PalArch's Journal of Archaeology of Egypt / Egyptology

XTRUST: A SEVERITY-AWARE TRUST-BASED ACCESS CONTROL FOR ENHANCING SECURITY LEVEL OF XML DATABASE FROM INSIDER THREATS

Aziah Asmawi[1], Lilly Suriani Affendey[2], Nur Izura Udzir[3], Ramlan Mahmod[4]

Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia. (a_aziah, lilly, izura, ramlan }@upm.edu.my

Aziah Asmawi1, Lilly Suriani Affendey2, Nur Izura Udzir3, Ramlan Mahmod4. Xtrust: A Severity-Aware Trust-Based Access Control For Enhancing Security Level Of Xml Database From Insider Threats--Palarch's Journal Of Archaeology Of Egypt/Egyptology 18 (3), 444-450. ISSN 1567-214x

Keywords: XML database, trust-based access control, severity-aware, trust values

ABSTRACT:
The topic of security in XML databases is important as it includes protecting sensitive data and providing a secure environment to users. To improve security and provide dynamic access control for XML databases, we developed severity-aware trust-based access control for XML databases. Severity aware trust-based access control for XML databases manages the access policy depending on users' trustworthiness (trust values) and prevents unauthorized processes, malicious transactions, and insider threats. Privileges are automatically modified and adjusted over time depending on user behaviour and query severity. In this paper, a severity-aware trust-based access control module for XML databases is evaluated in terms of security perspectives. The experimental results illustrate the effect of the severity factor on the calculation of Trust values compared to the existing work.

Keyword: XML database, trust-based access control, severity-aware, trust values.

INTRODUCTION

XML (Extensible Markup Language) is widely used in many applications as it can store, exchange, and transfer data. Much of the research on XML focuses on storage strategies and query performance. Although data storage and retrieval techniques are important, so is security and in comparison, this is a neglected area. XML databases are multi-user systems, meaning they can be accessed by millions of users and can provide a huge amount of data. Much of this data is sensitive and personal. Confidential data need to be protected and saved in a secure environment. Security research for XML

databases is crucial in protecting data from unauthorised processes and misuse.

Different models for XML database access control have been proposed and developed. Access control systems for XML databases can be categorized into three core approaches which are discretionary access control (DAC), mandatory access control (MAC), and role-based access control (RBAC) [1] [2] [3]. Most traditional access control models protect data from malicious activities of outside users but cannot protect the data from insiders [4]. Research has suggested that damage caused by insiders is more harmful than that of outsiders [5]. The insider threat is a huge topic in data security and many methods have been proposed to identify misuse behaviour, yet there has been no work on dynamic updates to access privileges concerning trust for XML databases. Trust-based access control has become an established technique in many areas, such as networks and virtual organisations. It depends on a trust management system, which automatically calculates and updates the trust values of users. Trust values rely on users' behaviours, users' histories, users' credit, and users' operations. Users can access resources through trust values and levels [6] [7] [8] [9] [10].

In this paper, we propose the implementation of a severity-aware trust-based access control approach for XML databases. The system consists of two modules: a Trust Module and an Access Control Module. The key idea in this work is to test the effect of the severity factor on the calculation of trust value to enhance the security of the XML database. We improve and extend the trust value calculation to four equations; each of which is used for specific cases.

XTrust Architecture

In this section, a practical trust-based access control module for XML databases is described. This system is dynamic and responsive to users' history of errors, bad transactions, and also queries severity.

