ISSN 2347 - 3983
Volume 8. No. 5, May 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter68852020.pdf
https://doi.org/10.30534/ijeter/2020/68852020

Joint Implementation of Mandated and Role-Based Delimitation of Access to Information Flows in Info-communication Systems

Mir-khusan Kadirov[1], Dilfuza Tadjibaeva[2], Alisher Rasulev[3], Farida Islomova[4]
[1]Department Information Technologies, Tashkent State Technical University named after Islam Karimov, Tashkent, Uzbekistan, mirxusan.qodirov@tdtu.uz
[2]Department Informatics, Automation and Control, Tashkent Chemical-Technological Institute, Tashkent,Uzbekistan, tadjibaevad.1978@gmail.com
[3]Department Informatics, Automation and Control, Tashkent Chemical-Technological Institute, Tashkent,Uzbekistan, rasulev1977@gmail.com
[4]Department Informatics, Automation and Control, Tashkent Chemical-Technological Institute, Tashkent,Uzbekistan, dildoranosirova073@gmail.com

ABSTRACT

Protection of data integrity, which is granted access to subjects in the information system, is achieved due to the fact that the information system is presented as part of the formal security model of logical mandatory and role-based access and information flow management and integrity control. Based on the de jure and de facto rules, an algorithm has been developed for the implementation of mandatory and role-based access control access to check the security status of a secure information system.The main idea is to prevent the role of violating entities of the protected information system from using role parameters, including access rights of roles to entities, for implementing information flows from memory or time from entities with a high level of confidentiality to entities with a low level of confidentiality of information.

Key words: Object, subject, access, information flows, access control, integrity control, de jure rules, de facto rules.

1. INTRODUCTION

Currently, most info-communication systems are built on the basis of database management systems. Database management systems provide a universal mechanism for storing, processing and providing data that can be used in various fields of human activity. The use of database management systems as a reliable basis for building info-communication systems is associated with the problem of differentiating user access to stored information. According to the analysis of the causes of security breaches [1, 2], 69% of the deficiencies in the means of protection in info-communication systems account for precisely the access control system. Among the models used for access control, the most common are mandatory and role-based. A feature of mandatory access delimitation is the prevention of the possibility of an intentional or accidental decrease in the value of information due to the control of information flows. In addition, with the help of the mandatory model, it is possible to simplify the task of administering info communication systems significantly during installation and operation, which reduces the number of errors. A mandatory access model provides high data protection efficiency and is closer to the needs of real life. All modern databases use the concept of roles to issue permissions to users, thus implementing role-based access control [3, 4]. Modern information processing systems are a combination of several specialized information services [5, 6]. For each type of information services its own mathematical model has been developed, which is not directly applicable to composite systems. Therefore, the task of modeling composite computer systems is very relevant.

2.IMPLEMENTATION IN THE INFORMATION SYSTEM

The combination of role and mandatory access control without taking into account the specifics of each of them can adversely affect the security of the obtained solution, including while providing protection from prohibited information flows. For example, forbidden information flows over time can be created by cooperating entities using assignment and revocation of access rights of roles at agreed time points [7].

A security policy can be built both informally, in the form of rules of instructions, and formally, in the form of a rigorous mathematical model. Despite the fact that the informal approach is easier to develop and implement, it is significantly inferior to the formal one in reliability, since it does not allow rigorous evidence of guaranteed security. This circumstance is reflected in various documents defining the security classes of computer systems [8, 9, 10].

Formally, when implementing the method, in general, the information system $\sum(G^*, OP)$is represented by the set of all its states - G^* and the set of state transformation rules - OP. Moreover, each state of the information system $\sum(G^*, OP)$is represented by a tuple (PA, A, F) and includes the following elements in its description [11].

3.DE JURE AND DE FACTO STATE CONVERSION RULES

Table 1: De jure rules for transforming states of an information system $\sum(G^*, OP)$.

The rule	$aceess\_own(x,y)$	$acees\_write(x,y)$	$aceess\_read(x,y)$	$delete\_access(x,y,\alpha_a)$	$grant\_rights(x,r,y,\alpha_a$	$remove\_rights$ (x,r,y,α_r)
The initial state $G = (PA, A, F)$	$a(y) \le b(x)$, $m(y) \le n(x)$	$c(y) \le b(x)$, $k(y) \le n(x)$	$c(y) \le b(x)$, and $k(y) \le k(x)$	$(x,y,\alpha_a) \in A$	$(x,r,write_a) \in A$, $\alpha_r \in \{own_r, write\}$	$(x,r,write_a) \in A$, $(x,y,own_a) \in A$
Resulting state G' $= (PA', A', F')$	$A' = A \cup \{(x,y,own_a)\}$	$A' = A \cup \{(x,y,write_a)\}$	$A' = A \cup \{(x,y,read_a)\}$	$A' = A\{(x,y,\alpha_a)\}$	$A' = A, F' = F'$, $PA'(r') = PA(r')$	$A' = A, F' = F'$, $PA'(r') = PA(r')$

Table 2 : De facto information system state conversion rules $\sum(G^*, OP)$.

