ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

SQL Injection Attack Detection and Prevention Methods: A Critical Review

Dr.Manju Kaushik[1],Gazal Ojha[2]

Associate Professor, Department of Computer Science and Engineering, JECRC University, Jaipur, India[1]

P.G. Student, Department of Computer Science and Engineering, JECRC University, Jaipur, India[2]

ABSTRACT: We have several news regarding data alteration in the communication background. 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. In this paper our main motivation to discuss and find the better methodology so that in future we can design better attack detection and prevention methods. For better security we also discuss encryption and decryption techniques.

KEYWORDS: SQL injection attack, attack detection, attack prevention, encryption and decryption techniques

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 (SQLInjection Attacks), but many of these solutions havelimitations 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 isused to change the original plain text to encode or make unreadableform of text [15]. The excruciating materials are clandestine on the sender comrade in order to have them secluded and

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

spellbound from illicit access and then sent via the network. When thedata are received then the opposite process will be employed for decryption depending on an algorithm.Decryption is the process of converting data from encryptedformat 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 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 on SQL Injection Attacks. Other sections are arranged in the following manner: Section 2 describes about Literature Review; Section 3discusses about problem domain; section 4 shows the analysis; Section 5 describes Conclusions and future work.

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 themodel proposed in this paper is superior to existingmodels at detecting SQL injection attacks. They also evaluate the effectiveness of their model at detectingdifferent types of SQL injection attacks.

In 2010, CristianPinzón 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

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

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, AtefehTajpour 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 todatabases, are one of the topmost threats to web applications. Based on theirobservation that the injected string in a SQL injection attack isinterpreted differently on different databases, they propose a novel and effective solution TransSQL to solve thisproblem. TransSQL automatically translates a SQL request toa LDAP-equivalent request. After queries are executed on aSQL database and a LDAP one, TransSQL checks thedifference in responses between a SQL database and a LDAPone to detect and block SQL injection attacks. Their Experimentalresults show that TransSQL is an effective and efficientsolution against SQL injection attacks.

In 2012, RamyaDharam 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.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

In 2012, WAN Min et al. [27] suggest that Web applications have brought with them new classesof network security vulnerabilities, such as SQL Injection Attack.SQL Injection Attack is a class of attacks that many of the Webs basedsystems are highly vulnerable to, and there is no knowfool-proof defense against such attacks. Static analysis is one ofthe techniques in defense of SQL Injection. They proposed an improved technique eliminates the need to modifysource code of application scripts. The improved EliminatingSQL Injection Attacks technique bases the regular expressionsinstead of using SQL Graph representation using SQL-FSM instatic 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, AmirmohammadSadeghian 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, AmirmohammadSadeghian 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, AmirmohammadSadeghian 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.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

3. PROBLEM DOMAIN

In [32] author suggest tokenizing the original queryand the query with injection, and after tokens are obtainedthey constitute arrays' elements. By comparing lengths ofthe resulting arrays from the two queries injection can bedetected. This method can be used with some other prevention technique so that it will be helpful in prevention and detection.

Table 1:Comparison of Techniques Based on Deployment Requirements [22]

Techniques	Modify Code Base	Detection	Prevention	Additional Infrastructure
Positive Training[33]	No	Auto	Auto	None
SQL Prevent [34]	No	Auto	Auto	None
Java Static Tainting [35]	No	Automated	Code Suggested	None
Waves [36]	No	Automated	Generated Report	None
SQLDOM [37]	Yes	N/A	Automated	Developer Training
SecuriFly [38]	No	Automated	Automated	None
Gateway [39]	No	Manual Specification	Automated	Proxy Filter

In [22] author provides the comparison as shown in table 1. In this they first identified the various types ofSQLIAs. Then they investigated on SQL injection detectionand prevention techniques. After that they compared thesetechniques in terms of their ability to stop SQLIA. Regardingthe results, some current techniques' ability should be
improved for stopping SQLI attacks. Moreover, they compared these approaches in deployment requirements that
lead to inconvenience for users. They suggest a future work to implement as tools then compare effectiveness,efficiency, stability, flexibility and performance of tools toshow the strength and weakness of the tool.

Table 2: Scalp Coverage Results [23]

Web Application	All attack attempts Coverage
Tikiwiki	13.33%
phpBB2	8.97%

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

MyReferences	22.45%

In [23] authors suggested that Attack Injection can be avery useful instrument to assess the detection ability ofintrusion detection tools in specific contexts and for specificweb applications. In practice, the technique can be extremelyuseful to assess the level of trust a system administrator canhave about the security tools installed to protect his system [23]. They suggest that creating the instruments needed tofully automate all the steps of the evaluation procedure thatwas used in their experiments in future. The aim should be to create astandard benchmark procedure for using Attack Injection asan advanced assessing instrument of the detection performance of security tools deployed in a web servingsystem.