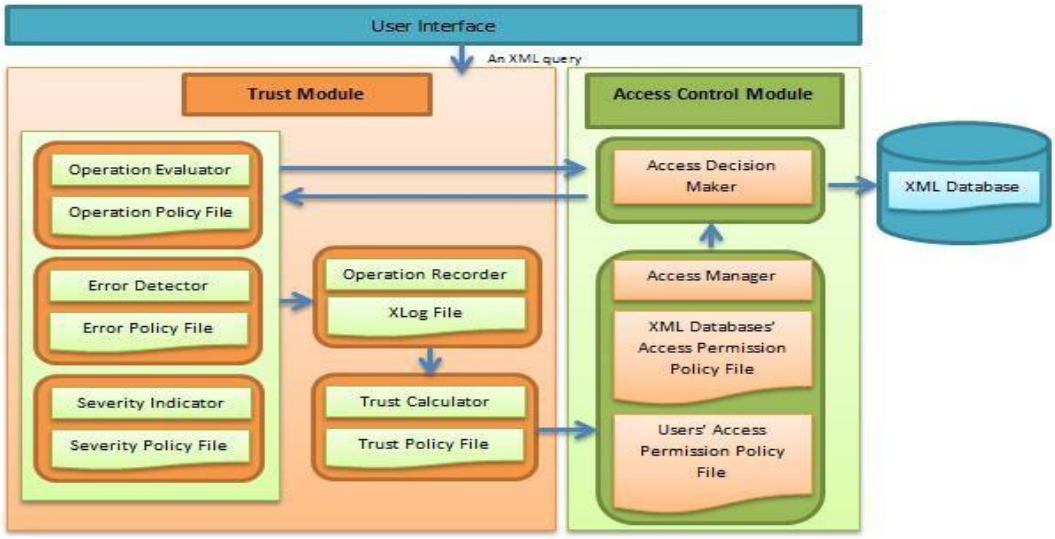

Figure 1: XTrust Architecture

The module consists of two main parts: the Trust Module and the Access Control Module. The Trust Module is responsible for recording errors, bad transactions, and query severity, evaluating them, and calculating the new trust value. The Access Control Module is responsible for the access permission policy and access decisions. System architecture for XTrust is depicted in Figure 1.

Based on the figure, there are two main modules in XTrust. The first module is Trust Module and second module is Access Control Module. The Trust Module is the main part of the Trust-based access control system for XML databases. It receives XML queries from users through the user interface, evaluates their queries, and calculates their trust values. The evaluation process depends on the users' existing trust values, new bad transactions, new errors, and query severity. After calculating the new Trust Value for the user, it will send the Trust Value to the Access Control Module to update the user's privileges. This module consists of many parts: the Operation Evaluator, the Error Detector, the Severity Indicator, the Operation Recorder, and the Trust Calculator. Each part has its functions and works in light of the related policy rules. All these parts are connected to achieve the main goal of calculating trust values for users.

The Trust Module

The Trust Module is constructed of many parts that work together to achieve the main goal of calculating users' trust values. These parts consist of an Operation Recorder, an Error Detector, a Severity Indicator, an Operation Evaluator, and a Trust Calculator. The Operation Recorder records errors, bad transactions, and queries' severity in the Xlog file. The Xlog file is designed to be dynamic and to be stored temporarily for a set period to reduce storage and improve searching performance. Error Detector, Operation Evaluator, and Severity Indicator work in the light of the error policy file, operation policy file, and severity policy file. Each of these policy files has the role of defining what an error or a bad transaction is and also what is the severity of each query. The trust calculator uses the data recorded by the Error Detector, the Operation Evaluator, and Severity Indicator. The main goal of the trust calculator is to compute a new trust value. The trust value depends on the user history of malicious bad transactions, error, and queries' severity factors.

Access Control Module

The Access Control Module consists of the access manager and the access decision-maker. The access manager deals with access permission policies that primarily depend on trust value. These policies are

Table 1: Trust value equations

Where EF>0 and BTF>0 and SF>0 Then	TV=ETV*ETVW-EF*EFW-BTF*BTFW-SF*SFW	(1)
Where EF>0 and BTF = 0 and SF>0 Then	TV=ETV*ETVW-EF*EFW-SF*SFW	(2)
Where EF—0 and BTF>0 and SF>0 Then	TV—ETV*ETVW-BTF*BTFW-SF*SFW	(3)

Where EF=0 and BTF=0 and SF>0 TV=ETV*ETVW-SF*SFW (4)
Then

divided into subject policy by assigning TV to the subject and object policy, which concentrates on giving each item of data the appropriate TV. The access decision-maker handles the XML query and then either permits or denies the request. The final decision depends directly on defined access permission policies in the access manager. The TV of the user is compared with the TV of the required XML data. If the trust value for users equals or is larger than the trust value of data, then the user is allowed to access the data; otherwise, access is denied.

Evaluating the Trust Module

Trust value is changed depending on the existing trust value, error factor, bad transaction factor, severity factor, and their weights. In general, this experiment shows how trust value is affected by the error, bad transaction, and severity factors.