The rule	$flow\_memory\_access(x,y$	$flow\_time\_access(x,y$	$take\_flow(x,y)$	$find(x,y,z)$	$post(x,y,z)$	$pass(x,y,z)$
The initial state $G = (PA, A, F)$	$\alpha_a \in \{read_a, write_a\}$	R, (x, y, α_a) $\in$ A	(x, y, own_a) $\in$ A	$\alpha f, \beta f$ $\in \{write_m, write_t\}$	$(x,y,\alpha f) \in F$, $(z,y,\beta a) \in A$	$(y,x,\alpha_a) \in A$, $(y,z,\beta f) \in F$
Resulting state G' $= (PA', A', F')$	$A' = A, \{(y, x, write_m)\}$	F′ = F ∪ {(y, x, $write_t$)} $\{(x,y,write_a)\}$	$PA' = PA, F' = F'$, $A' = A\ \alpha \in \{write_m, write_t\}$	F' $= F \cup \{(x,z,write_m)\}$, F' $= F$ $\cup \{(x,z,write_t)$	$F' = F \cup \{(x,z,write_m)\}$, $F' = F \cup \{(x,z,write_t)$	$F' = F \cup \{(x,z,write_m)\}$, $F' = F \cup \{(x,z,write_t)\}$

In the information system $\sum(G^*, OP)$, at least, the following state transformation rules (de jure rules) are implemented, the conditions for their use in the current state and the results of their application in the subsequent state of the system are given in table 1.

De jure rules. The rules are intended to describe formally the following basic functions of the access control mechanism of secure information systems [12, 13]:

- creating, deleting, renaming, receiving or changing parameters of user accounts, roles, administrative roles, entities or "hard" links to them, subject-sessions;
- gain subject-session access to entities, roles, or administrative roles;
- changing access rights of roles or administrative roles to entities, subject to sessions, roles, or administrative roles;
- changing access rights of roles or administrative roles to entities, subject to sessions, roles, or administrative roles;

De facto rules. To justify the achievement of the technical result, at least the following rules for converting the states of the information system $\sum(G^*, OP)$are used, the conditions for the use of which in the current state and the resulting information flows in the subsequent state of the system are given in table 2.

3. SOFTWARE PRODUCTIVITY ASSESSMENT

Let A units of time be required to obtain information about one group, and B units of time is required to obtain information about one user record [14]. Let N users and M groups be registered in the system, then the completion time of the first stage of collecting system information can be estimated as

$$T_{s1} = AM + BN.$$

Let C units of time be spent on the comparison operation, then the time for determining the user's membership in the group is calculated as

$$T_{s2} = \sum_{i=1}^{M} u_i NC$$

where$u_i -$ is the number of users in group i. Then the total time of the stage of collecting information about the subjects of the system is estimated as

$$T_s = T_{s1} + T_{s2} = AM + BN + \sum_{i=1}^{M} u_i NC.$$

Let W units of time be cumulatively spent on underestimating and returning from the stack, the time of getting the list of subdirectories is X, and the time of getting the list of files is Y. $Z$is the time of getting information about one file. Then the time to obtain a list of objects can be estimated by the formula:

$$T_o = l(X + Y + W\sum_{i=1}^{Ml} k_i + Z\sum_{i=1}^{Ml} f_i,$$

where$l$ is the nesting depth, k_i and f_i respectively the number of directories and files at the $i-$th level of the directory tree.

Suppose l is the length of the list of the i −th file, e is the number of records about group j in list i, h is the number of records about the current user in list i, p is the number of groups the user belongs to.

$$T_d = Kl(pe + h) + E,$$

where$K$ is the time of retaining rights to the object in the internal structure of the program, $E$is the time of formation of the general list of rights.

Suppose that in a system of S subjects, O objects, and the time of copying a bit string takes τ units of time, then the time for creating the adjacency matrix:

$$T_m = \tau T_d(S + O)^2.$$

The total time for collecting information about the system and forming the internal structures of the program can be calculated as:

$$T_{si} = T_s + T_o + T_m$$

After the formation of internal program structures, the time for obtaining information on the access rights of user i to object j will be determined by the time of copying the bit string from the adjacency matrix, parsing the string, and displaying information on the screen. Denote this time by T_{ij}. Then the time of obtaining information about access of all users to the object j will be determined as:

$$T_{sj} = ST_{ij}$$

where$S$ is the number of subjects in the system. The time for obtaining information about user i access to all system objects:

$$T_{io} = OT_{ij}$$

where$O$ is the number of objects in the system.

