBAs**), and authorized users. This architecture is illustrated as followed:

3.1 Super Administrator (SA)

The super administrator (**SA**) is located on the top hierarchy of the database security system. He manages all authorizations and capabilities to control and monitor database administrators, users, and transmissions through the database server. He begins the system by performing the following operations:

1. Creating his account with a secret password and an additional secret key certificate for more privacy.

2. Providing the maximum number of operations (insert-update-delete-select) allowed to each user to distinguish between active, intermediate and inactive user profiles. For example, the super administrator fills in the parameters for (Insert) operation as follows {Insert: Max (20), Active (15), Intermediate (8), Inactive (1)}. As a result, any active user can perform between 15 and 20 insert operations. Intermediate users can perform insert operations >= 8 and < 15. Inactive users can perform insert operations >=1 and <8. The DML operations limits will be a countermeasure to prevent any malicious user from transmitting large number of operations. The super administrator (SA) can modify the predefined limits at any time if there are any critical operations that must be transmitted.
3. Determining the total number of database administrators (**N DBAs**) who can connect to the database system and the number of shadows (**K DBAs**) that must be found to provide a secret sharing process for every request from a single administrator. The maximum number of shadows (**K DBAs**) must be less than or equal to the maximum number of administrators connected to the system (**N DBAs**).
4. Providing the *username*, *password*, and a *secret key certificate* for each database administrator. The secret key certificate will be generated as follows:

   - If the database scheme contains N DBAs such that any K DBAs shadows can combine together to control the transmission process, the DBMS generates the following polynomial equation:

   $$F(x) = (ax^{k-1} + bx^{k-2} + cx^{k-3} + \ldots + nx^{0}) \bmod p \quad (1)$$

   Where ($a, b, c, \ldots, n$) are random coefficients and p is a prime number. These parameters are inserted into the system by the super admin (**SA**). Any K shadows can be used to create K equations by evaluating the polynomial at n different points such that:

   $$k_i = F(x_i) \quad (2)$$

3.1.1 Building Danger Value Signal

According to the danger theory presented in [1, 2], one of our major contributions in this paper is to adapt the danger theory to the field of database security by building the following parameters:

- Main Danger value Threshold (**MDVT**)

The main danger value threshold (**MDVT**) is the backbone of the system that provides the amount of sensitivity to the security system. If the super administrator (**SA**) wants to increase the sensitivity of the system to any hostile act, he can decrease the main danger value threshold (**MDVT**). This operation is similar to the "inflammation system" in the human immune system. In the human immune system, the more the inflammation of the human body, the less the amount of treatment dose can be given. The main danger value threshold (**MDVT**) is given a percentage value to specify the degree of inflammation that the system will alarm starting from this value.

- System Privileges Danger Signals

The super administrator (**SA**) specifies a danger percentage value for DML system privileges. Each DML system privilege can take different danger value according to its dangerous to the security system. For example, {**Select: 30%** danger, **Insert: 70%** danger, **Update: 80%** danger, **Delete: 90%** danger**}**. This means that insert operation has a dangerous effect of 70% to the security system and so on.

- Database Privileges Danger Signals

The super administrator (**SA**) can specify a danger value signal for each database privilege that may be granted to users. This danger value signal (**DVS**) explains the amount of danger that may harm the system if the user succeeds in breaching the security system by passing these operations. This is presented in Table 1. Table 1 presents 14 privileges of database that may be granted to the authorized users. The presented danger value signals (**DVSs**) can store different signal values according to the super administrator requirements in increasing or decreasing the sensitivity of the security system.

Table 1. Database Privileges

Database Privileges DVS			
Privilege	**DVS**	**Privilege**	**DVS**
Create Table	80%	Create Index	40%
Create View	70%	Create Synonym	10%
Create Function	50%	Create Procedure	50%
Create Sequence	10%	Alter Table	80%
Alter Index	40%	Alter View	70%
Alter Synonym	10%	Alter Function	50%
Alter Procedure	50%	Alter Sequence	10%

- R-Contiguous Bit Matching (**RCB**)

The (**RCB**) matching algorithm is one of the algorithms that will be used later in our security system detection phase to detect unauthorized users from harming database. Matching requirement is defined as **R** contiguous matching symbols in corresponding positions. As presented in [3], considering the fact that perfect matching is rare, the choice of (**RCB**) is mainly to simplify mathematical analysis with some flavor of immunology. In fact, such a partial match achieves generalization from limited self samples. That is an important characteristic of a learning algorithm. The value of **R** can be used to balance between more generalization and more specification. This rule has been the most popular one in the later works of negative selection algorithms.