Calculating Trust Values

A new Trust Value (TV) is generated using four values: Existing Trust Value (ETV), Bad Transaction Factor (BTF), Error Factor (EF), and Severity Factor (SF). Each factor is multiplied by a weight that reflects the importance of the factor in the system and shows to what extent the factor affects the final TV. Each weight is a percentage that shows how much the factor will affect the general equation and the new TV. The weights are (ETVW), Bad Transaction Factor Weight (BTFW), Error Factor Weight (EFW), and Severity Factor Weight (SFW).

EFW, BTFW, and SFW range is between 1 % and 20%. The ETVW range is between 80% and 99%. Range values are selected to keep the TV within suitable bounds. The maximum for error, bad transaction, and severity weights is 20% and not higher because the system aims to adjust user privilege according to behaviour and not to block user access completely. For example, if this weight was high, such as 60%, then the TV would drop suddenly and dramatically and may cause other access problems.

ETVW is regarded as the basic value to calculate the new TV. The new TV is derived from the previous existing one and this explains why this weight should be in the range between 80% and 99%. The TV increases when there are no bad transactions, errors, or severity but it drops markedly when the BTF, EF, SF, or all increase. There are eight different equations to calculate the Trust Value and each one applies to specific cases. Trust Value equations are as depicted in Table 1.

Based on Table 1, equation (4) is used to calculate TV when there are no errors and bad transactions but there is a severity factor and this increases the TV slightly. If there are errors or bad transactions, equations (2) and (3) are used to calculate TV. The EF and SF are subtracted from the ETV in (2) when there

is no bad transaction. The same principle applies to (3) when there is no error factor; the BTF and SF are subtracted from ETV. In general, if there are errors, bad transactions, and severity, the TV should decrease in all cases. This case scenario applies in equation (1). It subtracts the EF, BTF, and SF from ETV to find the new TV. Table 2 illustrates the calculation of TV for some general cases when the ETV is 0.5 without severity factor (Farooqi, 2012) while this work has considered on severity factor. The calculation of TV with severity factor is depicted in Table 3.

From the graph shown in Figure 2, it can be concluded that when there is no error, bad operation, and query severity, the trust value for each approach increases but it will decrease if there is an error, bad transaction, or query severity. Trust value with severity factor is lower than trust value without severity factor which indicates that when there is severity factor, the security in XML database is improved. The trust value is updated automatically in the users' access permission policy file. The XML database's access permission policy file describes the trust values required to access XML nodes which have their trust value. The access manager sub-model matches two files to manage a user's right to access requested data. The access decision maker allows or denies users access to XML databases according to the results. As an example, user have TV = 0.200 when applying (2) with EV=0.5, ETVW=0.8, EF=1, EFW=0.05, BTF=1, BTFW=0.1, SF=1 and SFW=0.05. This means that this user can only access XML database nodes that have TV 0.200 or less.

Table 2: The calculation of Trust Value (TV) without severity factor

ETV	ETVW	EF	EFW	BTF	BTFW	TV
0.5	80%	0	5%	0	10%	0.550
0.5	80%	0.25	5%	0.25	10%	0.362
0.5	80%	0.5	5%	0.5	10%	0.325
0.5	80%	0.75	5%	0.75	10%	0.287
0.5	80%	1	5%	1	10%	0.250

(Farooqi,2012)

Table 3: The calculation of Trust Value (TV) with severity factor

ETV	ETVW	EF	EFW	BTF	BTFW	SF	SFW	TV
0.5	80%	0	5%	0	10%	0	5%	0.600
0.5	80%	0.25	5%	0.25	10%	0.25	5%	0.350
0.5	80%	0.5	5%	0.5	10%	0.5	5%	0.300
0.5	80%	0.75	5%	0.75	10%	0.75	5%	0.250
0.5	80%	1	5%	1	10%	1	5%	0.200