Time to obtain information on access of all users to all objects:

$$T_{so} = T_{sj} + T_{io} = T_{ij}(S + O).$$

Now we reveal the dependence of the speed of the software product on the number of system objects. The parameters of the computing system on which the software product was tested are shown in table 3.

Table 3: Computing system characteristics for software product testing.

Processor model	AMD Athlon 64 x2 Dual-Core 6000
CPU frequency	3.01 MHz
Memory	2048 Mb
Hard Access Speed	67 Mb/sec
Operating system	Microsoft Windows 7 Professional 64 bit
Version Microsoft.NET	4,5
Number of users	5
Number of user groups	20

The average time for collecting information about the subjects of the system after 10 starts of the program was 18.4 s.

The dependence of the time of collecting information about files on the number of files is shown in Figure 1, when compiling the diagram, the number of files in a single-level directory changed, there were no sub-directories.

The dependence of the construction time of the adjacency matrix on the number of files is shown in Figure 2.

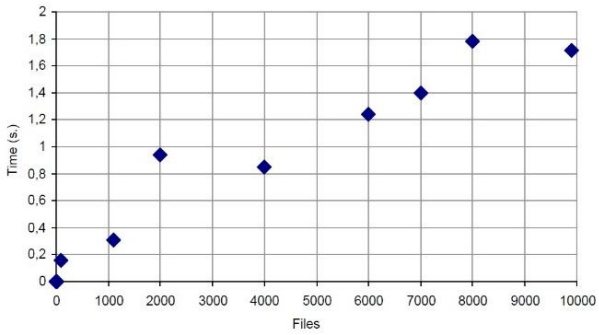

Figure 1: Diagram of the dependence of the time for collecting file information on the number of files

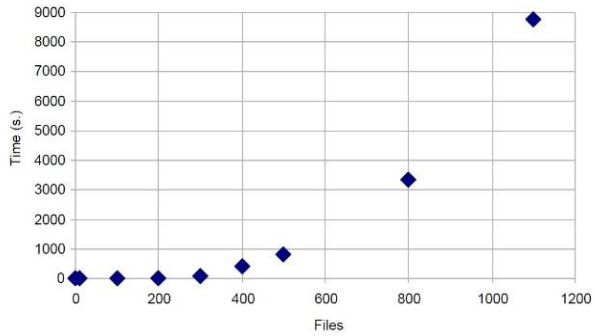

Figure 2: Diagram of the dependence of the construction time of the adjacency matrix on the number of files

From the above diagrams it can be seen that, if we neglect the error, the time for collecting information about files grows almost linearly depending on the number of files. Also, it can be seen that the construction time of the adjacency matrix for the access graph has an exponential dependence on the number of files. Thus, the experimental data confirm the validity of the theoretical performance assessment.

4. RESULT

To substantiate the results, we consider the following state:

- the initial state $G_0 = (\mathrm{PA}_0, \mathrm{A}_0, F_0)$ of the information system $\sum(G^*, OP)$, in which there are no subjects access to entities or roles and there are no information flows ($\mathrm{A}_0 = \varnothing, F_0 = \varnothing$);

- using mathematical logic and set theory they prove the impossibility of the transition of the information system $\sum(G^*, OP)$ from the initial state $G_0 = (\mathrm{PA}_0, \mathrm{A}_0, F_0)$ as a result of applying an arbitrary finite de jure sequence or "de facto" rules $\mathrm{op}_1, \ldots, op_N$defined in tables 1 and 2, respectively, in the state $G_N = (\mathrm{PA}_N, \mathrm{A}_N, F_N)$, where $G_0 \vdash_{op1} G_1 \vdash_{op2} \ldots \vdash_{opN} G_N, N > 0$ in which forbidden information flows exist by memory or time from entities or roles with a high level of confidentiality to entities or roles with low privacy level (stream type information $(\mathrm{x}, y, write_m)$ or $(\mathrm{x}, y, write_t)$, where $\mathrm{x}, y \in \mathrm{E} \cup R$ and $f_e(x) > f_e(y)$).

