UNIVERSITI PUTRA MALAYSIA

TWO LEVEL SECURITY APPROACHES FOR SECURE XML DATABASE CENTRIC WEB SERVICES AGAINST XPATH INJECTIONS

AZIAH ASMAWI

FSKTM 2016 34

TWO LEVEL SECURITY APPROACHES FOR SECURE XML DATABASE CENTRIC WEB SERVICES AGAINST XPATH INJECTIONS

By

AZIAH ASMAWI

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia in fulfilment of the requirements for the Degree of Doctor of Philosophy

May 2016

DEDICATIONS

I dedicate this thesis to my precious family,

My husband, *Kamil Ikmal Kamarudin*
My mother, *Sharifah Abdullah*
and
My adorable kids, *Aqil Iman, Alya Ariana and Ammar Darwisy*

whose love, supports and the prayers has helped me complete this thesis.

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in
fulfilment of the requirement for the degree of Doctor of Philosophy

TWO LEVEL SECURITY APPROACHES FOR SECURE XML DATABASE CENTRIC WEB SERVICES AGAINST XPATH INJECTIONS

By

AZIAH ASMAWI

May 2016

Chairman : Associate Professor Lilly Suriani Affendey, PhD
Faculty : Computer Science and Information Technology

Web services are deployed using eXtensible Markup Language (XML), which is an independent language for easy transportation and storage. As an important transportation for data, Web services has become increasingly vulnerable to malicious attacks that could affect essential properties of information systems such as confidentiality, integrity, or availability. Like any other application that allows outside user submission data, Web services can be susceptible to code injection attacks, specifically XPath (XML Path Language) injection attacks. This kind of attack can cause serious damage to the database at the backend of Web services as well as the data within it. To cope with this attack, it is necessary to develop effective and efficient secure mechanism from various angles, outsider and insider. This thesis addresses both outsider and insider threats with respect to XPath injections in providing secure mechanism for XML database-centric Web services which yields the following significant contributions.

We propose the two level security approaches for the ultimate solution within XML database-centric Web services. The first approach focuses on preventing malicious XPath input within Web services application. In order to address issues of XPath injections, we propose a model-based validation (XIPS) for XPath injection attack prevention in Web service applications. The second approach focuses on preventing insider threat within XML database. In order to deal with insider threat, we propose a severity-aware trust-based access control model (XTrust) for malicious XPath code in XML database. A prototype of the solution and each approach was designed, implemented and evaluated using synthetic data through experimental research approach to evaluate its security performance. Evidently, result analysis proved that the two level security approaches solution able to provide overall protection for XML database centric Web services environment from outsider and insider threats with respect to XPath injections. Meanwhile, the first approach, XIPS provides alternative solution for Web service applications against malicious XPath input compared to the previous work and the second approach, XTrust provide more secure access control for XML database against malicious XPath code compared to the previous work. As a conclusion, the two

level security approaches solution improved security level in XML database-centric Web services.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia sebagai memenuhi keperluan untuk Ijazah Doktor Falsafah

PENDEKATAN DUA TAHAP KESELAMATAN BAGI KESELAMATAN LAMAN SESAWANG PERKHIDMATAN BERTERASKAN PANGKALAN DATA XML TERHADAP SUNTIKAN *XPATH*

Oleh

AZIAH ASMAWI

Mei 2016

Pengerusi : Prof. Madya Lilly Suriani Affendey, PhD
Fakulti : Sains Komputer dan Teknologi Maklumat

Laman sesawang perkhidmatan dibangunkan menggunakan *eXtensible Markup Language (XML)* untuk memudahkan pengangkutan dan penyimpanan. Sebagai pengangkutan yang penting untuk data, laman sesawang perkhidmatan telah menjadi semakin terancam dan terdedah kepada serangan jahat yang boleh memberi kesan kepada ciri khas penting di dalam sistem maklumat seperti kadar kerahsiaan, integriti dan kadar ketersediaan. Laman sesawang perkhidmatan juga seperti aplikasi lain yang membenarkan pengguna luar untuk menghantar data, boleh terdedah kepada serangan suntikan kod, khususnya serangan suntikan *XPath* (Bahasa *Path XML*). Serangan seperti ini boleh menyebabkan kerosakan yang serius kepada pangkalan data di sebalik laman sesawang perkhidmatan dan juga data di dalamnya. Untuk mengatasi serangan ini, adalah perlu untuk membangunkan mekanisma keselamatan yang efektif dan efisyen dari segenap sudut, luaran dan dalaman. Tesis ini mengalamatkan kedua-dua ancaman luaran dan dalaman yang berkaitan dengan suntikan *XPath* di dalam menyediakan mekanisma keselamatan untuk laman sesawang perkhidmatan berasaskan pangkalan data *XML*.

Kami menyarankan penyelesaian menggunakan pendekatan dua tahap keselamatan untuk penyelesaian yang muktamad di dalam laman sesawang perkhidmatan berasaskan pangkalan data *XML*. Pendekatan pertama menumpukan kepada pencegahan masukan data *XPath* berniat jahat di dalam aplikasi laman sesawang perkhidmatan. Bagi menangani isu ini, kami mencadangkan validasi berasaskan model *(XIPS)* untuk mencegah serangan suntikan *XPath* dalam aplikasi laman sesawang perkhidmatan.

