Quantitative Evaluation of Information System Security

R. Ortalo, Y. Deswarte
LAAS-CNRS
7, avenue du Colonel Roche, 31077 Toulouse cedex 4, France
Tel.: +33/ 5-61-33-62-00, Fax: +33/ 5-61-33-64-11
E-mail: ortalo@laas.fr, deswarte@laas.fr

Abstract

This paper presents a method for the evaluation of the security of information systems. First, we outline briefly the overall guidelines of the method: specification of the security policy, description of the vulnerabilities of the target organization, and a quantitative evaluation approach based on the privilege graph model. Then, the paper presents how the method applies to the description of the security requirements of a real organization: a medium-size bank agency. To illustrate the interest of the security measures, this example is used as a basis for applying the evaluation methodology taking into account some vulnerabilities of the bank agency.

Keywords

Quantitative evaluation of security, information systems, privilege graph.

1 INTRODUCTION

An information system encompasses both a data processing system and a human organization which uses it. Such information systems become more and more vital for most companies. At the same time, these systems are made more and more vulnerable by new user requirements, and increased flexibility. New services such as information sharing facilities, interconnection to insecure networks, or powerful desktop applications may hide serious security flaws. However, for most organizations, security is not the only concern and they are not prepared, for the sake of security, to waive their system ease of use, to give up information sharing facilities, and to endanger their competitiveness with restrictive control procedures.

Therefore, the main task of information system managers is to maintain a satisfactory level of security, without impeding the operation of the system and the organization. And this is not an easy task: usual security policies, such as the Bell-LaPadula multilevel policy, generally imposes the enforcement of mandatory rules which may not fit the actual organization. Furthermore, security administrators should be able to assess the security level achieved by the organization *in operation*, and to compare the impact on security of different possible mechanisms. However usual security evaluation methods, such as evaluation criteria (ITSEC, 1991) or risk analysis (Anderson, 1991) do not help for that because they focus on the information system design, rather than on the actual system operation.

In this work, we present a theoretical and practical method for the quantitative evaluation of security that can help a security administrator to improve the security of an information system. The method first implies a description of the security needs of the information system through the specification of its security policy. The security policy gives a description of an ideal information system. This specification should come with a description of the real system, including its vulnerabilities, that gives the opportunity to evaluate quantitatively the system with respect to its security objectives defined in the policy. The measurements delivered by the evaluation method should represent as accurately as possible the security of the system in operation, i.e. the difficulty for an attacker to exploit the vulnerabilities present in the system to defeat its security objectives. Finally, this methodology allows the security administrator to propose modifications in the system, which correspond to eliminate of reduce some of the vulnerabilities identified, and evaluate the corresponding security improvements.

The structure of this paper is the following. Section 2 presents an overview of the security policy specification and evaluation method. Section 3 develops a practical application example, addressing the security needs of a small bank agency, to illustrate the various steps of the method. Finally, section 4 draws a conclusion.

2 METHOD OVERVIEW

2.1 Security policy specification

The first step of the method involves the description of the security needs of the system through the definition of its security policy. According to (ITSEC, 1991, §2.9), the **security policy** is "*the set of laws, rules and practices that regulate how sensitive information and other resources are managed, protected and distributed within a specific system*". In this paper, we consider that a security policy defines:

- the **security objectives**, i.e. the confidentiality, integrity and availability properties expected of the system;
- and the **security rules** which are imposed on the mechanisms which can modify the security state of the system, in order to guarantee the security properties.

The definition of the expected security properties may imply the description of some of the internal elements of the system (such as specific subjects or objects, available operations, or an organization chart for example). These **description elements** are included into the security policy and form a partial description of the system and its functions.

The specification language we use to describe the security policy is inspired by the deontic logic language presented in (Cholvy and Cuppens, 1997). This language allows a flexible and rigorous formulation of the security policy that allows the policy to fit the organization.

2.2 Modeling vulnerabilities of the organization

The security policy deals with the expected behavior of the information system. In order to obtain an evaluation of the actual security, it is also necessary to consider residual vulnerabilities existing in the real system. Then, the system security could be assessed quantitatively by a measure of the effort that an attacker should spend to exploit these vulnerabilities and defeat the security objectives.

