International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-4 Number-2 Issue-15 June-2014

Attack Penetration System for SQL Injection

Manju Kaushik[1], Gazal Ojha[2]

Abstract

The SQL Injection attack is a popular way of attack in terms of document structure and common threats now a day. There are several ways of attack detection as per our study and also prevention methods had been discussed in several research papers. So the main motivation of our paper to penetrate the attack. For this we have proposed an efficient framework from which the central authority can control all the visited IP and restrict those IP. If the IP is in the restricted zone then the contents are restricted by using SQL update command with some status modification states. If the restricted IP wish to access the data from their credentials, it is immediately inform to the admin and the time of attack alert will be stored in the log area of that attack. Finally by comparison we will justify our results which are better in comparison to the previous test and methodology.

Keywords

SQL injection attack, attack detection, attack prevention, Restricted IP

1. Introduction

There are lot of attacks with different intension can be happen in the internet world. The challenging and most threating attack is SQL Injection attack [1]. In this attack the attacker can gain access the data, by fooling authentication mechanisms, for the purpose of alteration and to execute arbitrary code [2]. There is several methodologies and algorithm are suggested in [3], [4], [5], [6], [7], [8], [9], but there is need of enhancement in the said field. In [10] author suggested that instantaneously a dissonant and host level entry point is fully secured; the depose interface uncovered by a fascination becomes the only source of Feign. SQL Injection Attack can be used by kindred who scarcity to carry out access to the database and steal, change or delete data for which they do not have permission. In [11] different techniques was proposed to provide a solution for SQLIAs (SQL Injection Attacks), but many of these solutions have limitations that affect their effectiveness and practicability.

Encryption and decryption of the data in the communication channel are also helpful for protecting the data. For encryption and decryption we can use DES, RSA, RC4 and RC5 algorithms [12]. Block based division can be possible with subset superset mining or partitioning techniques [13][14] It is also useful in the scene where the sending data and the wrapper will be different so that confusion will be increases and the security in the receiving side will be more imposed. In cryptography we perform encryption on the original text to create the cipher text and decryption is just an opposite mechanism to form the plaintext. In steganography we hide the original plaintext within any other, text, PDF, images etc. The mechanism of reading the original text will be separately sent to the receiver for data reading. Cryptography is used to change the original plain text to encode or make unreadable form of text [15]. The excruciating materials are clandestine on the sender comrade in order to have them secluded and spellbound from illicit access and then sent via the network. When the data are received then the opposite process will be employed for decryption depending on an algorithm. Decryption is the process of converting data from encrypted format back to their original format [16][17][18].

In the SQL attack the attacker can apply the insertion or "injection" of either a partial or complete SQL query via the data input or transmitted from the client (browser) to the web application. If it will be successful then insertion in the unauthorized area, deletion, and updation can be possible without the permission of the legitimate user. So it is a serious threat and we need some solution in this regard to prevent it. For prevention we first need proper detection so that we get the timely alert and recognize the attack. SQL statements can be constructed in various ways and the string form data will be prevented a encryption technique with proper SQL parser to retrieve it and find it suitable in the case of

Manuscript received June 10, 2014.
Manju Kaushik, Department of Computer Science and Engineering, JECRC University, Jaipur, India.
Gazal Ojha, Department of Computer Science and Engineering, JECRC University, Jaipur, India.

matching the SQL parser. Then after the short analysis we have to plan a log file to maintain it so that exact comparison will be possible and we find the malicious content.

We provide here a brief survey and efficient penetration technique. Other sections are arranged in the following manner: Section 2 describes about Literature Review; Section 3 discusses about proposed work; section 4 shows the result analysis; Section 5 describes Conclusions.

2. Literature Review

In 2006, Ke Wei et al. [19] suggest that by using SQL injection attacks, an attacker could thus obtain and/or modify confidential/sensitive information. They also suggest that an attacker could even use a SQL injection vulnerability as a rudimentary IP/Port scanner of the internal corporate network. There are very little emphasis is laid on securing stored procedures in the database layer which could also suffer from SQL injection attacks. As stored procedures reside on the database front, the methods proposed by them cannot be applied to secure stored procedures themselves. They proposed a novel technique to defend against the attacks targeted at stored procedures. This technique combines static application code analysis with runtime validation to eliminate the occurrence of such attacks. In the static part, they design a stored procedure parser, and for any SQL statement which depends on user inputs, they use this parser to instrument the necessary statements in order to compare the original SQL statement structure to that including user inputs. The deployment of this technique can be automated and used on a need-only basis.

