2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing

Access Control Algorithm on File View in Intranets

Yong Ai
Computer School
Wuhan University
Wuhan, China
aiywhu@gmail.com

Hongbin Dong
State Key Lab of Software Engineering
Wuhan University
Wuhan, China
hbdong@whu.edu.cn

Xing Wu
Computer School
Wuhan University
Wuhan, China
wuxing.vip@gmail.com

Yiwen Liang
Computer School
Wuhan University
Wuhan, China
ywliang@whu.edu.cn

***Abstract*—Nowadays, a new security problem is arising in intranets. The threats from inside an organization account for a rapidly increasing proportion of losses. A new concept of "File View" is proposed to resolve this security problem in intranets, which uses the structure of database view for reference. Because of the differences between file system and database, there are some challenges in extending this proposal to file systems. This paper proposes an algorithm to protect confidential information in file system from being accessed by illegal users. First, the paper proposes an algorithm as "File View Access Control Algorithm"(FVACA) which revises the algorithms proposed in previous work. Then, to verify the feasible of this algorithm, this paper suggests to realize it on "Microsoft Office Word" which is one of the most popular digital formats used in intranets. For protecting the confidential information in file view, a method based on the compound format of "MS Word" is adopted. Finally, the experiment based on "MS Word" shows that FVACA is effective on protecting content from being accessed by illegal users.**

***Keywords*-File View; Access Control; Information Security; Information Hidden**

I. Introduction

Today, attacks coming from inside an organization are becoming an important issue. It is important to strengthen the security of intranets of enterprises and organizations [1]. At present, the security of file system is increasingly important in the intranet. Most intranets adopt some form of access control to protect digital files [2]. The access control model used in intranets results in the security problem. The operating systems used in most intranets belong to class (C2) [3] of the TCSEC (Trusted Computer System Evaluation Criteria) – Unix/Linux and variants of Microsoft Windows. In this class, the access control model is Discretionary Access Control (DAC) [4, 5]. There is a security problem in DAC model, in that it is unable to enforce information flow controls [6].

At present, there are two methods to solve this problem: a) use EDMS (Enterprise Document Management System) to manage digital documents in intranets, b) change the operating system to a more secure one. But these two methods do not accord well with the environment in intranets [1]. To satisfy the security requirements of an intranet, a solution is needed which a) uses DAC as access control model; b) uses file systems rather than databases [1]. The concept of "File View" proposed in [7] can satisfy the requirements mentioned above. The "File View" approach is intended to improve the DAC model when either used alone, or used in conjunction with the MAC model, to provide security control within a single security level [1].

In the rest of this paper, we propose an algorithm as "File View Access Control Algorithm (FVACA)" which revises the algorithms in previous work. Then, this paper realizes it base on the "Microsoft Office Word" format. Finally, the result of experiment on "MS Word" shows that FVACA is effective on protecting content from being accessed by illegal users.

II. Related Work

The concept of an enterprise document management system was proposed in 1991 [8]. The stored objects in an EDMS are organized, accessed and manipulated through a database management system. An EDMS is designed to securely manage the production and delivery of high-value documents[9]. But all documents must be exchanged through the database. This is inconvenient to users in an intranet in some situation.

The traditional access control models most commonly used are DAC[5, 10], MAC[11] and RBAC(Role-Based Access Control Model)[12]. Today, most operating systems use DAC as the access control model. The DAC model, is unable to enforce information flow controls[6]. This security problem can not be avoided in intranets.

The concept of "File View" based on no-structured file is proposed in 2009 [1, 7]. But there are some studies using an XML(*eXtensible Markup Language*) [13] view which is semi-structured. The work on querying and updating on XML views [14–18] forms a useful background for file views. But in a file system, there is no access language corresponding to XPath (XML Path Language) [19] and XUpdate (XML Update Language) [20]. The files in a file system have no common structure. It would be infeasible to transform all unstructured files to XML file before transferring them. So a similar mechanism for unstructured files to that for semi-structured files is needed.