Therefore, the first step of the evaluation method consists in the identification of the various vulnerabilities existing in the system. The security policy already describes a set of security mechanisms implemented in the system. Any such mechanism can be broken or bypassed if sufficient effort is spent by an attacker and thus is a potential vulnerability. But, usually, other information can also help in finding vulnerabilities related to the operation of the system. For example, the study of the security failures of other similar organizations indicates some of the potential vulnerabilities of the evaluated information system. Similarly, the study of the overall operation of the organization leads to the identification of the different low-level operations that build up a complex task. In order to identify vulnerabilities, it is possible to analyze the difficulty to bypass such low-level operations and take advantage of the natural behavior of the organization to defeat security mechanisms. The search for vulnerabilities in the system can also be compared to the validation procedure performed in the context of a security evaluation according to normalized criteria, and methodologies designed to perform such evaluation can be reused in this context. Once a satisfying list of the various potential vulnerabilities of the organization has been built, the organization should be checked in order to determine the vulnerabilities that can effectively be exploited by a potential attacker. This analysis leads to the construction of a model representing the vulnerabilities of the organization.

We have adopted a model previously developed for representing computing systems vulnerabilities, called a privilege graph (Dacier and Deswarte, 1994). In this model, a privilege is defined as a set of rights owned by a subject representing the authorization for the subject to execute operations on objects. The nodes of the privilege graph represent sets of privileges. An arc exists between two nodes if, given the privileges defined by the origin node, there exist a method enabling to obtain the privileges corresponding to the target node. Therefore, the privilege transfer methods associated to these arcs correspond to the vulnerabilities of the system. Such vulnerabilities may correspond to flaws of the system, but they may be associated to authorized and useful privilege transfer mechanisms (such as delegation). Each arc in the privilege graph is identified by a name corresponding to the privilege transfer method. An example of such graph is given in Figure 1.

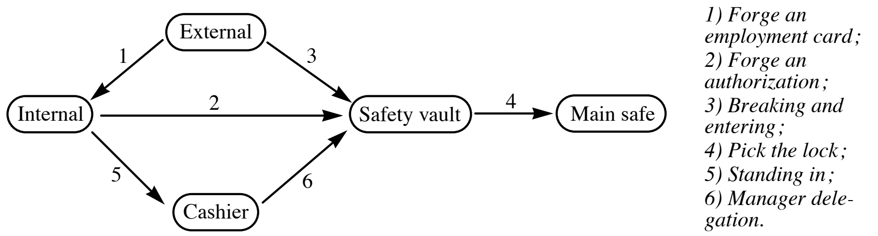


Figure 1 Privilege graph example.

In this figure, ‘External’ refers to an individual who does not belong to the organization, ‘Internal’ refers to an individual who belongs to the organization and has easy access to the various parts of the office, ‘Cashier’ refers to a specific position in the organization, ‘Safety vault’ and ‘Main safe’ refer to the fact that one has physical access to the safety vault or the main safe respectively. The labels associated to each arc indicates the method that can be used to obtain new privileges, starting with others. For example, in order to access the safety vault, an external individual may follow various paths: directly try to access it by breaking and entering, or indirectly access it by successively forging two documents: first an employment certificate (with which he could gain access to the bank), and then a faked authorization to access to the safety vault (e.g. for maintenance). A more complicated method could involve provoking a disease of the usual cashier, and then standing in this position to use the delegation usually granted by the agency manager.

2.3 Security measure computation

In this vulnerability model, to each privilege transfer method considered in the graph we can associate a quantitative value corresponding to the effort needed for an attacker to exploit the method successfully. Security evaluation criteria provide several guidelines with respect to the assessment of this effort. For instance, the ITSEC suggest three classes for the mechanisms strength rated basic, medium or high (ITSEC, 1991, §3.5-3.8). The evaluation manual associated to these criteria provides more detailed means for assessment during evaluation using four parameters: expertise, collusion, time and equipment (ITSEM, 1993, §3.3.29-32,6.C.28-34). All these values can be gathered in an effort value that is associated to an arc in the privilege graph.