Pendekatan kedua menumpukan kepada pencegahan serangan salahguna pengguna dalaman di dalam pangkalan data *XML*. Bagi menangani isu ini, kami mencadangkan model kawalan capaian berasaskan kepercayaan kesedaran-keterukan untuk serangan kod *XPath* berniat jahat di dalam prosedur simpanan pangkalan data *XML*. Prototaip untuk cadangan penyelesian dan setiap pendekatan telah direkabentuk, dilaksanakan dan

dinilai menggunakan data sintetik menerusi pendekatan penyelidikan bereksperimen untuk menilai pencapaian keselamatan setiap cadangan. Ternyata, analisis keputusan membuktikan penyelesaian menggunakan pendekatan dua tahap keselamatan berupaya menyediakan perlindungan kepada laman sesawang perkhidmatan berasaskan pangkalan data *XML* secara keseluruhannya daripada ancaman luaran dan dalaman yang berkaitan dengan suntikan *XPath*. Sementara itu, pendekatan pertama, XIPS menyediakan penyelesaian alternatif untuk aplikasi laman sesawang perkhidmatan terhadap masukan data *XPath* berniat jahat berbanding penyelidikan terdahulu dan *XTrust* juga menyediakan kawalan capaian yang lebih selamat ke dalam pangkalan data *XML* daripada kod *XPath* berniat jahat berbanding dengan penyelidikan yang terdahulu. Sebagai kesimpulan, penyelesaian menggunakan pendekatan dua tahap keselamatan meningkatkan tahap keselamatan di dalam laman sesawang perkhidmatan berteraskan pangkalan data *XML*.

ACKNOWLEDGEMENTS

Praise to The God, The Most Beneficent, and The Most Merciful, who has given me the opportunity to complete this thesis.

I would like to express my gratitude to my supervisor, Assoc. Prof. Dr. Lilly Suriani Affendey who has supported and guided with me throughout the research years. Also, I thank to Assoc. Prof. Dr. Nur Izura Udzir and Prof. Dr. Ramlan Mahmod for their contributions during my research work. As members of the committees, they have helped to improve the quality of this work with their constructive comments during the progress of the work.

I would also like to thank the Ministry of Higher Education Malaysia for the scholarship, Universiti Putra Malaysia and Faculty of Computer Science and Information Technology for the study leave and allowances they granted me during my studies.

Finally, heartiest thanks to all family members for their encouragement, support and prayers. Without them, it would hard to get through various stages of this life. Last but not least, very special thanks to my husband, Kamil Ikmal for his endless love, patience, sacrifices throughout these years.

This thesis was submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of Doctor of Philosophy. The members of the Supervisory Committee were as follows:

Lilly Suriani Affendey, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

Nur Izura Udzir, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

Ramlan Mahmod, PhD
Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

____
ROBIAH BINTI YUNUS, PhD
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date:

Declaration by graduate student

I hereby confirm that:

- this thesis is my original work;
- quotations, illustrations and citations have been duly referenced;
- this thesis has not been submitted previously or concurrently for any other degree at any other institutions;
- intellectual property from the thesis and copyright of thesis are fully-owned by Universiti Putra Malaysia, as according to the Universiti Putra Malaysia (Research) Rules 2012;
- written permission must be obtained from supervisor and the office of Deputy Vice-Chancellor (Research and Innovation) before thesis is published (in the form of written, printed or in electronic form) including books, journals, modules, proceedings, popular writings, seminar papers, manuscripts, posters, reports, lecture notes, learning modules or any other materials as stated in the Universiti Putra Malaysia (Research) Rules 2012 ;
- there is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity is upheld as according to the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) and the Universiti Putra Malaysia (Research) Rules 2012. The thesis has undergone plagiarism detection software.

Signature: ________________________ Date:______________________

Name and Matric No.: Aziah Asmawi, GS29708

Declaration by Members of Supervisory Committee

This is to confirm that:

- the research conducted and the writing of this thesis was under our supervision;
- supervision responsibilities as stated in the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) are adhered to.

Signature: ______________________________

Name of Chairman of Supervisory Committee: Lilly Suriani Affendey, PhD

Signature: ______________________________

Name of Member of Supervisory Committee: Nur Izura Udzir, PhD

Signature: ______________________________

Name of Member of Supervisory Committee: Ramlan Mahmod, PhD

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BTF	Bad Transaction Factor
BTFW	Bad Transaction Factor Weight
BTNum	Bad Transaction Number
DAC	Discretionary Access Control
DOM	Data Object Model
DTD	Document Type Definition
EF	Error Factor
EFW	Error Factor Weight
ENum	Error Number
ETV	Existing Trust Value
ETVW	Existing Trust Value Weight
MAC	Mandatory Access Control
SDL	Secure Development Lifecycle
SDLC	Software Development Life Cycle
RBAC	Role Based Access Control
RDBMS	Relational Database Management System
SAX	Simple API for XML
SGML	Standard Generalised Mark-up Language
SQL	Structured Query Language
TBAC	Trust Based Access Control
TV	Trust Value
XCure	XPath Secure Coding
XIPS	XPath Injection Prevention System
XLog	XML Log File for Security Rather Than Recovery
XML	Extensible Mark-Up Language
XPath	Xml Path Language
XQuery	Xml Query Language
XTrust	Severity-aware Trust-based Access Control
W3C	World Wide Web Consortium