In 2008, Mehdi Kiani et al.[20] describe an anomaly based approach which utilizes the character distribution of certain sections of HTTP requests to detect previously unseen SQL injection attacks. Their approach requires no user interaction, and no modification of, or access to, either the backend database or the source code of the web application itself. Their practical results suggest that the model proposed in this paper is superior to existing models at detecting SQL injection attacks. They also evaluate the effectiveness of their model at detecting different types of SQL injection attacks.

In 2010, Cristian Pinzón et al. [21] presents a hybrid approach based on the Adaptive Intelligent Intrusion Detector Agent (AIIDA-SQL) for the detection of those attacks. The AIIDA-SQL agent incorporates a Case-Based Reasoning.
(CBR) engine which is equipped with learning and adaptation capabilities for the classification of SQL queries and detection of malicious user requests. To carry out the tasks of attack classification and detection, the agent incorporates advanced algorithms in the reasoning cycle stages. Concretely, an innovative classification model based on a mixture of an Artificial Neuronal Network together with a Support Vector Machine is applied in the reuse stage of the CBR cycle. This strategy enables to classify the received SQL queries in a reliable way. Finally, a projection neural technique is incorporated, which notably eases the revision stage carried out by human experts in the case of suspicious queries. Their experimental results obtained on a real-traffic case study show that AIIDA-SQL performs remarkably well in practice.

In 2010, Atefeh Tajpour et al. [22] suggest that Database driven web application are threaten by SQL Injection Attacks (SQLIAs) because this type of attack can compromise confidentiality and integrity of information in databases. Actually, an attacker intrudes to the web application database and consequently, access to data. For stopping this type of attack different approaches have been proposed by researchers but they are not enough because usually they have limitations. Indeed, some of these approaches have not implemented yet and also most of implemented approaches cannot stop all type of attacks. Authors evaluate these approaches against all types of SQL injection attacks and deployment requirements.

In 2010, Ivano Alessandro Elia et al. [23] present an experimental evaluation of the effectiveness of five SQL Injection detection tools that operate at different system levels: Application, Database and Network. To test the tools in a realistic scenario, Vulnerability and Attack Injection is applied in a setup based on three web applications of different sizes and complexities. Results show that the assessed tools have a very low effectiveness and only perform well under specific circumstances, which highlight the limitations of current intrusion detection tools in detecting SQL Injection attacks. Based on experimental observations they underline the strengths and weaknesses of the tools assessed.

In 2011, Kai-Xiang Zhang et al. [24] suggest SQL injection attacks, a class of injection flaw in which specially crafted input strings leads to illegal queries to databases, are one of the topmost threats to web applications. Based on their observation that the injected string in a SQL injection attack is interpreted differently on different databases, they propose a novel and effective solution TransSQL to solve this problem. TransSQL automatically translates a SQL request to a LDAP-equivalent request. After queries are executed on a SQL database and a LDAP one, TransSQL checks the difference in responses between a SQL database and a LDAP one to detect and block SQL injection attacks. Their Experimental results show that TransSQL is an effective and efficient solution against SQL injection attacks.

In 2012, Ramya Dharam et al. [25] present a framework which can be used to handle tautology based SQL Injection Attacks using post-deployment monitoring technique. Their framework uses two pre-deployment testing techniques i.e. basis path and data flow testing techniques to identify legal execution paths of the software. Runtime monitors are then developed and integrated to observe the behavior of the software for identified execution paths such that their violation will help to detect and prevent tautology based SQL Injection Attacks.

In 2012, XI-Rong Wu et al. [26] proposed a new method named k-centers (KC) to detect SQL injection attacks (SQLIAs). The number and the centers of the clusters in KC are adjusted according to unseen SQL statements in the adversarial environment, in which the types of attacks are changed after a period of time, to adapt different kinds of attacks. The experimental results show that the proposed method has a satisfying result on the SQLIAs detection in the adversarial environment.