III. Access Control on File View

A. Framework of File View

The concept of "File View" uses the structure of database view for reference. But there are some differences between database views and file views from the storage format. A database view is structured while a file view in file system is unstructured [1]. The framework for file views in previous work can be described as Fig. 1 [1].

978-0-7695-4011-5/10 $26.00 © 2010 IEEE
DOI 10.1109/NSWCTC.2010.175

IEEE computer society

Authorized licensed use limited to: SRM Institute of Science and Technology. Downloaded on August 04,2023 at 07:03:36 UTC from IEEE Xplore. Restrictions apply.

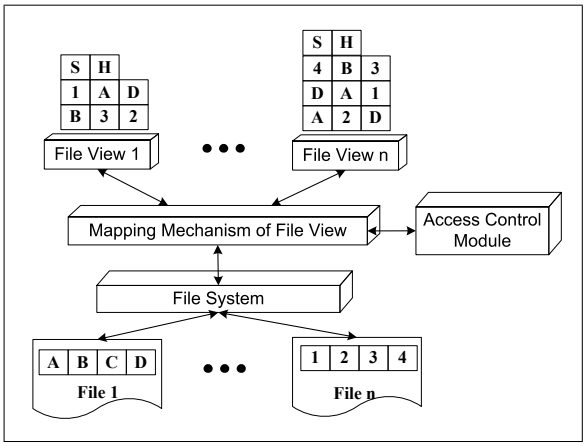

Fig. 1. The File View Framework

As seen in Fig. 1, the logical files in file system are divided into logical blocks. This structure makes the access modes for logical files in a file system more flexible [1].

B. File View Access Control Algorithm

The access control algorithms are presented in [1]. With these algorithms, confidential information would not be accessed by illegal users. But there are three shortcomings in the algorithms in it.

1) The grant access is not considered in it. When user wants to grant access authorizations to others, the grant authorization should be verified first. Then, the new authorizations should be updated in the file view.
2) When users modify the content of the logical blocks of a file view, the content is modified also in write algorithm. But if a user changes the whole content of a logical block, the meaning of this logical block is totally changed. The owner of this logical block should be changed as the reviser, but not the original owner. This would be more reasonable.
3) When users modify a logical block, the logical file in intranet which is cited by this logical block should be changed also in write algorithm. It would result in some mistake. The original owner's meaning may be changed by others without his/her agreement. The modify on the logical block should just change the content in file view.

To revise the shortcomings mentioned above, a file view access control algorithm could be described as algorithm 1. Considering the percentage of modification while writing file view, there is a new threshold value M defined in write process. After a user modify the content of a logical block, the owner of the content of this logical block should be redefined. If he/she changes the meaning of the content totally (greater than M), he/she should be treated as the new owner of this logical block. But if he/she just changes the meaning a little (less than M), the owner of this logical block should not be changed.

Algorithm 1 File View Access Control Algorithm

```
if has authorizations on file view FV then
  if Read Operation then
    for each logical block LB_i do
      if has read authorization on LB_i then
        show content of LB_i
      end if
    end for

  else if Write Operation then
    for each logical block LB_i do
      if has write authorization on LB_i then
        if delete all content of LB_i then
          delete LB_i from FV
        else
          p ⇐ updated percentage
          if p ⩾ M then
            create a new file F
            F ⇐ new content
            add a new logical block cited F
          else
            update content of LB_i to FV
          end if
        end if
      end if
    end for
    if exist new content besides all LBs then
      create a new file F
      F ⇐ new content
      add a new logical block cited F
    end if

  else if Grant Operation then
    for each logical block LB_i do
      if has grant authorization on LB_i then
        update ACL of LB_i
      end if
    end for
  end if
end if
update access history of FV
```

IV. File View on "Microsoft Office Word"

"Microsoft Office Word" is one of the most popular digital format used in intranet. It is one of the best choice to verify the feasibility of this framework and algorithm. So this paper adopts it to realize the algorithm.