CHAPTER 1

INTRODUCTION

1.1 Research Background

Modern web applications today are database-driven. Most web applications support features such as login, registration, online payment, money transfer and billing address. In order to access these features, the client must submit personal and confidential information such as one's name, username, bank account number, social security number, password, credit card number, and address, which are stored in the database of the application. Attacks on these kinds of application cost not only losing credentials, but also misuse of them.

Data is a very critical asset in any organizations. Over the last few decades, it has become an organization's most precious asset and everything an organization does involves using data in some way or other. It is important that partners and customers have access to the data. For that purpose, the data cannot simply be hidden behind a firewall because the partner and customer need to access the data for business transactions or data sharing as well. Therefore, employing secure mechanisms to protect data in a database system from being exposed to outsiders and misused by unauthorized users is vital.

Korth and Silberschatz (1997) stated that the Web is, in effect, a large distributed database, though with a query language and access mechanism quite different from those traditionally included in a database system. Almost all computer system applications have the database at its back-ends as the main information sources. This information is considered as an organization's most important asset, and thus needs to be protected from rival companies or malicious attacks. Web services that are connected to the Internet expose the databases that lie at the backend to attacks. One needs to secure the databases in order to protect the information stored in them.

In this section, the motivations behind this research are highlighted. The discussion starts by explaining the Web services' security issues which motivate the real need to prevent XPath injection attacks in Web service applications. Furthermore, the discussion on XML database security issues motivates the need to improve access control in database systems to prevent against insider misuse attacks in XML database stored-procedures. Overall, the motivation for this research is to provide secure mechanisms to increase the level of security in XML database-centric Web services.

1.1.1 XPath Injection Issues in Web Service Applications

A report by McKinsey indicates Web services as one of the most important trends in modern software development (McKinsey, 2008). However, the wide use and exposure of Web services results in any existing security vulnerability being most probably uncovered and exploited by hackers. In fact, XPath injection is one of the most frequent types of attacks in the web environment (Stuttard, 2007).
A recent report by Torrid Networks (2015) explains that their application security team encountered a few XPath injections during a recent engagement to audit an application for a large telecom provider, Telecom X-Factor. XPath injection added some interest to their assignment since the application was using XML to store data and used XPath to query the data (Torrid Networks, 2015).

These attacks take advantage of improperly coded applications to change queries sent to a database, enabling, for instance, access to critical data. If a website uses an XML (eXtensible Markup Language) document to store data and user input is included in an XPath query against that document, the user may be vulnerable to an XPath injection.
Vulnerabilities allowing XPath injection attacks are particularly relevant in Web services (Antunes et al. 2009), as their exposure is high and they frequently use a data persistence solution based either in a traditional relational database or in a XML database. Currently major database vendors and several open source efforts provide XML databases, and typically, access to these types of databases uses XPath expressions. Therefore, the goal of XPath injection is to maliciously explore any existing vulnerabilities in XPath expressions used by an application to access an XML database.

There are two versions of XPath queries: which are Version 1.0 (released in 1999) and Version 2.0 (released in 2010). The XPath query can be altered to achieve authentication bypass, business logic bypass and extraction of arbitrary data from the XML database (Siddharth, 2012). Upon noticing XPath injection vulnerability in an XPath-based application, an attacker does not need to fully understand the structure of the application. In fact, the attacker can generate a data query template that can be used for Blind XPath Injection within a few injection attack attempts (Klein, 2004).

Among the issues that arise from XPath injection, the most serious of these is the lack of secured protection in Web service applications against Blind XPath Injection attacks. Therefore, our research proposed a preventive mechanism that employed model-based validation to prevent Blind XPath input in Web service applications.

1.1.2 Insider Misuse Issues in XML Database

A large quantity of information is presented in XML format on the web for easy transportation. Due to the increased use of XML databases over the web, the need to secure these databases has increased. In a multiuser system, where the information is being shared across users who have different permissions, the need to implement a security model which gives controlled access to the authorized users is vital. XML access control was introduced to suit this purpose. XML access control is a security mechanism which restricts the access of the XML data to authorized users only. Many access control models and enforcement mechanisms have been proposed to prevent the unauthorized disclosure of XML data.

Different models for XML database access control have been proposed and developed. Access control systems for XML databases can be categorised into three core approaches: discretionary access control (DAC), mandatory access control (MAC) and role-based access control (RBAC) (Zhu et al., 2007). Most traditional access control models protect data from malicious activities of outside users but cannot protect the data from insiders (Chagarlamudi et al., 2009). Research has suggested that damage caused by insiders is more harmful than that of outsiders (Park et al., 2006).