In 2012, WAN Min et al. [27] suggest that Web applications have brought with them new classes of network security vulnerabilities, such as SQL Injection Attack. SQL Injection Attack is a class of attacks that many of the Webs based systems are highly vulnerable to, and there is no know fool-proof defense against such attacks. Static analysis is one of the techniques in defense of SQL Injection. They proposed an improved technique eliminates the need to modify source code of application scripts. The improved Eliminating SQL Injection Attacks technique bases the regular expressions instead of using SQL Graph representation using SQL-FSM in static analysis.

In 2012, TIAN Wei et al. [28]discuss how to generate more effective penetration test case inputs to detect the SQL injection vulnerability hidden behind the inadequate blacklist filter defense mechanism in web applications. They propose a model based penetration test method for the SQL injection vulnerability, in which the penetration test case generation is divided into two steps: i) Building model for the penetration test case, and ii) Instantiating the model of penetration test case. Their method can generate test case covering more types and patterns of SQL injection attack input to thoroughly test the blacklist filter mechanism of web applications. Their Experiments show the penetration test case generated by their method can effectively find the SQL injection vulnerabilities hidden behind the inadequate blacklist filter defense mechanism thus reduce the false negative and improve test accuracy.

In 2013, Amirmohammad Sadeghian et al. [29] suggest that a successful SQL injection attack interfere Confidentiality, Integrity and availability of information in the database. Based on the statistical researches this type of attack had a high impact on business. Finding the proper solution to stop or mitigate the SQL injection is necessary. To address this problem security researchers introduce different techniques to develop secure codes, prevent SQL injection attacks and detect them. They present a comprehensive review of different types of SQL injection detection and prevention techniques. They criticize strengths and weaknesses of each technique. Such a structural classification would further help other researchers to choose the right technique for the further studies.

In 2013, Amirmohammad Sadeghian et al. [30] suggest SQL injection is one of the biggest challenges for the web application security. Based on the studies by OWASP, SQL injection has the highest rank in the web based vulnerabilities. In case of a successful SQL injection attack, the attacker can have access to the web application database. With the rapid rise of SQL injection based attacks, researchers start to provide different security solutions to protect web application against them. One of the most common solutions is the using of web application firewalls. Usually these firewalls use signature based technique as the main core for the detection. In this technique the firewall checks each packet against a

list of predefined SQL injection attacks known as signatures. The problem with this technique according to the author is that, an attacker with a good knowledge of SQL language can change the look of the SQL queries in a way that firewall cannot detect them but still they lead to the same malicious results. Authors described the nature of SQL injection attack, then they analyzed current SQL injection detection evasion techniques and how they can bypass the detection filters, afterward they proposed a combination of solutions which helps to mitigate the risk of SQL injection attack.

In 2013, Amirmohammad Sadeghian et al. [31] first they provided background information on this vulnerability. Next they present a comprehensive review of different types of SQL injection attack. For each attack they provide an example that shows how the attack launches. Finally they propose the best solution at development phase to defeat SQL injection and conclusion.

3. Proposed Methodology

In this paper, we propose an effective and flexible SQL Injection detection mechanism which is control from the admin. Our proposed methodology provides secure centralized control system with blocking of data system on the restricted IP. Our approach achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving clients and it can be control by the Admin. It can provide data accessing on the fly if the IP is in unblock list. This flexibility is archived by remote consistency through Java Remote Method Invocation (RMI). The data status is changed runtime and the instance is changed automatically. If the data accessing is stopped from the admin then any remote SQL injection will be failed as the used tokens from SQL server is topped and the penetration system will start working. The working procedure is better understand by figure 1.

Phases

Client Phase

Authentic client registered from the admin is capable of accessing data blocks. Based on the query the admin adds the corresponding file on the basis of the IP not on the basis of the client. It provides the flexibility to block/unblock the IP on the fly mechanism. Because the data belongs to the current IP but not to the client. So the status will also change according to the restriction. Means if the belonging IP will be blacklisted than the data status will be automatically updated.

Admin Phase

Data is updated and the logs will be maintained by the Admin. The Admin provides the data in the same set of specified regions. The data operation will be needed by the users in some cases to make it authentic. Admin should be equipped with security means so that they can assure with their copies that there copies are not be put in violation or any unauthorized means not connect with their data. . Then the facilities provided by the admin or the central authority are the updating of client lists, causing a reduction in authentication time and timely validation of that IP which are restricted by the Admin.