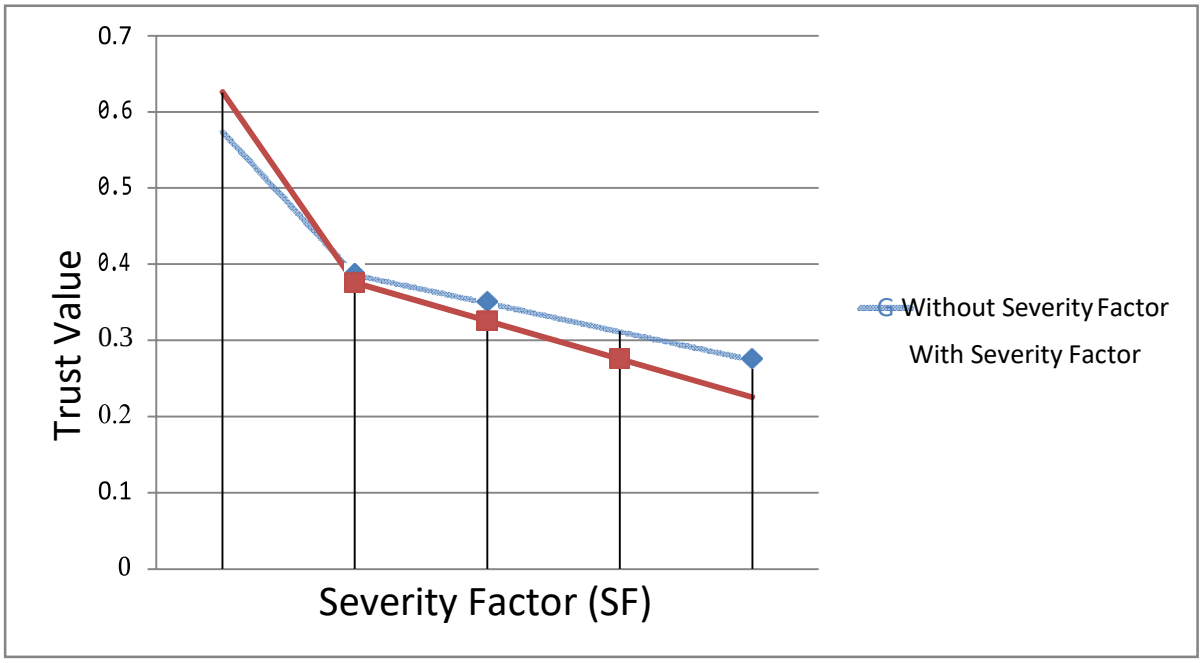

Figure 2: Comparison of trust value with and without severity factor

CONCLUSION

In this paper, we have evaluated our trust-based access control module for XML databases in terms of its security performance. From the experiment conducted, it shows that by adding a severity factor in trust value calculation, it improves the security environment for accessing XML databases by detecting misuse from insiders. This approach combines detecting insider threats and improving access control by using severity-aware trust-based access control. The access decision depends on matching the node trust value and the user trust value. Since the established technique is role-based access control, the model can be initiated using role-based trust values but then trust can grow or decrease according to user behaviour and query severity.

REFERENCES

M. Hitchens and V. Varadharajan, "RBAC for XML Document Stores," in Information and Communications Security, Lecture Notes in Computer Science, vol. 2229, S. Qing, T. Okamoto and J Zhou, Eds. Springer Berlin/Heidelberg, 2001, pp. 131-143.

J. Wang and S. L. Osborn, "A role-based approach to access control for XML databases," in the Proceedings of the Ninth ACM Symposium on Access Control Models and Technologies, Yorktown Heights, New York, USA, 2004, pp. 70- 77.

H. Zhu, R. Jin and K. Lu, "A flexible mandatory access control policy for XML databases," in the Proceedings of the Second International Conference on Scalable Information Systems, Suzhou, China, 2007.

M. Chagarlamudi, B. Panda and Y. Hu, "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.

J. S. Park and J. Giordano, "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.

A. Lin, E. Vullings and J. Dalziel, "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.

F. Almenarez, A. Marin, D. Diaz and J. Sanchez, "Developing a model for trust management in pervasive devices," in *Pervasive Computing and Communications Workshops,* 2006. *PerCom Workshops 2006. Fourth Annual IEEE International Conference on,* 2006, pp. 267-271.

X. Ma, Z. Feng, C. Xu and J. Wang, "A Trust- Based Access Control with Feedback," in *International Symposiums in Information Processing (ISIP),* 2008, pp. 510-514.

X. Han-fa, C. Bing-liang and X. Li-lin, "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.

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