Commonly, when there are reports on cybercrime against computer systems, the first thing that comes across the readers' mind is that a hacker or attacker is involved. It is less likely that they will think the crime involves employees or insiders in the organization. In reality however, employees or insiders often cause the most significant and costly security incidents. Research also suggested that damage caused by insiders is more harmful than that of outsiders (Richardson, 2011). Indeed, the fact that insiders are already within the organization often puts them in an ideal position to misuse a system if they intend to do so. The greater an individual's knowledge of an organization's internal resources, the greater the potential threat from that person. In fact, insiders are not interested in damaging systems or applications, but focus on obtaining critical information and accessing the internal level of resources for their personal advantage and gain.

Insider misuse can threaten personal data, national security, and economic prosperity. The 2008 CSI Computer Crime and Security Survey ranks insider abuse second only to viruses in terms of attack types experienced by respondents (Richardson, 2008). Furthermore, 87.1 percent of respondents said that 20 percent or less of their losses should be attributed to malicious insiders (Richardson, 2011). The damage is difficult to quantify, because it can extend far beyond the actual cost of the items stolen or corrupted.

In this research, we focus on mitigating malicious XPath code in XML database-stored procedures. Stored procedures are a part of a database that allows the programmer to set an extra abstraction level in the database. By using stored procedure a user can store its own function according to its needs (Wei, 2006). In stored procedures, a collection of XPath queries are included. As stored procedures could be coded by programmers, so too is this one of the causes of XPath injection attacks. XPath injection is one type of insider threat when it occurs in the stored procedure database level. This type of insider threat is a huge topic in data security and many methods have been proposed to identify and prevent misuse.

1.2 Problem Statement

Many Web services applications are used to distribute information from organizations to different users over a network. Most often, these applications that accept interactions from users, and perform some accesses to databases (DBs), are based on assumptions about legitimate input and legitimate code that are used to build XPath queries. These Web services applications are possibly vulnerable to XPath injection attacks (Vieira et al., 2009), which rely on some weak validation of the textual input that is somehow used to compose XPath queries. Attacker can maliciously crafted input, that contains XPath instructions or fragments of these, produces queries whose semantics is different from the one meant by the designers and may threaten the security policies of the underlying databases. Also, regardless of input validation, insiders may introduce malicious code in an application that, when triggered by some specific input, for example would violate designer's intention regarding security and accesses.

Researchers have started to contribute in the area of XPath injection and its possible liabilities (Blasco, 2007; Jinghua and Sven, 2008; Mitropoulos 2009; Antunes et al., 2009; Shanmughaneethi et al. 2011; Karumanchi and Aquicciarini 2015; Thome et al. 2015). Above all existing works in XPath injection; they focus on the solution for outsider threats without considering the insider threats caused by XPath injection.

The problem of insider threats is one of the most challenging to the organizations and research community since a long time. It is well proved that the damage done by insiders is more severe than that of external attackers (Shatnawi et al., 2011). Database at the backend of every Web services is vulnerable to insider threat (Farooqi & North, 2011). For that reason, researchers start to give attention to this area (Magklaras et al., 2006; Kandias et al., 2010; Farooqi & North, 2011; Greitzer et al., 2012; Brdiczka et al., 2012; Eldardiry et al., 2013; Legg et al., 2015; Hashem et al., 2015). Based on the previous works on insider threat, none concentrates on such threat in XML database stored procedure. Furthermore, above all existing works in insider threat, none of them consider on the XPath malicious code as their research issue.

Mitigating both insider and outsider are very important issue but yet, there are lack of researches provide the solution (Naidu, S., 2013). Only several researchers discussed and tackled both insider and outsider threats in their work (Nayak & Rao, 2014; Lindvall, J., & Rueda, D., 2014; Merlo et al., 2006). The problems of the existing works are none of them focus on the solution for XPath injections. Therefore, this research will consider both outsider and insider threats with respect to XPath injections. We come out with an effective way for securing the XML database from these threats by proposing two level security approaches solution. In this solution, we will combine the model based validation prevention system and severity-aware trust-based access control in order to improve the effectiveness and the efficiency of the propose approach compared to previous work by Shanmughaneethi et al., (2011) which use schema-based validation. The implementation of two level security approaches solution should overcome the XPath injections and hence, introducing the XML database-centric Web service for better security.

1.3 Research Aim

The aim of this research is to provide a secure mechanism for XML database-centric Web services. It must not only be capable of protecting against malicious XPath input in Web services application, but also manage to protect against malicious XPath code in XML database.

1.4 Research Objectives

In order to achieve the aim of this research, several research objectives have been identified as follows:

i) To propose two level security approaches solution which consist both model–based validation and severity-aware trust-based access control for XML database-centric Web services.

ii) To design system architecture for two levels security approaches solution which consist both model–based validation and severity-aware trust-based access control for XML database-centric Web services.

iii) To design system architecture for intrusion prevention system which employs model-based validation for Web services application.

iv) To design system architecture for access control which employs severity-aware trust-based access control for XML database stored-procedure.