A security problem is needed to be resolved at first. Files in file system is text format, which means all the information in this file is uncovered to users. So the security attribute and access history is the confidential information in a file view should be hidden effectively and could not be accessed through illegal manner. The technology of information hidden in text file should be adopted while realizing this algorithm.

Authorized licensed use limited to: SRM Institute of Science and Technology. Downloaded on August 04,2023 at 07:03:36 UTC from IEEE Xplore. Restrictions apply.

A. Information Hidden in Text File

At present, the most popular media used for information hiding are image or video. It is because that these media include much redundant information, with the characteristic of vision and hearing, these media can take information which is not sensed by people. But it is difficult to hide information in text file, because there is almost none redundant information in it. It is easily discovered by people because of the obvious change [21].

The methods of information hiding in text document which are in common use can be separated as two classes: one belongs to natural language data hiding while the other is format-based [22]. The first one means not to change the meaning of the information, but it changes the content of the information. This would influence the accuracy in some condition. The format–based method exploits end–of–line spaces, inter–sentence spacing, or inter–word spacing. This tiny change is not obvious, but there are two shortcomings as below.

1) When users modify or delete some sentences or paragraphs which contain invisible information, the hiding information in them would be deleted.
2) The format–based method could hide few information in text document. If the size of carrier is too small, the information could not hide in it. It is not convenient to users to choose carrier.

B. Information Hidden in "Microsoft Office Word"

'MS Word" is compound format file, which named "Microsoft Compound Document File Format" (MCDFF) [23]. The information in this format is stored as streams, the typesetting tools could only read specific streams defined beforehand. So users could insert self-defined streams which is used to store confidential information. These streams could not be identified by the typesetting tools (MS Word). A compound format document with a user-defined stream named as "UserDefined" is showed as Fig. 2.

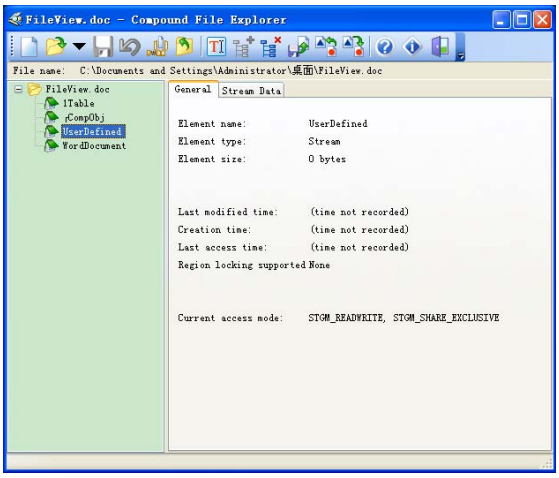

Fig. 2. An example of compound file with user-defined stream

The characteristic of this format could be used for information hiding. This method has three advantages.

1) When users open the file view with hidden information, the typesetting tool (MS Word) could only recognize the specific streams. It could not access the user-define streams. Users could not modify or delete the hidden information while they modifying sentences or paragraphs.
2) The size of the stream is not limited in typesetting tools. So users could hide any information they want in streams.
3) If the access control module is not installed in a operating system, "MS Word" would not access the user-defined streams. Users would not feel there exist any difference between file with or without user-define streams. This is convenient to users to exchange information.

V. Experiment on "MS Word"

The purpose of this experiment is to verify the feasible of the framework and the algorithm in section III. It is important to verify these two hypotheses.

1) The authorizations can be granted on the logical blocks but not on the document level. Different users could access different parts of the same document as appropriate.
2) It should prevent information from leaking out unintentionally while being transferred in an intranet. Even the file view is duplicated unintentionally, users could just access the content they could access before with the access control module. If they access it without the access control module, they could view nothing from the typesetting tools.