Another matching algorithm that is considered an extension to (**RCB**) algorithm is the **r-chunk** detector. The r-chunk detector is a string of r bits together with a specific window. The detector **d** is said to match a string **x** if all bits of **d** are equal to the **r** bits of **x** in the window specified by **d**. The difference between **RCB** algorithm and r-chunk algorithm boils down to the fact that the matching window is specified for each individual detector. A group of **r-chunk** detectors that cover all possible windows has the same effect as an (**RCB**) algorithm.

After completing the previous operations, the super administrator (**SA**) submits all operations as a request to the database server. The database server respond to the super administrator (**SA**) with a confirmation message that all operations are carried out, or none are.

3.2 Database Administrators (DBAs)

The database administrator (**DBA**) is the second level of the security hierarchy. Once the super administrator (**SA**) stores all database administrators' accounts, each database administrator can now access the database security system separately using his authentication parameters: ***username***, ***password***, and ***secret key certificate***. Once the database administrator (**DBA)** proceeds with the system, his operations can be executed as followed:

3.2.1 Building Database Users

The first operation of the database administrator (**DBA**) is to create all database users who can connect to the database security system. The database administrator (**DBA**) must provide four parameters of information for each created user. These parameters are explained as followed:

- Login Information

The login parameters such as username and password for each user will be used during the detection phase of the security system. These parameters will be the first security layer to prevent unauthorized users from breaching the system.

- User Certificate Authorization (UCA)

The database administrator (**DBA**) specifies a secret key certificate for each created user to be used as a final countermeasure if unauthorized users succeed in breaching the security system defenses. The user certificate authorization (UCA) is a secret certificate encrypted using 128 bit AES encryption algorithm and is stored in the database server.

- User Profile Classification

The database administrator (**DBA**) classifies the user according to his activity in the system to be an *Active*, *Intermediate*, or *Inactive* user. Once the user classification is granted to each user, his privileges must comply with all profile authorizations that have been created by the super administrator (**SA**) as presented in section 3.1.

3.2.2 Building Database Roles

The second operation of the database administrator (**DBA**) is to build database roles that will be granted to the created users. Each database administrator (**DBA**) must determine the following two parameters:

- System Privileges

The database administrator (**DBA**) specifies which DML operations: insert, update, delete, and select privileges that can be granted to each role.

- Database Privileges

The database administrator (**DBA**) specifies which database privileges will be specified for each role. The set of database privileges are explained in Table 1. Once the database administrator (**DBA**) specifies the system and database privileges for the created roles, each danger value signal (**DVS**) for each privilege that has been created by the super administrator (**SA**) in section 3.1.1 must be triggered to be stored in the database server for the detection phase. If the user tries to change or violate his privileges, the danger value threshold will raise an alarm.

3.2.3 Granting Roles

The third operation of database administrator (**DBA**) is to grant created database roles to the created legitimate users. The same role can be granted to different users and vice versa.

3.2.4 Building User Authorization

The last process for database administrator (**DBA**) has two main stages:

- Determine the authorizations for each user and determine whether there are any sensitive attributes that must be hidden from the user.
- Fill in the actual parameters to be granted to each user in the authorization block to determine the number of (Insert - Update – Delete – Select) operations allowed for each user. These parameters must comply with the classification parameters that have been inserted by the super administrator.

It is necessary to distinguish between the main DML parameters stored by the super administrator (**SA**) and the actual DML parameters to be granted to each user by the database administrator (**DBA**). The database administrator (**DBA**) must take into consideration the main DML parameters created by the super administrator (**SA**) as presented in section 3.1. For example, if the database administrator (**DBA**) passes the parameter for an active user (Max Insert = 12). The database server will raise an error because the inserted parameter violates with the stored parameter which must be between 15 and 20 insert operations.