The procedures are dividing into four different phase as shown below:

RMI Interphase:

```
//Client Interphase
Step 1: public String CliDet(String a, String b,  String c)
//User and key are matched
Step 2: public String CliDet_key(String user,String key)
//String file name and IP are matched
public File res(String filename,  String ip)
//String file name and IP are matched with the option
public String res(String filename,  String ip,String fileop)
// IP is Restricted
public String checkipp(String ip)
```

Server Interphase:

```
//Server Socket
ServerSocket svc=new ServerSocket(4567);
//Server Socket acceptance
Socket sock=svc.accept();
//Connection to the IP address
 ("connected to "+sock.getInetAddress());
//Database Connection
 dbcon db=new dbcon();
```

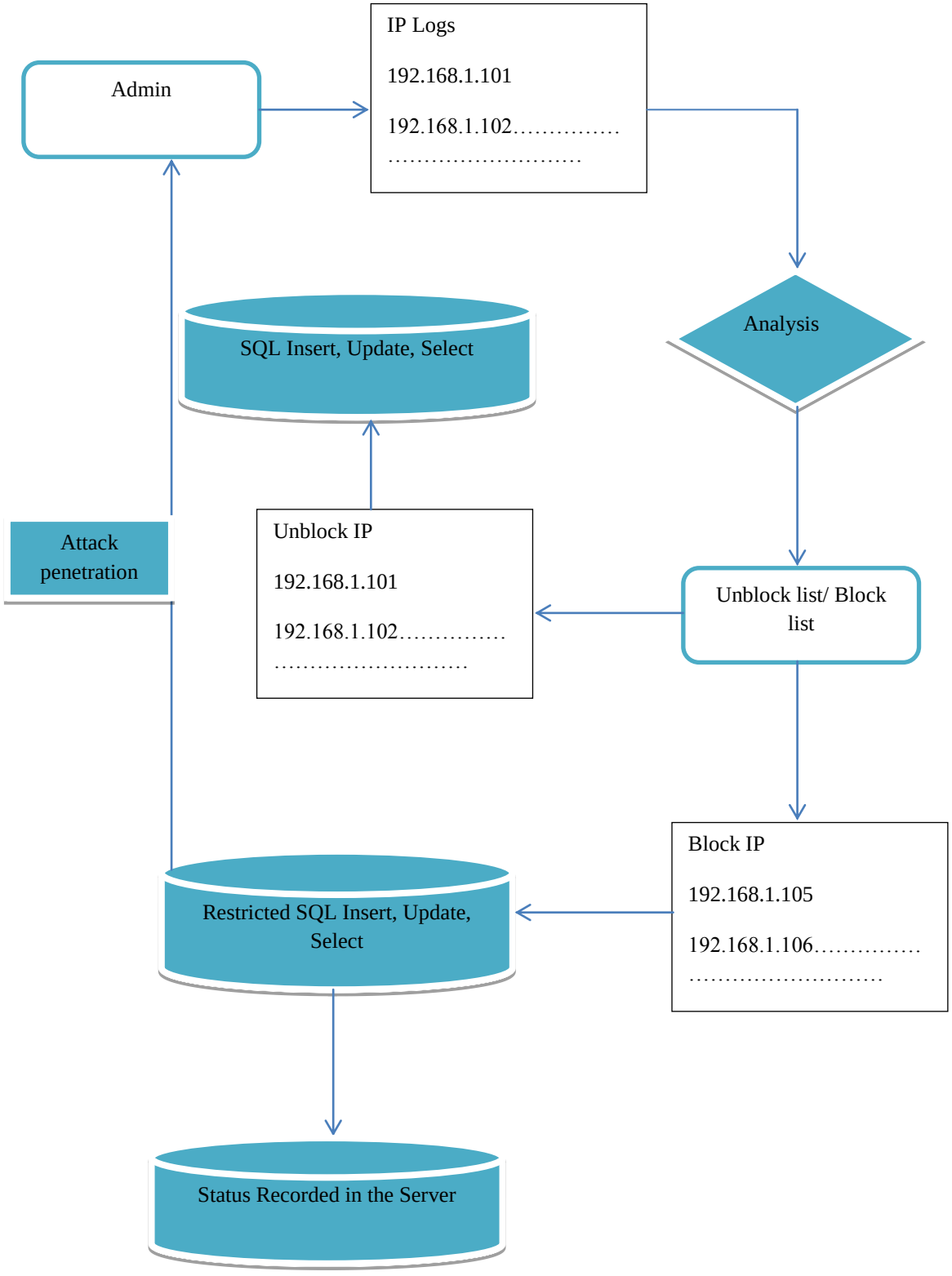

Figure 1: Attack Penetration System

4. Result Analysis

In this section we are discussing our results. The admin function is shown in figure 2. The SQl statement Injection details are shown in the figure 3. Figure 4 and Figure 5 shows the Attack Penetration difference, which is improved by our methodology, it shows the effectiveness of our approach.