Furthermore, the security objectives of the system identify in the graph several sets of privileges to be protected (called the target nodes) from some other sets of privileges (called the attacker nodes). Therefore, it is possible to define a quantitative measure of security as the value corresponding to the mean effort needed for the attacker to obtain the privileges of the target, taking into account all the paths existing in the privilege graph, and the weights associated with each vulnerability. Such value should represent the overall difficulty to defeat the security objectives of the system thanks to the vulnerabilities it contains. The evaluation depends on the basic quantitative values associated to each vulnerability, but it depends also on the allowed combinations of several vulnerabilities. Indeed, in most situations, it is probable that exploiting only one vulnerability does not directly defeat a security objective of the system, but is a step in this direction. Therefore, in order to obtain a quantitative evaluation of the system, it is necessary to consider the opportunity for an attacker to exploit a combination of several of these vulnerabilities.

In (Dacier *et al.*, 1996), such a measure, denoted METF for Mean Effort To Failure*, is defined with respect to the set of all possible scenarios deduced from the privilege graph, called an intrusion process. It is needed to state several high-level assumptions on the behavior of the attacker to define completely the intrusion process.

By using this quantitative security evaluation technique, it is possible to observe the measure evolutions corresponding to modifications of the privilege graph. These modifications correspond to potential corrections or changes in the running of the organization, which impact on the overall security can be analyzed. Therefore, the system administrator who proposes modifications of the system operation in order to improve its security can justify his proposal with precise and quantitative data extracted from the system. Furthermore, the system administrator can compare the security improvements brought by different proposals, with the possible additional constraints imposed on the organization. Thus, a quantitative measure of security helps him to manage the trade-off between functionality and security in the organization.

* This METF measure has been chosen by analogy with the usual MTTF reliability measure.

3 EXAMPLE

To illustrate the various steps of the method we present a real application example in this section. The organization target of this experiment is a small bank agency, counting around 30 employees, and located in a rural area.

3.1 Security policy definition

The security policy presented in this section was built on the basis of several internal documents describing the structure of the organization, the various positions appearing in it, and the numerous operations appearing in a bank. The analysis of these documents has been completed by a study in the field of several days.

We use a graphical and hierarchical notation to define the various description elements needed to formulate the specification. A set counting several elements is represented by a frame, entitled by the set name, and containing the various subsets names. Further refinement is shown using tabulations (e.g. see the definition of 'Action types' in Figure 2). Furthermore, we allow the use of sets in the formulas describing the functioning of the organization (see Figure 3 for examples). Due to space limitations, the precise definition of the language extension needed to use this notation in a deontic logic language has not been included in this paper.

The description elements of the specification are presented in Figure 2. First, we note in this figure the various individuals mentioned in the policy (among which the agents of the organization itself), a set of roles (described in the following) and several actions. The mapping of each agent to roles is defined separately. Miscellaneous elements needed for the description are also defined here, as well as the actions types. The description of the general actions taken into account is limited here to: bank operations performance, loan agreement, and physical access. These various actions are sufficient here to describe the functional rules needed.

Details of some of the description elements are also given in this figure. We define the various operations existing in the bank at a high level, several secondary elements, and the various roles used in the description. These roles allow to define generic description rules or security rules independently of the positions of the actual agents in the organization. In our case, the various roles are related to the different functions appearing in the organization. The mapping of agents to roles is found in the set 'Roles mapping'. It describes the function(s) associated to each employee in the bank agency. Finally, the basic description elements of the specification include a simple representation of the data associated to a bank account.

Basic elements define the vocabulary with which the security policy is built. Additional description rules and security rules are added to the specification in order to include a simple representation of some of the operations performed in the organization, and the mechanisms related to the possible evolutions of the security state. We show in Figure 3 how accounts and money movements can be represented by several simple description rules. For example, we see that loan authorization implies a credit operation, or that a negative balance automatically tags some operations on the account as frozen. In this figure, we integrate several security rules in the description. To formulate these rules, we use the deontic logic operators provided by the specification language, denoted **O**, **P** and **F**, that are read respectively as "Obligation of", "Permission of" and "Interdiction of" (Chellas, 1980, §6; Cholvy and Cuppens, 1997). For example, we see that it is permitted that an agent does usual operations only if he is a registered employee of the bank (first rule). We see also that agents performing a frozen operation are obliged to tag this operation as forced (forth rule).