1.5 Research Scope

This research was conducted within the research scope as described below:

i) This research focuses on security and performance.

ii) The two level security approaches solution would focus on protecting XML database-centric Web services from both outsider and insider threats with respect to XPath injections.

i) The intrusion prevention system would focus on preventing Web services from malicious XPath input.

ii) The severity-aware trust-based access control would focus on protecting XML database stored-procedures from malicious XPath code.

1.6 Research Significance

The purpose of this research is to provide a secure mechanism for XML database-centric Web services from outsider and insider threats with respect to XPath injections. The aim of the research can be achieved by providing two level security approaches solution for

XML database Web services using model-based validation and severity-aware trust-based access control against malicious XPath input and malicious XPath code.

1.7 Thesis Structure

This thesis presents the general issues in database security, the specific issues in both XPath injections upon database-centric Web services. This chapter has introduced the basic concepts in database security as a platform for the understanding of the research. The following is the outline of the thesis.

- **Chapter 2:** A detailed review regarding the study and work that relates to this research is given. The review covers topics related to the research issues focusing on XPath injections in Web services and XML database systems.
- **Chapter 3:** We discussed the research methodology employed in this research in this chapter. We elaborated each stages consists in this chapter, Research Problem Identification, Experimental Research Planning, Conducting the Experiment, Data Analysis, Evaluation and Discussion and Report Writing.
- **Chapter 4:** The implementation of the two level security approaches solution, XIPS, and XTrust is given in this chapter. This chapter presents the system architecture and description of each module.
- **Chapter 5:** Experimental results and discussion for each experiment are discussed in this chapter.
- **Chapter 6:** The discussion on research contributions and conclusion of this thesis are given in this chapter. Recommendations and suggestions for future improvements are covered.

REFERENCES

Antunes, N., Laranjeiro, N., Vieira, M., & Madeira, H. (2009). Effective detection of SQL/XPath Injection vulnerabilities in web services. *SCC 2009 - 2009 IEEE International Conference on Services Computing*, 260–267. http://doi.org/10.1109/SCC.2009.23

Apvrille A., and Pourzandi M., Secure Software Development by Example, *IEEE Security & Privacy*, 3 (4):10-17, July/August 2005.

Barrantes, E., Ackley, D., Forrest, S., Palmer, T., Stefanovic, D. & Zovi, D. (2003), Randomized instruc- tion set emulation to disrupt binary code injection attacks. *CCS 2003: Proceedings of the 10th ACM Conference on Computer and Communications Security*, pp. 281–289. Retrieved from http://citeseer.ist.psu.edu/barrantes03randomized.html

Benedikt, M., Fan, W., & Geerts, F. (2008). XPath satisfiability in the presence of DTDs. *Journal of the ACM (JACM)*, *55*(2), 8.

Bertino, E., Leggieri D., and Terzi E. (2004). Securing DBMS: Characterizing and Detecting Query Flood. *Proc. Ninth Information Security Conf. (ISC ’04)*.

Black Hat USA 2011. Retrieved from https://www.blackhat.com/html/bh-us-11/bh-us-11-home.html

Blasco, J. (2007). Introduction to XPath Injection techniques, 24–31.

Brdiczka, O., Liu, J., Price, B., Shen, J., Patil, A., Chow, R., & Ducheneaut, N. (2012). Proactive insider threat detection through graph learning and psychological context. In *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on* (pp. 142-149). IEEE.

Butts, J. W., Mills, R. F., & Baldwin, R. O. (2005). Developing an insider threat model using functional decomposition. In *Computer Network Security*(pp. 412-417). Springer Berlin Heidelberg.

Cannings, R., Dwivedi, H. & Lackey, Z. (2007) Hacking Exposed Web 2.0: Web 2.0 Security Secrets and Solutions (Hacking Exposed), McGraw-Hill Osborne Media.

Castano, S., Fugini, M.G., Martella, G. and Samarati, P. (1995). Database Security. Wokingham, England: Addison-Wesley Publishing Company.

Chagarlamudi M., Panda B. and Hu Y. (2009). Insider Threat in Database Systems: Preventing Malicious Users' Activities in Databases in *2009 Sixth International Conference on Information Technology: New Generations*, *ITNG '09*, 2009, pp. 1616-1620.

Chess B. and McGraw G., “Static Analysis for Security,” *IEEE Security and Privacy 2(6)*, 2004, pp. 76-79.

Chess, B., and West, J. (2007). Secure programming with static analysis, first ed. Addison-Wesley Professional.

Chi, H., Jones, E. L., & Brown, J. (2013). Teaching Secure Coding Practices to STEM Students. In *Proceedings of the 2013 on InfoSecCD'13: Information Security Curriculum Development Conference* (p. 42). ACM.

Chinchani, R., Iyer A., Hung Q. Ngo, Upadhyaya, S., (2005). Towards A Theory of Insider Threat Assessment, *Proceedings of the 2005 International Conference on Dependable Systems and Networks (DSN’05), IEEE*.

Chong,S., Liu,J., Myers,A., Qi,X., Vikram,K., Zheng,L., Zheng, X. (2007). Secure Web applications via automatic partitioning. In Proceedings of the 21st ACM SIGOPS Symposium on Operating Systems Principles.