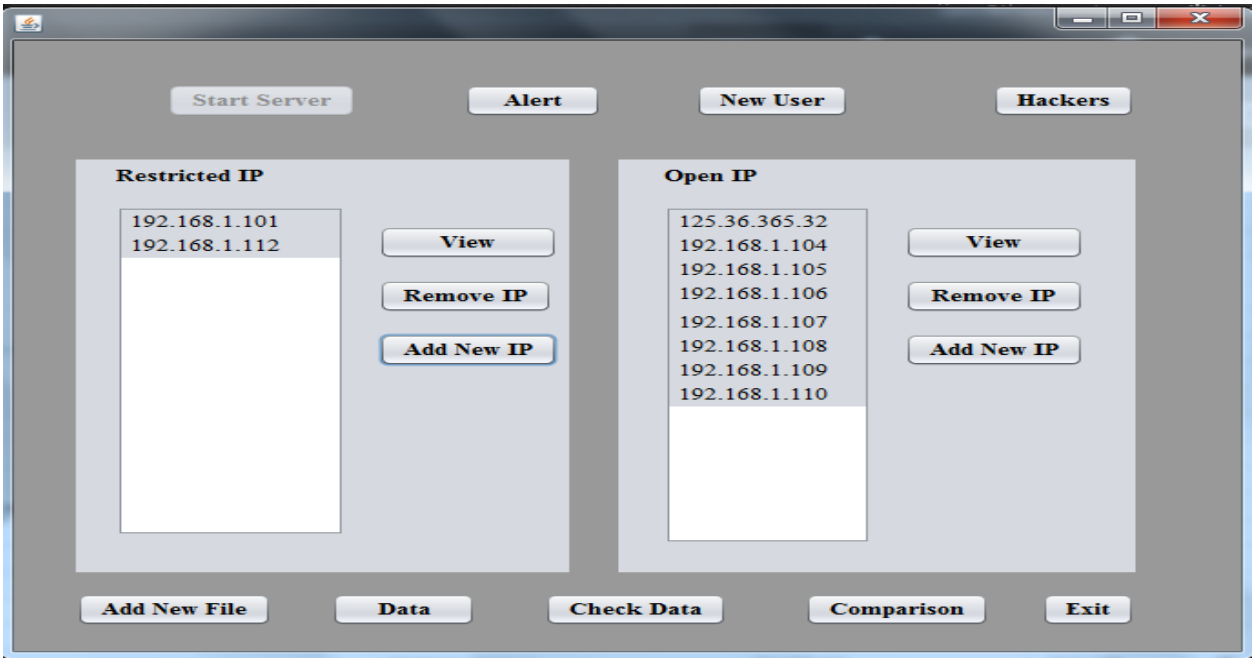

Figure 2: Admin Phase

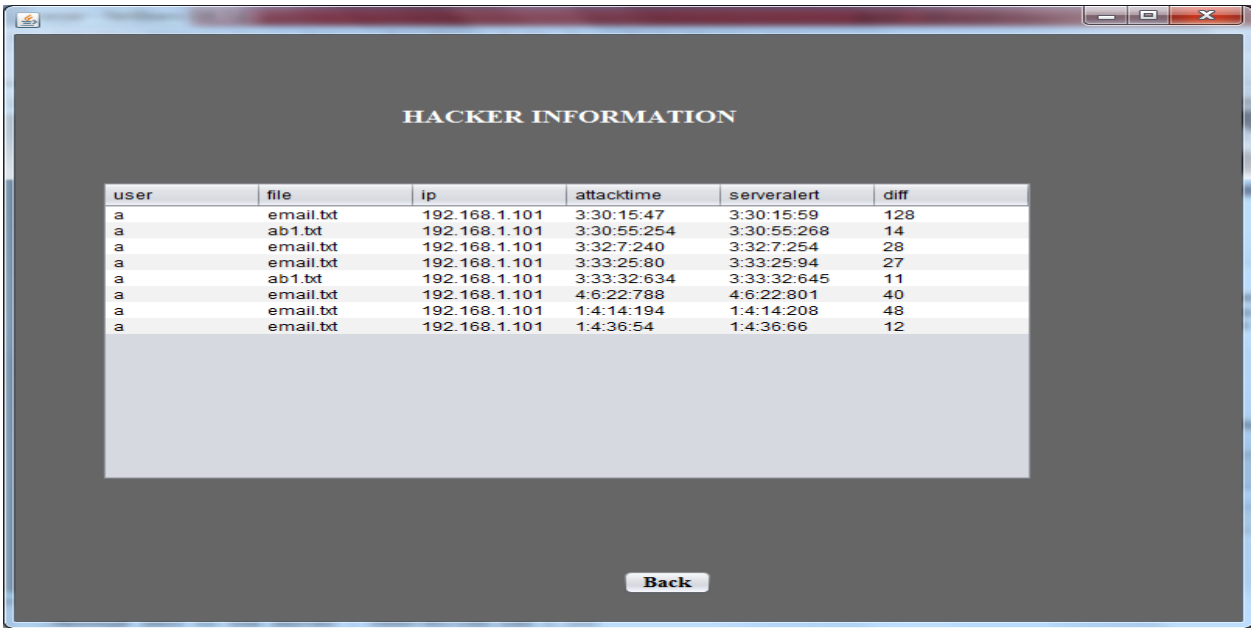

Figure 3: SQL Injection Information

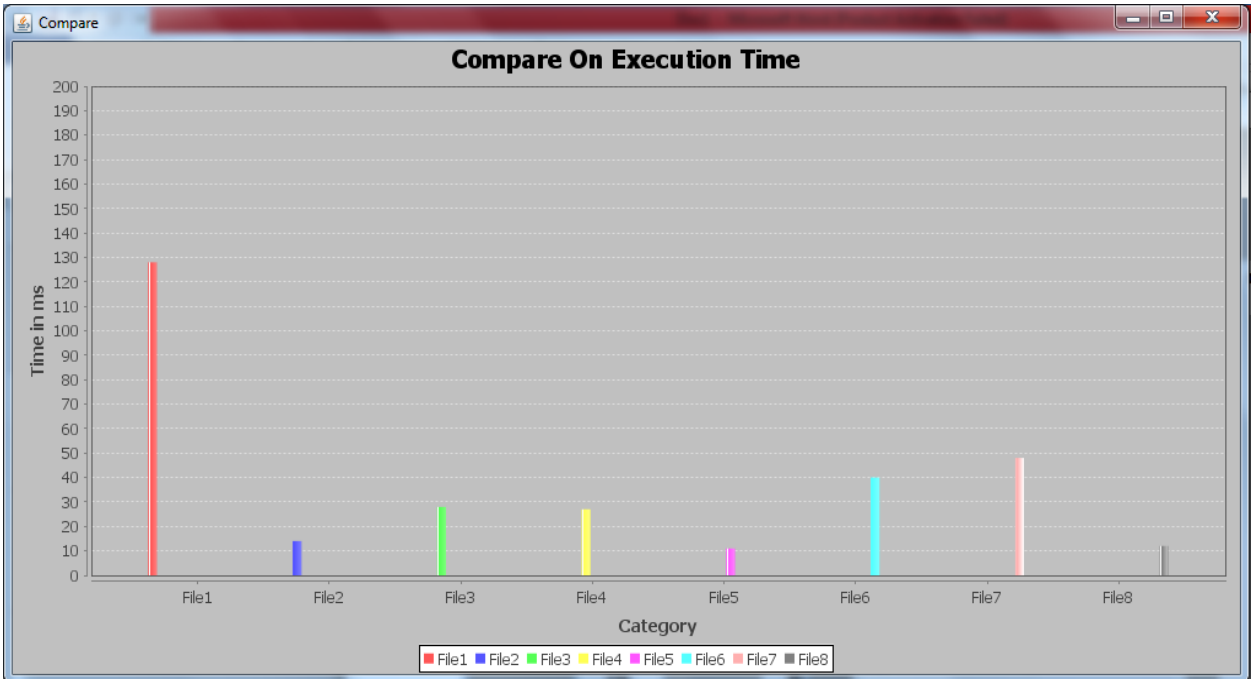

Figure 4: Compare on Execution time

Figure 5: Compare on Execution time

5. Conclusion

In this paper we survey and analyse different SQL Injection attacks used in the previous techniques as well as different cryptography and steganography mechanism. We also present an efficient RMI based SQL attack detection mechanism and the time penetration. Our methodology shows better time penetration in comparison to the previous methodology.