When implementing the algorithm, we obtain the following results:

- in relation to roles: access right own_r – owner of the role, $read_r$ – right to receive the role as current, view its parameters, $write_r$ – right to change many access rights of the role, $execute$ – right to access roles subordinate This role in the role hierarchy access own_a – ownership (change of parameters) of the role, $read_a$ –obtaining by the subject of the role as current, $write_a$ – access to change access rights of the role or composition of roles subordinate to it in the hierarchy;

- to prevent the possibility of using role access rights when creating forbidden information flows, roles are implemented by container entities (for example, in operating systems they create a "virtual" hierarchical file system, each element of which is a role, and the hierarchy of its elements corresponds to the role hierarchy) and sets the levels of privacy of roles (for example, operating systems use extended attributes of file system elements for this);

- all components of a secure information system (subjects, entities and roles) are divided into two disjoint sets that affect the integrity and security of the system or not, and assign them the corresponding integrity levels i_high and i_low;

- the $execute\_container$ function used in the formal description of the $\Sigma\ (G\ *, OP)$ information system is not directly implemented in the information system, since its value is calculated each time the subject accesses the entity when sequentially checking access rights to container entities containing this entity, starting with root container entity.

5. CONCLUSION

To sum up, we suggest the following outcomes regarding proposal paper:

-the possibility of building a consistent security policy in a single computer system that includes role-based and mandatory access control based on the existing role-based or mandatory security policy has been proved;

-when applying the described method in a secure information system, they implement a single mechanism of mandatory and role-based access control, and as a result prevent the ability of violating entities to use role parameters to create prohibited information flows from memory or time from entities with a high level of confidentiality to entities with a low level of confidentiality of information;

- the performance assessment of the developed software product module for analyzing access to file system objects was carried out.

REFERENCES

[1] https://www.sei.cmu.edu/about/divisions/cert/index.cfm

[2] https://www.ptsecurity.com/ru-ru/research/analytics/cybersecurity-threatscape-2019/

[3] Kuhn D. R. **Implementation of role-based access control in multi-level secure systems**: p. 6023765 USA. – 2000.

[4] Chatterjee A., Pitroda Y., Parmar M. **Dynamic Role-Based Access Control for Decentralized Applications** //*arXiv preprint arXiv*:2002.05547. – 2020.

[5] Woźniak M., Graña M., Corchado E. **A survey of multiple classifier systems as hybrid systems** //*Information Fusion*. – 2014. – T. 16. – C. 3-17. https://doi.org/10.1016/j.inffus.2013.04.006

[6] Sagatov M., Irgasheva D., Mirhusan K. **Construction Hardware Protection Infocommunication Systems from Network Attacks** //*Proceedings of International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE)*, 2015. – P. 271.

[7] P.N. Devyanin, V.V. Kuliamin, A.K. Petrenko, A.V. Khoroshilov, I.V. Shchepetkov. **Using Refinement in Formal Development of OS Security Model**. *In Lecture Notes in Computer Sciences #9609 "Perspectives of System Informatics: 10th International Andrei Ershov Informatics Conference", Springer International Publishing*, 2016, pp. 107-115. DOI: 10.1007/978-3-319-41579-6_9.

[8] Jaidi F., Ayachi F. L. **An approach to formally validate and verify the compliance of low level access control policies** //*2014 IEEE 17th International Conference on Computational Science and Engineering*. – IEEE, 2014. – С. 1550-1557. https://doi.org/10.1109/CSE.2014.287

[9] Devyanin P.N. **Security models for computer systems. Control of access and information flows**. *Textbook for higher schools. 2nd ed. M.: Goryatchayaliniya – Telecom*, 2013. 338 p.

[10] Jaïdi F., LabbeneAyachi F., Bouhoula A. **A methodology and toolkit for deploying reliable security policies in critical infrastructures** //*Security and Communication Networks*. – 2018. – Т. 2018. https://doi.org/10.1155/2018/7142170

[11] Mir-khusanKadirov, ZoxidjonTulyaganov, NodirakhonTojikhujaeva, Nazimakhan Karimova. **Development of an algorithm for implementing mandatory and role-based access control**. *International Journal of Emerging Trends in Engineering Research (IJETER), Vol. 8, Issue 4*, India 2020, P.1027-1033. https://doi.org/10.30534/ijeter/2020/12842020

[12] Patent RU 2525481. **Method for making security of information flows in protected information systems with mandate and role access managemen***t*. Devyanin P. N. Publ. –20.08.2014.

[13] Devyanin P.N. **On the problem of representation of the formal model of security policy for operating systems**. *Trudy ISP RAN/Proc. ISP RAS*, vol. 29, issue 3, 2017.p. 7-16. DOI: 10.15514/ISPRAS-2017-29(3)-1

[14] Miguel J. P., Mauricio D., Rodríguez G. **A review of software quality models for the evaluation of software products** //arXiv preprint arXiv:1412.2977. – 2014.