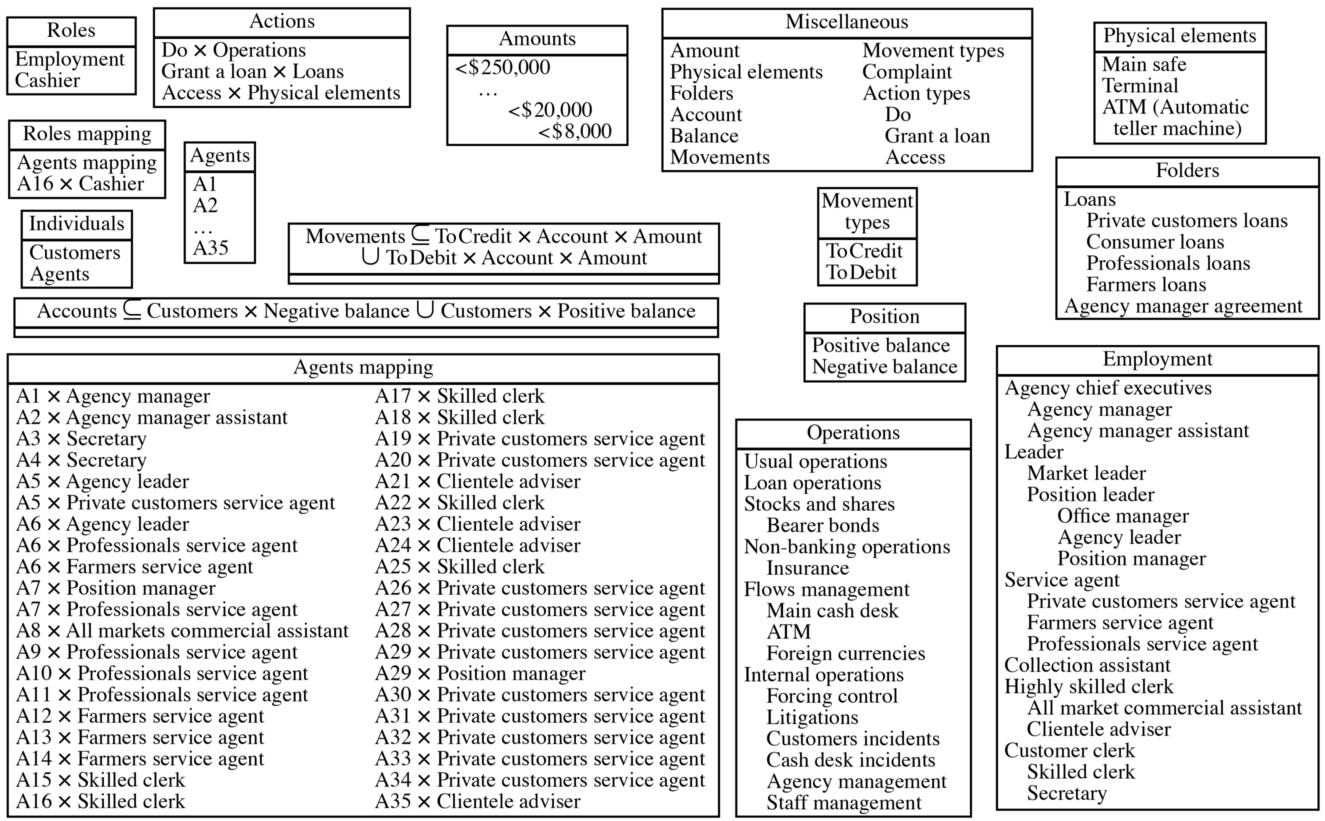


Figure 2 Security policy specification: description elements.

Among these rules, we focus on a specific aspect involving delegation of privileges: loan agreement delegation. Chief executives are allowed, in the bank, to delegate some of their privileges to other employees, in the limit fixed by several conditions.

Finally, Figure 3 presents three simple security objectives. The first one is related to customers, but the two others apply to the agents of the organization. The second one aims at enforcing the delegation authorizations with respect to loan agreement (with respect to loans of more than a given amount), and the last one at protecting customers from abuses perpetrated by employees.