Christensen, E, Curbera, F, Meredith, G and Weerawarana, S (2001). Web Services Description Language (WSDL) Version 1.1, WorldWideWeb Consortium (W3C).

Christensen A. S, Møller A., and Schwartzbach M. I. (2003). Precise analysis of string expressions. In Proc. 10th Intern. *Static Analysis Symposium (SAS 2003)*, pages 1–18.

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., & Weerawarana, S. (2002). Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. *IEEE Internet computing*, *6*(2), 86.

CVE. 2010. Common vulnerabilities and exposures. Retrieved from http://cve.mitre.org.

Dowd,M.,Mcdonald,J., and Schuh, J. (2007). The Art of Software Security Assessment. Addison-Wesley.

Eldardiry H., Bart E., Liu J., Hanley J., Price B., Brdiczka O.(2013). Multi-Domain Information Fusion for Insider Threat Detection. *SPW*, 2013, 2013 IEEE CS Security and Privacy Workshops (SPW2013) pp. 45-51, doi:10.1109/SPW.2013.14

Farooqi, N., & North, S. (2011). Trust-based access control for XML databases. *2011 International Conference for Internet Technology and Secured Transactions*, (December), 764–765.

Flawfinder. 2010. Retrieved from http://www.dwheeler.com/flawfind

Gartner S., Ruhroth T., Burger J., Schneider K., and Jurjens J. (2014). Maintaining requirements for long-living software systems by incorporating security knowledge, *Proceedings of the 2014 IEEE 22nd International Requirements Engineering Conference (RE2014)*, pp.103-112, IEEE CPS, 2014.

Gertz M., Jajodia S. (2007). Handbook of database security. Berlin: Springer-Verlag.

Grazie P., "Phd sqlprevent thesis. (2008). Ph.D. dissertation, University of British Columbia(UBC) Vancouver, Canada, 2008.

Greitzer, F. L., Kangas, L. J., Noonan, C. F., Dalton, A. C., & Hohimer, R. E. (2012). Identifying at-risk employees: Modeling psychosocial precursors of potential insider threats. In *System Science (HICSS), 2012 45th Hawaii International Conference on* (pp. 2392-2401). IEEE.

Guo P., A Scalable Mixed-Level Approach to Dynamic Analysis of C and C++ Programs. (2006). Master of Engineering thesis, Massachusetts Institute of Technology, USA, May 2006.

Hafiz M., Johnson R.E., and Afandi R. (2004). The security architecture of qmail," Proceedings of the 11th Conference on Patterns Language of Programming (PLoP'04), 2004.

Halfond W. and Orso A., "Preventing SQL injection attacks using AMNESIA," 28th Intl. Conf. on SW Engineering, 2006.Dec. 2007, pp. 57-64, doi:10.1109/SCIS.2007.357670.

Han-fa X., Bing-liang C. and Li-lin X. (2010). A mixed access control method based on trust and role," in *2010 Second IITA International Conference on Geoscience and Remote Sensing (IITA-GRS), 2010*, pp. 552-555.

Hao, D., Zhang, L., Zhang, L., Sun, J., and Mei, Vida H. (2009). Visual interactive debugging. In *Proceedings of the 31st International Conference on Software Engineering, ICSE '09, IEEE Computer Society (2009)*, 583–586.

Hashem Y., Takabi H., GhasemiGol M., and Dantu R.(2015). Towards insider threat detection using psychophysiological signals. In Proc. of the 7th ACM CCS International Workshop on Managing Insider Security Threats (MIST'15), Denver, Colorado, US, pages 71–74. ACM, October 2015.

Hazeyama, A., Saito, M., Yoshioka, N., Kumagai, A., Kobashi, T., Washizaki, H. & Okubo, T. (2015). Case Base for Secure Software Development Using Software Security Knowledge Base. In *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual* (Vol. 3, pp. 97-103). IEEE.

Hitchens, M. & Varadharajan, V. (2001). Rbac for Xml Document Stores. *LNCS*, 2229, 131-143.

Howard, M. & LeBlanc, D. (2003), Writing Secure Code, second edn, Microsoft Press, Redmond,WA.

Howard M., Lipner S. (2006). The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software", Microsoft Press.

Hui, F., Weinan, L., Wenchang, S., Zhaohui, L. & Bin, L. (2011). Trust-Oriented Access Control Based on Sources of Information Flow. *The 13th International Conference on Advanced Communication Technology (ICACT)*, 797-801.

Hunker, J., & Probst, C. W. (2008). Insiders and Insider Threats An Overview of Definitions and Mitigation Techniques, 4–27.

Jinghua Groppe,Sven Groppe. (2008). Filtering unsatisfiable XPath queries", *Journal Data & Knowledge Engineering , Vol.64 No. 1,Amsterdam*, 2008,pp.no 134-169.

Juillerat, N. (2007). Enforcing code security in database Web applications using libraries and object models. In *Proceedings of the Symposium on Library-Centric Software Design*. 31–41.