3.2 Legitimate Users

Once the database administrator (**DBA**) finished building user profiles and grants user privileges, the operations are sent to the database server as a request and the database server sends a confirmation that the operations are stored. Each user can proceed his own session with his pre-defined privileges. If there are any deviations from database administrators or legitimate users from their pre-defined privileges, the system will record anomalous behavior and will raise an alarm.

4. USER DETECTION ALGORITHM

The proposed detection algorithm can be used as a countermeasure to detect and prevent intruders or malicious users from performing hostile acts. Based on the previous section into which the database security learning mechanism collects information about the super administrator (SA), database administrator (DBA), and legitimate users; the created authorized users are considered now responsible for all database transactions through database server.

4.1 Intruder Recognition

The proposed intruder recognition is based on two separate processes. These processes are explained as followed:

4.1.1 Captcha

Captcha is a challenge-response test used in computing an attempt to ensure whether the response is generated by a human being or not. The process usually involves a computer asking a user to complete a simple test. This test is designed to be easy for a computer to generate but difficult for a computer to solve. If a correct solution is received, it can be presumed to have been entered by a human.

4.1.2 Access Control

The security system requires a username and password from any user entering the system to classify and filter out intrusion attempts. The username and password for each authorized user must be defined by the database administrator (DBA)

during user creation as presented in section 3.2.1 and must be known to the legitimate users only.

Algorithm 1: Intruder Recognition

```
1. If Captcha is valid Then
2.  {
3.  Get all usernames and passwords from Security. Users
4.  Put usernames and passwords into system cache
5.    If entered username and password is authentic Then
6.     {
7.        Go to Danger Signal II
8.     }
9.    Else
10.    {
11.       Raise Danger Signal I Alarm
12.    }
13. }
```

As presented in Algorithm 1, the main mechanism is to check for a human being interaction using Captcha system. This mechanism is extremely crucial to eliminate any computer-machine generators. If the Captcha entry is valid, the security system brings all usernames and passwords that have been created by database administrators (DBAs) from the database server and puts them into system cache. If the user enters authentic username and password, the security system will move from danger signal I alarm to danger signal II alarm. This means that the user may or may not be an intruder. If the user entry is invalid, the security system will initiate danger signal I alarm. Danger signal II algorithm is the second layer of security for detecting unauthorized users. This algorithm will be implemented based on the danger value signal. Danger signal II algorithm is under implementation and testing in order to be merged with the first algorithm and will be presented in future researches.

5. ARTIFICIAL IMMUNITY FEATURES

This paper is based on developing a database security system which is inspired from the features of artificial immune system. By updating the features of artificial immunity, the developed security system will be more flexible, adaptable, efficient, and secured.

5.1 Artificial Immune System Attributes

The artificial immune system (AIS) has different theoretical properties that have not been experimented yet. These properties are self-maintenance, adaptability, communication, and self-tolerance [2]. These properties have been enhanced and embedded into the developed security system to achieve high performance and flexibility. The enhanced properties are explained as followed:

- **Self-Maintenance**

  The developed security system has the ability to afford large number of normal users (self users) with the ability to achieve high detection rate with low false positive and low false negative alarms.

- **Communication**

  When the database administrator (DBA) creates normal users and grants privileges to them, the security system must perform a secret sharing communication with other database administrators (DBAs) to permit or deny this request. This means that a single database administrator (DBA) cannot lead the operation alone. The secret sharing communication can avoid the probability that the database administrator may be the intruder.

- **Adaptability**

  Adaptability is considered one of the major properties of artificial immune system (AIS). When the database administrator (DBA) creates a self user with his privileges, the developed security system can permit the database administrator to modify his privileges without affecting the performance of the system.

- **Self-Tolerance**

  The developed security system has the ability to afford large number of detected transactions (non-self users) in its memory without affecting the capability of the system.