Description rules
Agents × Actions ⇒ Actions
Agents × (Do × Operations) ⇐ Agents × (Access × Terminal)
Do × Loan operations ⇒ Loans
Do × Usual operations ⇒ ToDebit × Account × Amount ∨ ToCredit × Account × Amount
Customer × Negative balance ⇒ Freezing × (ToDebit × Account × Amount)
Grant a loan × Loans ⇒ ToCredit × Account × Amount

Security rules
P(Agents × (Do × Usual operations)) ⇒ (Agents × Employment)
P(Agents × (Do × Loan operations)) ⇒ (Agents × Service agent)
P(Agents × (Access × Main safe)) ⇒ (Agents × Cashier) ∨ (Agents × Agency manager)
Agents × Actions ∧ Freezing × Actions ⇒ **O**(Forcing × Agents × Actions)
P(Agency manager delegation)
Delegation rule

Delegation rule
Agency manager delegation ∧ (Agents × Employment) ∧ Loan authorizations ⇒ **P**(**P**(Agents × (Grant a loan × Loans)))

Loan authorizations
Professionals market
Service agent × (Professionals loans × <$70,000)
Leader × (Professionals loans × <$90,000)
Agency chief executives × (Professionals loans × <$110,000)
Farmers market
Service agent × (Farmers loans × <$70,000)
Leader × (Farmers loans × <$90,000)
Agency chief executives × (Farmers loans × <$110,000)
Private customers market
Service agent × (Private customers loans × <$40,000)
Leader × (Private customers loans × <$90,000)
Agency chief executives × (Private customers loans × <$140,000)
Consumer projects
Service agent × (Consumer loans × <$8,000)
Leader × (Consumer loans × <$18,000)
Agency chief executives × (Consumer loans × <$28,000)
Companies market
Leader × (Consumer loans × <$120,000)
Agency chief executives × (Consumer loans × <$240,000)

Security objectives
F(Customers × (Do × Usual operations))
F(Agent × Grant a loan × Loans ∧ (Agents × Employment) ∧ ¬Loan authorizations ∧ Loans × ¬(<$100,000))
F(Customers × Complaint)

Figure 3 Security policy specification: description and security rules.

3.2 Quantitative evaluation

Given the proposed security policy, we are interested in the study of the security objectives presented in Figure 3 with respect to several vulnerabilities typical of an organization essentially consisting of human beings.

3.2.1 Integration of vulnerabilities

The two vulnerabilities considered in the organization are described in Figure 4. The first one is related to trust relationships between the various agents of the organization. We also consider that some customer may trust the agent in charge of his business (second vulnerability). Such trust may allow a malicious agent of the organization to misappropriate funds of this customer. However, in our case, we think that this vulnerability arise only for specific operations, more precisely anonymous stocks and shares operations.

Vulnerabilities
Agents × Trust × Other agent ∧ **P**(Agents × Actions) ⇒ **P**(**P**(Other agent × Actions))
Customer × Trust × Agents ∧ Agents × Bearer bonds ⇒ **P**(¬(Customer × Complaint))

Figure 4 Vulnerabilities.

Table 1 Attack success rates

Rate	*Value*
λ_a	1
λ_c	1/2

In order to compute quantitative security measures, values are associated to these two vulnerabilities. The success rate are shown in Table 1, where λ_a represents the success rate of the vulnerability consisting in abusing of a colleague's trust, and λ_c of a customer's trust. The relative value of these two parameters is the most important point. These values have been chosen during an interview with several executives of the bank, and we can see that they consider it is twice more difficult for an employee to abuse a customer than a colleague.

3.2.2 Privilege graph construction

Identification of the impact of the two vulnerabilities presented in Figure 4 on the organization implies a new analysis of its security objectives. We need to determine if such vulnerabilities allow the violation of some of the security objectives of the organization. When it is the case, building the corresponding privilege graph (or intrusion process) is the first step before computing quantitative measures.