References

[1] W. G. J. Halfond, et al., "A Classification of SQL-Injection Attacks and Countermeasures," in Proceedings of the IEEE International Symposium on Secure Software Engineering, Arlington, VA, USA,2006.

[2] A. Asmawi, Sidek Zailani Mohamed Razak Shukor Abd, "System architecture for SQL injection and insider misuse detection system for DBMS," in International Symposium on Information Technology (ITSim'2008), 2008, pp. 1 -6.

[3] C. Bockermann, et al., "Learning SQL for Database Intrusion Dete ction Using Context-Sensitive Modelling (Extended Abstract)," in 6th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA '09), Berlin, Heidelberg, 2009, pp. 196--205.

[4] K. Kemalis and T. Tzouramanis, "SQL-IDS: a specification-based approach for SQL-injection detection," in Proceedings of the 2008 ACM symposium on Applied computing (SAC'2008), New York, NY,USA, 2008, pp. 2153--2158.

[5] M. Kiani, et al., "Evaluation of Anomaly Based Character Distribution Models in the Detection of SQL Injection Attacks," in Third International Conference on Availability, Reliability and Security (ARES'2008), Washington, DC, USA, 2008, pp. 47--55.

[6] E. Bertino, et al., "Profiling Database Applications to Detect SQL Injection Attacks," in Proceedings of the Performance, Computing, and Communications Conference (IPCCC'2007), 2007, pp. 449-458.

[7] W. Robertson, et al., "Using Generalization and Characterization Techniques in the Anomaly-Based Detection of Web Attacks," in 13th Annual Network and Distributed System Security Symposium (NDSS'2006), 2006.

[8] V. H. García, et al., "Web Attack Detection Using ID3," in International Federation for Information Processing 2006, pp. 323- 332.

[9] F. Valeur, et al., "A Learning-Based Approach to the Detection of SQL Attacks," in Proceedings of the Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA), Vienna, Austria, 2005, pp. 123--140.

[10] R. Ezumalai and G. Aghila. Combinatorial Approach for Preventing SQL Injection Attacks. IACC, 2009.

[11] Junjin, Mei. "An approach for SQL injection vulnerability detection." In Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on, pp. 1411-1414. IEEE, 2009.

[12] Ashutosh Kumar Dubey, Animesh Kumar Dubey, Mayank Namdev, Shiv Shakti Shrivastava,"Cloud-User Security Based on RSA and MD5 Algorithm for Resource Attestation and Sharing in Java Environment", CONSEG 2012.

[13] Ashutosh Kumar Dubey, Animesh Kumar Dubey, Vipul Agarwal, Yogeshver Khandagre, "Knowledge Discovery with a Subset-Superset Approach for Mining Heterogeneous Data with Dynamic Support",Conseg-2012.

[14] Preeti Khare, Hitesh Gupta, "Finding Frequent Pattern with Transaction and Occurrences based on Density Minimum Support Distribution", International Journal of Advanced Computer Research (IJACR), Volume-2 Number-3 Issue-5 September-2012.

[15] Lakhtaria, Kamaljit I. "Protecting computer network with encryption technique: A Study." In Ubiquitous Computing and Multimedia Applications, pp. 381-390. Springer Berlin Heidelberg, 2011.

[16] Chan, Aldar CF, and Claude Castelluccia. "A security framework for privacy-preserving data aggregation in wireless sensor networks." ACM Transactions on Sensor Networks (TOSN) 7, no. 4 (2011): 29.

[17] Stalling, W.,Cryptography and network security principles and practices ,4th edition Prentice Hall,2005.

[18] Shannon, Claude E. "Communication Theory of Secrecy Systems*." Bell system technical journal 28, no. 4 (1949): 656-715.