Table 3: Proposed Model's User Privileges Table[30]

Username	Privileges
viewer_user	select
editor_user	Select, insert, update, delete
Sturucture_user	Create, alter, drop, execute
Super_user	Shutdown, grant , reload ,

In [3] authors conclude that using IDSs alone cannot be a sufficientsolution to protect the web application against SQL injection.While IDSs are helpful in detecting different types ofmalicious activities in the network, a combination of goodconfiguration of web server and using parameterized queriesin the coding phase can increase the protection against SQLinjection attacks.

Table 4: Testing Time of Each Method [28]

	JSP (min)	ASP (hour)	Size of test case
Tool A	39	4.3	45
Tool B	11	2.3	32
NKSI scan	32	3.6	103

In [28] authors proposed a model based penetration test method for the SQL injection vulnerability in web applications. Their experiment shows that compared with randomly enumerated test case, the test case generated by our formal method can detect more SQL injection vulnerabilities hidden behind the inadequate blacklist defense, and thus reduce the false negative of penetration test.

4. ANALYSIS

After studying several research papers we come with the following analysis:

1) Need of wrapper in different format or it may be in the form of images text bytes, pdf and spreadsheet, so that identification probability is become low.
2) Need of tokenizing the original query and the query with injection.
3) Need of reducing the detection time.
4) Cryptography and Steganography techniques can be used to enhance the security.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

5) Use Parameterized queries for making dynamic queries to avoid SQL injection [31].
6) Every added query can go for the matching parser to analyze the attack and a log should be maintained for this.

5. CONCLUSION AND FUTURE SUGGESTIONS

In this paper we survey and analyze different SQL Injection attacks used in the previous techniques as well as different cryptography and steganography mechanism. We also discuss the merits and some of the findings which will be incorporated to improve the security and reduces the chances of prediction. Based on our study we will suggest hybrid framework for tokenization and run time dynamic matching parser is needed to detect the attack and also reduce the time of alert.

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, SidekZailani Mohamed RazakShukorAbd, "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 Detection 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. MeiJunjin. An approach for SQL Injection vulnerability detection. IEEE,2009.
12. Ashutosh Kumar Dubey,Animesh Kumar Dubey, MayankNamdev, 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, VipulAgarwal, YogeshverKhandagre, "Knowledge Discovery with a Subset-Superset Approach for Mining Heterogeneous Data with Dynamic Support",Conseg-2012.
14. PreetiKhare, 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 K. (2011) Protecting computer network with encryption technique: A Study. International Journal of u- and e-service, Science and Technology 4(2).
16. Chan, A. (2011) A Security framework for privacy preserving data aggregation in wireless sensor networks. ACM transactions on sensor networks 7(4).
17. Stalling, W. (2005) Cryptography and network security principles and practices ,4th edition Prentice Hall.
18. Shannon, C. E. (1948) Communication Theory of secrecy systems. Bell System Technical Journal.
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.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 4, April 2014

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.; JorJorZadeShooshtari, 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.
32. Lambert, N.; Kang Song Lin, "Use of Query tokenization to detect and prevent SQL injection attacks," Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on , vol.2, no., pp.438,440, 9-11 July 2010.
33. W. G. Halfond, A. Orso, Using Positive Tainting and Syntax Aware Evaluation to Counter SQL Injection Attacks, 14th ACM SIGSOFT international symposium on Foundations of software engineering 2006, ACM. pp 175 – 185.
34. P.Grazie., PhD SQLPrevent thesis. University of British Columbia (UBC) Vancouver, Canada.2008.
35. V. B. Livshits and M. S. Lam. Finding Security Errors in Java Programs with Static Analysis. In Proceedings of the 14th Usenix Security Symposium, pp 271–286, Aug. 2005.
36. Y. Huang, S. Huang, T. Lin, and C. Tsai. Web Application Security Assessment by Fault Injection and Behavior Monitoring. In Proceedings of the 11th International World Wide Web Conference (WWW 03), May 2003.
37. R. McClure and I. Kr¨uger. SQL DOM: Compile Time Checking of Dynamic SQL Statements. In Proceedings of the 27th International Conference on Software Engineering (ICSE 05), pp 88–96, 2005.
38. M. Martin, B. Livshits, and M. S. Lam. Finding Application Errors and Security Flaws Using PQL: A Program Query Language. In Proceedings of the 20th Annual ACM SIGPLAN conference on Object oriented programming systems languages and applications (OOPSLA 2005), pp 365–383, 2005.
39. D. Scott and R. Sharp. Abstracting Application-level Web Security. In Proceedings of the 11th International Conference on the World Wide Web (WWW 2002), pp 396–407, 2002.