We consider first the second security objective shown in Figure 3 related to loan agreement delegation. We consider the negation of the objective to exhibit the various privileges it refers to. Given the organization operation description, and more precisely the authorizations shown in Figure 3, the negation of this objective corresponds to the various permissions defined in (1), i.e. the privileges of the chief executives and of the leaders who are the only agents authorized to grant a loan of the given amount.

$$\begin{cases} \mathbf{P}(\text{A1} \times (\text{Loans} \times \neg(<\$100{,}000))) \\ \mathbf{P}(\text{A2} \times (\text{Loans} \times \neg(<\$100{,}000))) \\ \mathbf{P}(\text{A5} \times (\text{Consumer loans} \times <\$120{,}000)) \end{cases} \begin{cases} \mathbf{P}(\text{A6} \times (\text{Consumer loans} \times <\$120{,}000)) \\ \mathbf{P}(\text{A7} \times (\text{Consumer loans} \times <\$120{,}000)) \\ \mathbf{P}(\text{A29} \times (\text{Consumer loans} \times <\$120{,}000)) \end{cases} \quad (1)$$

Given the vulnerabilities considered, an agent can obtain the privileges of other agents only by abusing their trust. The study performed in the banking agency enabled us to identify the various cases in which such trust is sufficiently high to consider that the trusted agent can really obtain new privileges. Therefore, we build directly the privilege graph corresponding to this vulnerability shown in Figure 5, in which all arcs are associated to the same success rate value λ_a .

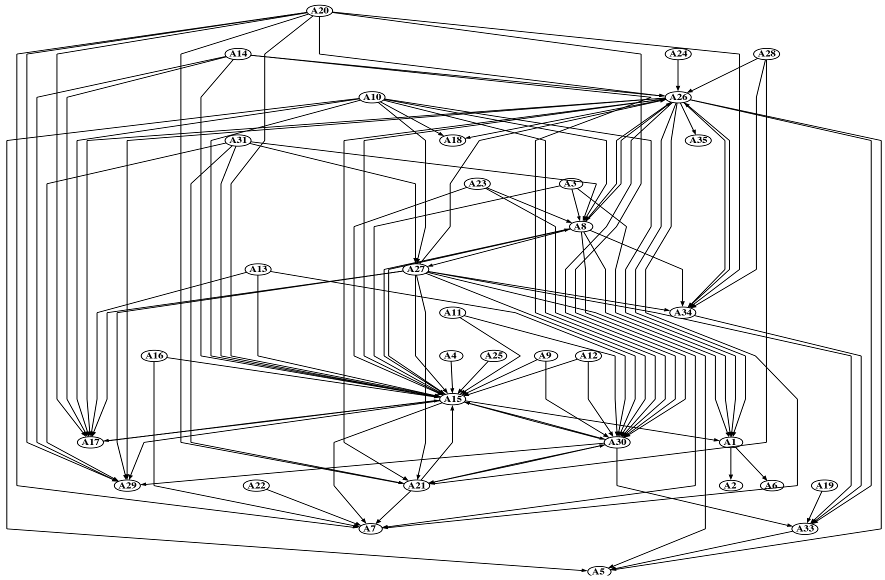
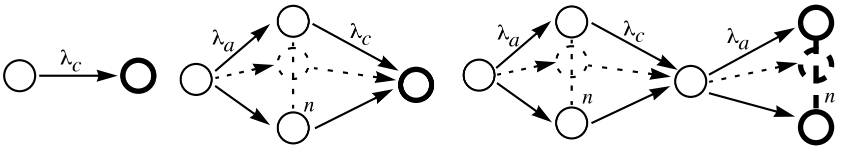


Figure 5 Privilege graph built considering the first vulnerability only (Fig. 4) with respect to the second security objective (Fig. 3)[a,b].

a. None of the arcs starting from one of the *targets* (A1, A2, A5, A6, A7 and A29) has been drawn.
b. This figure was built automatically by the graph visualization tool *daVinci* (Fröhlich and Werner, 1994).

Considering the second vulnerability (Figure 4) and the third security objective (Figure 3), the intrusion process defeating this objective, shown in Figure 6-a, is very simple and corresponds to exploiting the vulnerability. In this situation, we envisage modifications of the operation of the organization that would make the process harder. A natural solution consists in setting up a validation procedure. In this case, each time an agent performs an anonymous stocks and shares operation, the agreement of a second agent is required. Such procedure is feasible in practice, for example by associating different agents to the commercial negotiation task, and bonds delivery task. Given this proposal, a fraudulent operation would imply the use of two types of vulnerabilities: first a malicious agent should abuse one of his customers to avoid that he suspects misappropriation, and then he should also abuse the trust of some of his colleagues to obtain their validation (for the issue or conversion of the bearer bonds). In these cases, we obtain the intrusion processes presented in Figure 6-b and Figure 6-c, where λ_a is the success rate of the attack corresponding to abusing the trust of another agent, λ_c the success rate of the attack corresponding to abusing a customer, and n is the number of agents in the organization.