Kandias, M., Mylonas, A., Virvilis, N., Theoharidou, M., & Gritzalis, D. (2010). An insider threat prediction model. In *Trust, privacy and security in digital business* (pp. 26-37). Springer Berlin Heidelberg.

Karumanchi, S., & Squicciarini, A. (2015). A Large Scale Study of Web Service Vulnerabilities. *Journal of Internet Services and Information Security (JISIS)*, *5*(1), 53-69.

Kimelfeld, B. & Sagiv, Y. (2008), Revisiting redundancy and minimization in an xpath fragment, in 'EDBT '08: Proceedings of the 11th international conference on Extending database technology', ACM, New York, NY, USA, pp. 61–72.

Klein, A. (2004). Blind XPath Injection. (2005). Whitepaper from Watchfire, Director of Security and Research, Sanctum, pp. no 1–10.

Korth, H.F. and Silberschatz, A. (1997). Database Research Faces the Information Explosion. *Communications of the ACM*. 40(2): 139-142.

Kuang, C., Miao, Q. & Chen, H. (2006), Analysis of software vulnerability, in 'ISP'06: Proceedings of the 5th WSEAS International Conference on Information Security and Privacy', World Scientific and Engineering Academy and Society (WSEAS), Stevens Point,Wisconsin, USA, pp. 218–223.

Laranjeiro, N., Vieira, M., & Madeira, H. (2009). Protecting Database Centric Web Services against SQL / XPath Injection Attacks. *Engineering*, 271–278. http://doi.org/10.1007/978-3-642-03573-9_22

Legg P. A., Buckley O., Goldsmith M., and Creese S. (2015). Caught in the act of an insider attack: Detection and assessment of insider threat," in Proc. IEEE Int. Symp. HST, Waltham, MA, USA, 2015, in press.

Lin A., Vullings E. and Dalziel J. (2006). A Trust-based Access Control Model for Virtual Organizations, in *Fifth International Conference on Grid and Cooperative Computing Workshops, GCCW '06*, 2006, pp. 557-564.

Lindvall, J., & Rueda, D. (2014). The insider–outsider dilemma. *British Journal of Political Science*, *44*(02), 460-475.

Magklaras, G. B., Furnell, S. M., & Brooke, P. J. (2006). Towards an insider threat prediction specification language. *Information management & computer security*, *14*(4), 361-381.

Maybury M., Chase P., Cheikes B., Brackney D., Matzner S., Hetherington T., Wood B., Sibley C., Marin J., Longstaff T., Spitzner L, Haile J, Copeland J., and Lewandowski S., "Analysis and detection of malicious insiders," The MITRE Corporation, Tech. Rep., 2005.

McGraw. (2006).Software Security: Building Security In, Addison Wesley.

McKinsey&Company: Enterprise Software Customer Survey (2008).

Merlo, E., Letarte, D., & Antoniol, G. (2006). Insider and ousider threat-sensitive sql injection vulnerability analysis in php. In *2006 13th Working Conference on Reverse Engineering* (pp. 147-156). IEEE.

Mitropoulos, D. (2009). Fortifying Applications Against Xpath Injection Attacks, 1169–1179.

Naidu, S. (2013). Insider threat is as serious as outsider hacking. *The Business Times.*

Nayak, Umesha & Rao, Umesh, Hodeghatta. (2014). The InfoSec Handbook: An Introduction to Information Security. Apress.

Okun, V., Guthrie, W., Gaucher, R., and Black, P. (2007). Effect of static analysis tools on software security: Preliminary investigation. In Proceedings of the 3rd Workshop on Quality of Protection. 1–5.

OSVDB. 2010. Open source vulnerability database. Retrieved from http://osvdb.org.

OWASP CLASP. (2015). OWASP CLAPS Project. Retrieved from https://www.owasp.org/index.ph p/CLASP.

Park J. S. and Giordano J. (2006). Role-based profile analysis for scalable and accurate insider-anomaly detection," in *25th IEEE International Performance, Computing, and Communications Conference, IPCCC 2006*, 2006, pp. 463-470.

Pfleeger, C.P. (2000). Security in Computing. 2nd. ed. Upper Saddle River, N.J.: Prentice Hall, 2000.

President's Information Technology Advisory Committee (PITAC), Cyber Security: A Crisis of Prioritization, National Coordination Office for Information Technology Research and Development, Arlington. Retrieved from http://www.nitrd.gov/pitac/reports/20050301_cybersecurity/cybersecurity.pdf (2005).

Qi, N., Kudo, M., Myllymaki, J. & Pirahesh, H. 2005. A Function-Based Access Control Model for Xml Databases. The 14th ACM International Conference on Information and Knowledge Management, Bremen, Germany, 1099577: ACM, 115-122.

Richardson R. (2008). *2008 CSI Computer Crime and Security Survey.*

Richardson R. (2011). *2011 CSI Computer Crime and Security Survey.*

ROBBINS, T. (2000). Libformat. Retrieved from http://archives.neohapsis.com/archives/linux/lsap/2000-q3/0444.html

SEACORD, R. 2006. Secure coding in C and C++ of strings and integers. IEEE Secur. Priv. 4, 1, 74–76.