5.2 Security System Major Characteristics

- **Frequency – reflecting data**

  The developed artificial immune security system has the ability to collect all database privileges in order to create a complete learning mechanism. It has the ability to add any new privilege in the system to be granted to different self users.

- **Noisy data**

  The collected dataset in the developed artificial immune security system has not been created randomly. Instead, the dataset can be collected by using a harmonious mechanism between super administrator, database administrators, and self users. The collected data is presented in the form of user privileges that must be granted to different authorized users based on the acceptance of the super administrator and database administrators. This can eliminate noisy data.

6. SUMMARY

Database security mechanisms and techniques remain important goals in any data management systems whether the systems are commercial, industrial, educational, medical, or even military systems. The main goal of this paper is to develop a database security system based on danger theory. Danger theory is one of the most recent algorithms of artificial immunity that can provide interactive features for securing relational databases. By merging the developed features of artificial immunity to the security system, the secrecy of the database can be maintained.

7. REFERENCES

[1] Dasgupta, D., Ya, S., and Nino, F., “Recent Advances in Artificial Immune Systems: Models and Applications,” International Journal of Applied Soft Computing, ELSEVIER, 2011

[2] Ming Ou, C., “Host-based Intrusion Detection Systems Adapted from Agent-based Artificial Immune Systems,” International Journal of Neurocomputing, ELSEVIER, 2012

[3] Ji, Z., and Dasgupta, D., “Revisiting Negative Selection Algorithms,” International Journal of Evolutionary

Computation, Massachusetts Institute of Technology, 2007

[4] Forrest, S., Perelson, A., Allen, L., and Cherukuri, R., "Self-Nonself Discrimination in a Computer," IEEE Symposium on Research in Security and Privacy, 1994

[5] Hashem, M., El-Henawy, I., and Mostafa, A., "Interactive Multi-Layer Policies for Securing Relational Databases," IEEE International Conference on Information Society, UK, 2012

[6] Hosseinpour, F., Abu Bakar, K., Hardoroudi, A., and Kazazi, N., "Survey on artificial immune system as a bio-inspired technique for anomaly based intrusion detection system," IEEE International Conference on Intelligent Networking and Collaborative Systems, 2010

[7] Hashem, M., El-Henawy, I., and Mostafa, A., "Design and Implementation of Multi-Layer Policies for Database Security," International Conference on Security and Management, USA, 2012

[8] Mostafa, A., Hashem, M., and El-Henawy, I., "Design and Implementation of Extensible Service-Oriented Algorithms for Securing Relational Databases," International Journal of Digital Content Technology and its Applications JDCTA, ELSEVIER, 2013

[9] Stibor, T., Mohr, Ph., and Timmis, J., "Is Negative Selection Appropriate for Anomaly Detection?," ACM International Conference on Generic and Evolutionary Computation, 2005

[10] Luo, W., Wang, X., Tan, Y., and Wang, X., "A Novel Negative Selection Algorithm with an Array of Partial Matching Lengths for each Detector," 9th International Conference on Parallel Problem Solving from Nature, Springer, 2006

[11] Luo, W., Wang, J., and Wang, X., "Evolutionary negative selection algorithms for anomaly detection," 8th International Conference on Information Sciences, 2005

[12] J. Timmis, J., Neal, M., and Hunt, J., "An artificial immune system for data analysis," International Journal of Biosystems, ELSEVIER, 2000

[13] Zhang, C., and Yi, Z., "An Artificial Immune Network Model Applied to Data Clustering and Classification," 4th International Conference on Neural Networks, Springer, 2007

[14] de Castro, L.N., and Von Zuben, F.J., "Learning and optimization using the clonal selection principle," IEEE Transactions on Evolutionary Computations, 2002

[15] Al-Enezi, J.R., Abbod, M.F., and Alsharhan, S., "Artificial Immune Systems- models, Algorithms, and Applications," International Journal of Research and Reviews in Applied Sciences, 2010

[16] Coelho, G.P, and Zuben, F.J.V., "Omni-aiNet: An Immune-inspired Approach for Omni-optimization," 5th International Conference on Artificial Immune Systems, Springer, 2006