(a) Initial case **(b)** Validation at bonds issuing **(c)** Validation at bonds issuing and conversion

Figure 6 Intrusion process corresponding to the two vulnerabilities (Fig. 4) with respect to the third security objective (Fig. 3).

3.2.3 Measures

Once the privilege graph (or the intrusion process) is built, computation of the quantitative measures is done. In the context of our example, the objective of this computation is to show that the whole quantitative evaluation method can be applied successfully to an organization. These results allow us to study the operation of the organization and to compare possible evolutions. Given the restricted example that we study in this paper, the practical information brought by our evaluation is limited, but it shows the information that could bring a real application (which should extend the specification target and the number of vulnerabilities studied). With the privilege graph of Figure 5, we obtain the evaluation results presented in Table 2.

Table 2 Evaluation results (authorizations enforcement for loans <$100,000)

Att.	*NO*	*SP*	METF
A1			
A2			
A3	261	1	0.477
A4	65	2	1.277
A5			
A6			
A7			
A8	195	1	0.291
A9	137	2	0.834
A10	671	1	0.285
A11	137	2	0.834
A12	137	2	0.834
A13	202	2	0.768
A14	343	2	0.398
A15	65	1	0.277
A16	66	1	0.611
A17	65	2	1.277
A18	0	—	—
A19	1	2	2.000
A20	549	1	0.282
A21	79	1	0.514
A22	1	1	1.000
A23	332	2	0.624
A24	140	2	1.194
A25	65	2	1.277
A26	65	2	0.194
A27	272	1	0.273
A28	345	2	0.705
A29			
A30	72	1	0.476
A31	612	1	0.374
A32	0	—	—
A33	1	1	1.000
A34	126	2	0.888
A35	0	—	—

The results mention measures when each agent is assumed to be an attacker (*Att.*). The second column (*NO*) indicates the total number of paths existing between the attacker and the various target agents identified previously. The third column (*SP*) indicates the length of the shortest path enabling to defeat the security objectives. Finally, the last column presents the measure METF that evaluates the effort needed by the attacker to reach its target taking into account all the possible paths between them and the values associated to each vulnerability. As can be seen, this last measure exhibits the most interesting behavior for security evaluation.

These values correspond to the various opportunities to exploit the trust that other agents may place into a specific employee. Individuals in the organization that are most trusted by other people are, of course, those that have the best opportunities to break the second security objective. As it is surely impossible (and probably ill-fated) to regulate the everyday life and the trust relationships existing among the agents, these results do not allow one to influence directly the operation of the organization. However, comparison of these results with the situation of each agent (with respect to his seniority, position or skills) could reveal anomalies in privileges distribution among the organization. In our case, there is an homogeneous distribution of the results. No pathological tendency arise from the analysis, and we do not see any symptom of malfunctioning. One would only note that trust relationships are very strong among the various agents in this domain of activity.

The security measures corresponding to the three intrusion processes presented in Figure 6 can be obtained analytically. These equations are shown in Table 3.

(a) Initial case	**(b)** One validation	**(c)** Two validations
$METF_{(a)} = \frac{1}{\lambda_c}$	$METF_{(b)} = \frac{1}{n\lambda_a} + \frac{1}{\lambda_c}$	$METF_{(c)} = \frac{2}{n\lambda_a} + \frac{1}{\lambda_c}$

Table 3 METF evaluation

Numeric values, obtained for several values of parameter *n*, are presented in Table 4. The values show that the security improvement brought by a validation is significant only for small values of *n*. This simply means that it is less probable that a malicious agent is able to take advantage of the trust relationships he maintains if few persons are authorized to validate a bearer bonds operation.

Table 4 METF values for several *n*

n	$METF_{(a)}$	$METF_{(b)}$	$METF_{(c)}$
1	2	3	4
2		2.5	3
3		2.333	2.666
4		2.25	2.5
5		2.2	2.4
6		2.166	2.333

3.2.4 Suggested improvement

Finally, a proposal for modification of the operation of the organization would consist in adding a new mandatory validation step for bearer bonds delivery. This validation privilege should be given to one or two employees only. Such a validation step may be associated to bonds issuing, or bonds conversion, or both operations.