A. Technique Foundation

To prevent the content from being accessed by unauthorized users, the content of logical blocks in file view in "MS Word" should not be stored as usual. In this paper, the content of logical blocks is stored in user-defined streams named as "LogicalBlockNN"('NN' means ID of logical blocks). The access authorizations are stored in a stream named "SecurityAttribute". These user-defined streams would not be accessed by the typesetting tools.

When users try to access the file view with user-defined streams, their authorizatiosn would be verified. They could only get the content stored in the "LogicalBlockNN" streams which they has authorizations. First, the access control module intercept the read access. Then, the user's read authorization is compared with the information in "SecurityAttribute" stream. Finally, the logical block streams which are allowed to this user is restored to original content in the typesetting tools. The user could view the content which he/she has authorizations.

If a file view is viewed in a file system without the access control module in this framework in Fig. 1, the typesetting tool could only access the content stored in stream "WordDocument". Even if the user has the authorizations of the logical blocks, the content in the logical blocks could not be accessed in the file system without access control module. This function could protect the confidential information effectively.

Authorized licensed use limited to: SRM Institute of Science and Technology. Downloaded on August 04,2023 at 07:03:36 UTC from IEEE Xplore. Restrictions apply.

B. Preparing Work

The prepare work of this experiment contains four steps as below.

1) User Adam creates a "MS Word" format file as "WordView.doc". The original content of this document is showed as Fig. 3.

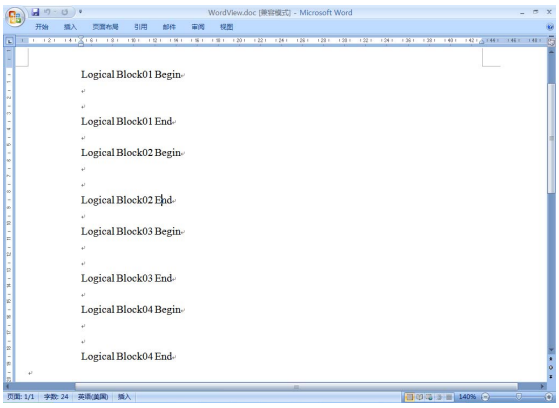

Fig. 3. The original content of "WordView.doc"

2) Adam creates four user-defined streams named "LogicalBlock01" – "LogicalBlock04", which are used to stored the content of logical blocks. Then Adam creates another user-defined stream named "SecurityAttribute", which is used to stored the authorizations. The structure of "WordView.doc" with four logical blocks is described as Fig. 4.

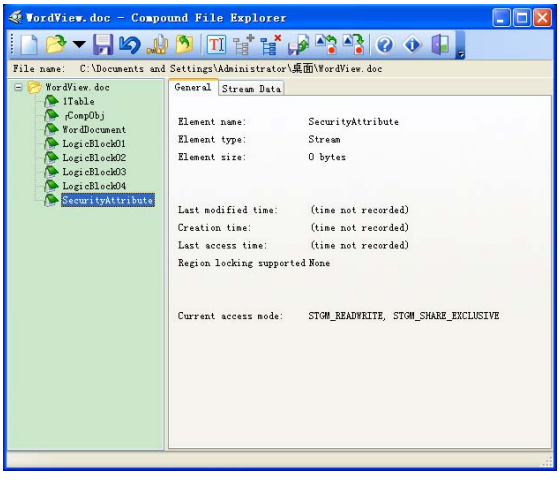

Fig. 4. The structure of file view "WordView.doc"

3) Adam divides the content of "WordView.doc" into four parts as the content describes. Then these parts are stored in "LogicalBlock01" – "LogicalBlock04" streams respectively. At last, he clear the content of "WordView.doc" in typesetting tool so that others could not view the content if they do not has access control module.

4) Adam grants the logical blocks to other users in intranet as Table I. Then these authorizations are stored in the streams named "SecurityAttribute".