[19] Ke Wei; Muthuprasanna, M.; Kothari, S., "Preventing SQL injection attacks in stored procedures," Software Engineering Conference, 2006. Australian, vol., no., pp.8 pp.,, 18-21 April 2006.

[20] Kiani, M.; Clark, A.; Mohay, G., "Evaluation of Anomaly Based Character Distribution Models in the Detection of SQL Injection Attacks," Availability, Reliability and Security, 2008. ARES 08. Third International Conference on , vol., no., pp.47,55, 4-7 March 2008.

[21] Pinzón, C.; De Paz, J.F.; Bajo, J.; Herrero, A.; Corchado, E., "AIIDA-SQL: An Adaptive Intelligent Intrusion Detector Agent for detecting SQL Injection attacks," Hybrid Intelligent Systems (HIS), 2010 10th International Conference on , vol., no., pp.73,78, 23-25 Aug. 2010.

[22] Tajpour, A.; JorJor Zade Shooshtari, M., "Evaluation of SQL Injection Detection and Prevention Techniques," Computational Intelligence, Communication Systems and Networks (CICSyN), 2010 Second International Conference on , vol., no., pp.216,221, 28-30 July 2010.

[23] Elia, I.A.; Fonseca, J.; Vieira, M., "Comparing SQL Injection Detection Tools Using Attack Injection: An Experimental Study," Software Reliability Engineering (ISSRE), 2010 IEEE 21st International Symposium on , vol., no., pp.289,298, 1-4 Nov. 2010.

[24] Kai-Xiang Zhang; Chia-Jun Lin; Shih-Jen Chen; Yanling Hwang; Hao-Lun Huang; Fu-Hau Hsu, "TransSQL: A Translation and Validation-Based Solution for SQL-injection Attacks," Robot, Vision and Signal Processing (RVSP), 2011 First International Conference on , vol., no., pp.248,251, 21-23 Nov. 2011.

[25] Dharam, R.; Shiva, S.G., "Runtime monitors for tautology based SQL injection attacks," Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on , vol., no., pp.253,258, 26-28 June 2012.

[26] Xi-Rong Wu; Chan, P.P.K., "SQL injection attacks detection in adversarial environments by k-centers,"

Machine Learning and Cybernetics (ICMLC), 2012 International Conference on , vol.1, no., pp.406,410, 15-17 July 2012.

[27] Wan Min; Liu Kun, "An Improved Eliminating SQL Injection Attacks Based Regular Expressions Matching," Control Engineering and Communication Technology (ICCECT), 2012 International Conference on , vol., no., pp.210,212, 7-9 Dec. 2012.

[28] Tian Wei; Yang Ju-Feng; Xu Jing; Si Guan-Nan, "Attack Model Based Penetration Test for SQL Injection Vulnerability," Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual , vol., no., pp.589,594, 16-20 July 2012.

[29] Sadeghian, A.; zamani, M.; Manaf, A.A., "A Taxonomy of SQL Injection Detection and Prevention Techniques," Informatics and Creative Multimedia (ICICM), 2013 International Conference on , vol., no., pp.53,56, 4-6 Sept. 2013.

[30] Sadeghian, A.; zamani, M.; Ibrahim, S., "SQL Injection Is Still Alive: A Study on SQL Injection Signature Evasion Techniques," Informatics and Creative Multimedia (ICICM), 2013 International Conference on, vol., no., pp.265, 268, 4-6 Sept. 2013.

[31] Sadeghian, A.; zamani, M.; Abdullah, S.M., "A Taxonomy of SQL Injection Attacks," Informatics and Creative Multimedia (ICICM), 2013 International Conference on , vol., no., pp.269,273, 4-6 Sept. 2013.

Dr. Manju Kaushik is presently working as Associate Professor in Department of Computer Science and Engineering at JECRC University, Jaipur. She is involved in Research activities in the field of Software Engineering. She was awarded Ph.D. degree from the Mohan Lal Sukhadiya University, Udaipur. She has published papers in various national/international journals. She has also attended many conferences/Seminars at national level.

Gazal Ojha pursuing her master degree in computer science and engineering from the JECRC University, Jaipur. She obtained her bachelor degree in Information Technology from Govt. Women Engineering College,Ajmer. Her research interests include-Software engineering, Database.