4 CONCLUSION

The information systems targeted in our study are commonly found in organizations, such as banks, industrial companies, etc. For such organizations, we propose a two levels approach to ensure that the security objectives are correctly addressed. At the design level, the specification of the security policy of the information system implies the definition of the security objectives of the organization, and the various security mechanisms existing in it, as well as a general description of its functioning. But at a second level, a pragmatic evaluation technique is needed for achieving a good compromise between security and efficiency in the information system. This technique allows the impact of vulnerabilities to be assessed and modifications to be selected to improve security without impeding system operation. In this paper, we present a method corresponding to this view. This method involves three parts:

- a description of the security policy of the organization;
- a description of the system vulnerabilities;
- and a quantitative evaluation method that allows to rate the efficiency of successive proposals of vulnerability reduction.

This presentation has been illustrated by a practical application in the context of a real-world organization. This example shows the definition of the security policy of the organization. Starting with the information provided by the security policy, we show that a quantitative evaluation method allows the integration in the security analysis of the various vulnerabilities existing in the organization. Assessment of the impact of these vulnerabilities on the information system security objectives allows the security administrator to propose and compare possible improvements of the functioning. The example studied remains somehow limited, but the method applies directly to information systems exhibiting a complex functioning, and various vulnerabilities. Therefore, it allows to represent accurately the security needs of a large organization, and provides a mean to evaluate and improve its security.

In conclusion, we support the idea that it is possible and useful to define a general method for the security evaluation of information systems, including a specification method of the security needs, and an evaluation method addressing operational aspects. Adopting a formal language as well as a quantitative evaluation technique, the method presented in this paper corresponds to this objective and shows the interest of this approach for the security analysis of general information systems.

5 REFERENCES

Anderson, A.M. (1991) Comparing Risk Analysis Methodologies, in *Information Security*, Elsevier Science Publishers, 301-11.

Chellas, B.F. (1980) Modal Logic: An Introduction. Cambridge University Press.

Cholvy, L. and Cuppens, F. (1997) Analyzing Consistency of Security Policies, in *IEEE Symposium on Security and Privacy,* Oakland, USA, May 4-7 1997, 103-12.

Dacier, M. and Deswarte, Y. (1994) Privilege Graph: an Extension to the Typed Access Matrix Model, in *Third European Symposium on Research in Computer Security (ESORICS 94),* U.K., LNCS 875, Springer-Verlag, 317-34.

Dacier, M., Deswarte, Y. and Kaâniche, M. (1996) Models and Tools for Quantitative Assessment of Operational Security, in *12th IFIP Information Systems Security Conference (IFIP/SEC'96)*, Greece, May 21-24 1996, Chapman & Hall, 177-86.

Fröhlich, M. and Werner, M. (1994) Demonstration of the Interactive Graph Visualization System daVinci, in *DIMACS Workshop on Graph Drawing,* USA, 1995, LNCS 894, Springer-Verlag. (see also `http://www.informatik.uni-bremen.de/~davinci/`)

ITSEC (1991) Information Technology Security Evaluation Criteria. Office for Official Publications of the European Communities, Luxembourg, v1.2.

ITSEM (1993) Information Technology Security Evaluation Manual. Office for Official Publications of the European Communities, Luxembourg, v1.0.

6 BIOGRAPHIES AND ACKNOWLEDGMENTS

Rodolphe Ortalo received the engineer diploma from the french engineering school Supélec in 1994. He currently completes his Ph.D. in the Fault Tolerance and Dependable Computing research group of LAAS-CNRS.

Yves Deswarte is a Research Director of INRIA, and member of the Fault Tolerance and Dependable Computing research group at LAAS-CNRS. He is currently the chairman of the International Steering Committee of ESORICS, the European Symposium on Research in Computer Security.

The authors are grateful to the Crédit Agricole (CRCAM Quercy-Rouergue) for their cooperation, as well as to the whole staff of the agency of Villefranche de Rouergue for their gracious welcoming. This work has been partially supported by UAP, and by the European ESPRIT Project 20072 “Design for Validation” (DeVa).