TABLE I
THE USERS' AUTHORIZATIONS ON FILES

User ID	LB01	LB02	LB03	LB04
Adam	RW	RW	RW	RW
Bob	R	R	–	RW
Lucy	–	–	–	–
Tim	R	–	RW	R
Campbell	–	–	R	R

C. Viewing Flexibility

When users try to access the document "WordView.doc", they would get different content because of the different authorizations they owned. The sketch map of the result about users access file view can be shown as Fig. 5.

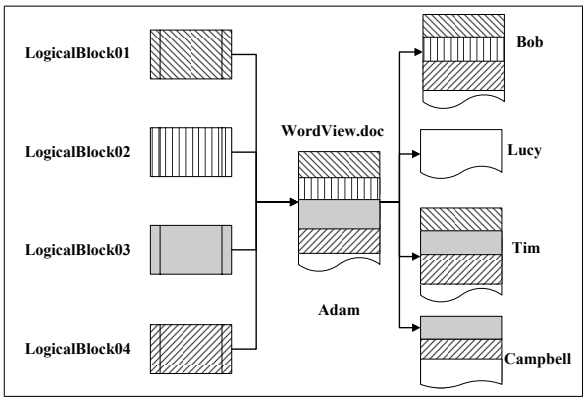

Fig. 5. Sketch map of the access result

Through this experiment, the function about "Viewing Flexibility" could be verified in file view framework. Different users could access different parts from one document because of their different authorizations.

D. Leaking out Unintentionally

When Bob gets the document "WordView.doc" from Adam, he copies it as "WordView_1.doc". And then, he regrant the access authorizations of the duplication "WordView_1.doc" to Campbell. Fig. 6 shows the content which Campbell reads from the two documents. Left part is the content he reads from "wordview.doc" while the right part is "WordView_1.doc".

Through this experiment, this framework and algorithm could effectively prevent information from leaking out unintentionally while being transferred in intranets.

VI. Conclusion and Future Work

In this paper, we propose an algorithm as "File View Access Control Algorithm (FVACA)" which revises the algorithms proposed in previous work. Then, to verify the feasible of this algorithm, this paper realizes it base on the "Microsoft Office Word" format which is one of the most popular used in intranets. For protecting the confidential information in file view, we adopt a method based on the stream in compound

Authorized licensed use limited to: SRM Institute of Science and Technology. Downloaded on August 04,2023 at 07:03:36 UTC from IEEE Xplore. Restrictions apply.

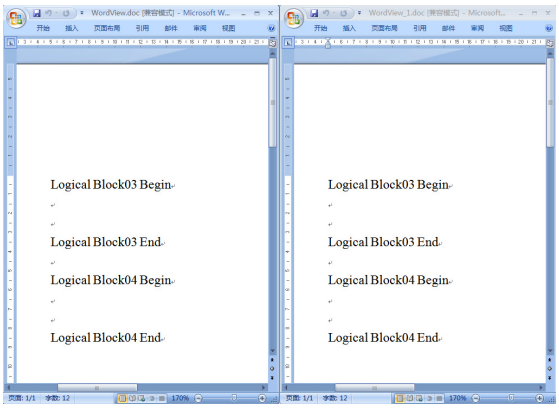

Fig. 6. Comparing result of duplication

format. Finally, the experiment based on "MS Word" shows that FVACA is effective on protecting content from being accessed by illegal users.

For the future, we are working on extending the file view mechanism defined in this paper in the following directions:

1) The information about access history should be included in this experiment, the function "Access Tracing" should be verified.
2) The information about security attribute and access history should not be accessed or modified illegally. A secure function should be adopted to protect the information which stored in streams in this experiment.

ACKNOWLEDGMENT

This work was supported by Research Grant No.60573038 and No.90204011 from National Natural Science Foundation of China. This work was supported by the Research Project No.A1420080183 from Ministry of Education of People's Republic of China. This paper was supported by "Self–research program for Doctoral Candidates (including Mphil–PhD) of Wuhan University in 2008.