Shanmughaneethi, Ravichandran, & Swamynathan. (2011). PXpathV: Preventing XPath Injection Vulnerabilities in Web Applications. *International Journal on Web Service Computing*, *2*(3), 57–64. http://doi.org/10.5121/ijwsc.2011.2305

Siddarth S., Forbes T."Hacking XPath 2.0. (2012). *BlackHat Europe Conference, Amsterdam (2012).* Retrieved from https://media.blackhat.com/bh-eu-12/Siddharth/bh-eu-12-Siddharth-Xpath-WP.pdf

Singh S., "Trust Based Authorization Framework for Grid Services. (2011). *Journal of Emerging Trends in Computing and Information Sciences,* vol. 2, pp. 136-144, 2011.

Singhal, A. and Winograd, T. (2006). Guide To Secure Web Services (Draft), Sp 800-95. In Special Publication NIST.

Smith S. and Marchesini J. (2007). The Craft of System Security. Addison-Wesley Professional.

Stuttard, D., Pinto, M. (2007).The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws. Wiley, ISBN10: 0470170778.

Sun, L., Wang, H., Jururajin, R. & Sriprakash, S. (2010). A Purpose Based Access Control in Xml Databases System. The 4th International Conference on Network and System Security (NSS), 486-493.

SYMANTEC. (2008. Internet security threat report, trends for July-September 07. Volume XII.
Retrieved from http://eval.symantec.com/mktginfo/enterprise/white papers/b-whitepaper exec summary internet security threat report xiii 04-2008.en-us.pdf.

Taylor B. and Kaza S. (2011). Security injections: modules to help students remember, understand, and apply secure coding techniques. In Proceedings of the 16th annual joint conference on Innovation and technology in computer science education (ITiCSE '11). 3-7.

Theoharidou, M. and Gritzalis, D. (2007), “Common body of knowledge for information security”, IEEE Security & Privacy, Vol. 5 No. 2, pp. 64-7.

Thome J, Shar LK, & Briand L. (2015) Security slicing for auditing XML, XPath, and SQL injection vulnerabilities. IEEE 26th International Symposium on Software Reliability Engineering; 2015. p. 553-564.

Torrid Networks (2015). https://www.torridnetworks.com/case-studies/xpath-injection-telecom-x-factor-application-security-case-study

Tripathi, A., & Singh, U. K. (2013). Evaluation of severity index of vulnerability categories. *International Journal of Information and Computer Security*, *5*(4), 275-289.

Tsai, T. and Singh, N. (2002). Libsafe: Transparent system-wide protection against buffer overflow attacks. In *Proceedings of the International Conference on Dependable Systems and Networks*. 541.

Verma, B., Kumar, S. & Sharma, P. (2012). A Novel Approach for Multi-Tier Security for Xml Based Documents. IOSR Journal of Computer Engineering (IOSRJCE), Volume 5, 1-4.

Viega, J., Bloch, J., Kohno, T., MCGRAW, G. 2002. Token-Based scanning of source code for security problems. ACMTrans. Inf. Syst. Secur. 5, 3, 238–261.

Vieira, M., Antunes, N., & Madeira, H. (2009). Using Web Security Scanners to. *Word Journal Of The International Linguistic Association*, 566–571. http://doi.org/10.1109/DSN.2009.5270294

Wang, J. & Osborn, S. L. (2004). A Role-Based Approach to Access Control for Xml Databases. *The Ninth ACM Symposium on Access Control Models and Technologies, Yorktown Heights, New York, USA*, 990047: ACM, 70- 77.

Wassermann, G. & Su, Z. (2004), An analysis framework for security in web applications, in *SAVCBS 2004: Proceedings of the FSE Workshop on Specification and Verification of Component-Based Systems*, pp. 70–78.

Weber,M., Shah,V., Ren, C. (2001). A case study in detecting software security vulnerabilities using constraint optimization. In *Proceedings of the Workshop on Source Code Analysis and Manipulation*. 3–13.

Wei,K., Muthuprasanna,M., Kothari, S. (2006). Preserving SQL injection attacks in stored procedures. In Proceedings of the *Australian Software Engineering Conference*. 191–198.

Wurster, G. and van Oorschot, P.C. (2008), “The developer is the enemy”, Proceedings of the *2008 Workshop on New Security Paradigms (NSPW’08), Lake Tahoe, CA, 22-25 September, ACM Press, New York, NY*, pp. 89-97.

Xie, J., Lipford, H. R., and Chu, B. (2011).Why do programmers make security errors? In *Proceedings of 2011 IEEE Symposium on Visual Languages and Human Centric Computing (2011),* 161–164.

Zhu, H., Jin, R. & Lu, K. (2007). A Flexible Mandatory Access Control Policy for Xml Databases. The 2nd International Conference on Scalable Information Systems, Suzhou, China, 1366890: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 1-4.

Zhao, L., Liu, S., Li, J. & Xu, H. (2010). A Dynamic Access Control Model Based on Trust. International Conference on Environmental Science and Information Application Technology (ESIAT), 548-551.