REFERENCES

[1] Y. Ai, H. Dong, Y. Liang, and R. Mckay, "A Framework for a File View Model in Intranets," in *2009 International Conference on Computer Sciences and Convergence Information Technology*, 2009, pp. 198–201.

[2] H. Zhang, J. Diao, and Q. Wen, "Secure Files Management System in Intranet," in *Internet Computing in Science and Engineering, 2008. ICICSE'08. International Conference on*, 2008, pp. 306–311.

[3] D. Latham, "Department of Defense Trusted Computer System Evaluation Criteria," *Department of Defense*, 1985.

[4] B. Lampson, "Protection," *ACM SIGOPS Operating Systems Review*, vol. 8, no. 1, pp. 18–24, 1974.

[5] R. Sandhu and P. Samarati, "Access control: principle and practice," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 40–48, 1994.

[6] R. Sandhu and Q. Munawer, "How to do discretionary access control using roles," in *Proceedings of the third ACM workshop on Role-based access control*. ACM New York, NY, USA, 1998, pp. 47–54.

[7] Y. Liang, Y. Ai, H. Dong, and T. Li, "File View: Secure Model in Intranet," in *2009 International Conference on Networks and Digital Society*, 2009, pp. 198–201.

[8] R. SMITH and M. T MENDELSSOHN, "DOCUMENT MANAGEMENT AND PRODUCTION SYSTEM," *No.: WO/1991/008538*, 1991.

[9] R. Perry and R. Lancaster, "Enterprise Content Management: Expected Evolution or Vendor Positioning?" *Yankee Group Report*, 2002.

[10] R. Sandhu and P. Samarati, "Authentication, access control, and intrusion detection," *The Computer Science and Engineering Handbook*, vol. 1, pp. 929–1, 1997.

[11] R. Sandhu, "Lattice-based access control models," *Computer*, vol. 26, no. 11, pp. 9–19, 1993.

[12] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[13] T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible markup language (XML) 1.0," *W3C recommendation*, vol. 6, 2000.

[14] W. Fan, C. Chan, and M. Garofalakis, "Secure XML querying with security views," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. ACM New York, NY, USA, 2004, pp. 587–598.

[15] W. Ni and T. Ling, "Update XML data by using graphical languages," in *ACM International Conference Proceeding Series; Vol. 334*. Australian Computer Society, Inc. Darlinghurst, Australia, Australia, 2007, pp. 209–214.

[16] G. Cong, "Query and Update Through XML Views," *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 4777, p. 81, 2007.

[17] B. Choi, G. Cong, W. Fan, and S. Viglas, "Updating Recursive XML Views of Relations," *Complexity*, vol. 2, no. 2, p. 2, 2007.

[18] E. Damiani, M. Fansi, A. Gabillon, and S. Marrara, "Securely Updating XML," *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 4694, p. 1098, 2007.

[19] J. Clark, S. DeRose, *et al.*, "XML path language (XPath) version 1.0," *W3C recommendation*, vol. 16, p. 1999, 1999.

[20] A. Laux and L. Martin, "XUpdateŁXML Update Language," *XML: DB Working Draft*, pp. 09–14, 2000.

[21] W. Cao, G. Dai, Y. Xia, and D. Mu, "Technology of Information Hiding Based on Text Document," *Application Research of Computers*, vol. 20, pp. 39–41, 2003.

[22] C. Chao, W. Shuozhong, and Z. Xinpeng, "Information Hiding in Text Using Typesetting Tools with Stego-Encoding," in *Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on*, vol. 1, 2006.

[23] D. Rentz, "Microsoft Compound Document File Format."

Authorized licensed use limited to: SRM Institute of Science and Technology. Downloaded on August 04,2023 at 07:03:36 UTC from IEEE Xplore. Restrictions apply.