Security Requirements Analysis of Web Applications using UML

Salim Chehida [1], Mustapha kamel Rahmouni [2]
[1] Department of Informatics, University of Mostaganem, Algeria
salimchehida@yahoo.fr
[2] Department of Informatics, University of Oran Es-Senia, Algeria
kamel_rahmouni@yahoo.fr

***Abstract*—** The security problems of the Web applications (processes and data) take a great importance nowadays. The transactions made through the network can be intercepted, more especially since adequate legislation has not yet been fully enforced on the Internet. The functional specification of the Web applications is not sufficient, the design and the realization of these systems must take into account the various security requirements. Taking into account the various security constraints (Availability, Authentication, Integrity, Secrecy, Non-Repudiation, etc.) in the modeling process constitutes one of the principal challenges for the designer of these systems. UML is the standard language for the modeling of the multiple views of systems by using the various mechanisms of extension. In this paper we describe our return on experiment concerning the modeling of the Web applications in order to analyze the security requirements of these systems by proposing new extensions of UML and a case study as illustration.

Keywords: *Web applications, Computer Security, Modeling, and UML.*

1. Introduction

If the generalization of the Internet connections offers new and promising possibilities, it also introduces a certain number of risks which we should be aware of, weigh their possible consequences, and take adequate measures. A company communicates today with its subsidiaries, its partners, and that induces a massive opening to information. The Web applications are thus increasingly likely to be the subject of various disturbances such as congestions, malicious accesses and attacks. The number of security problems has recently drastically increased and, unfortunately, this ascending curve certainly would not dip. In 2003, according to a study published [4], the damage caused by security incidents can amount, in Europe, between 0,2 and 0,5% of the sales turnover.

Security aspects of systems should be analysed and modeled during the entire system development process, so that the violated security requirements can be identified in the early stages of the development process. [17] UML is a standard language that is

used to visualize, specify, build and document a software system. This language is not adapted to all the system views: it uses extension mechanisms to model various aspects of the system.

This study proposes new extensions of *UML language* for the modeling of security requirements of *Web applications*, these extensions relate to the various phases of development (Specification, Logical analyze and technical structure). Firstly, we present the new vision of the computer security which makes it possible to treat the security constraints in the level of the development process, we explain after the *UMLsec* version*;* a whole of UML profiles proposed by Jürjens (Munich University of Technology) for security on the level of the conceptual models, and finally, we present the new extensions proposed and a case study of the *COMEX* system, an Information System of Commercial Management for a Harbor Company.

2. Security at the development process

Security of Information System consists in identifying the vulnerabilities, evaluating the threats and determining the risk which vulnerability allows threat given to be carried out, it uses a methods, techniques and tools to protect the resources of information system in order to ensure the availability of the services, the confidentiality and the integrity of information.

- The availability of the services: the services and information must be accessible to the authorized entity when they need some.
- Confidentiality of information: information does not belong to everyone; only can reach it those which have the right of it.
- Integrity of information: information (files, messages…) can be modified only by the authorized entity.

Adding security solutions to a system that has already been functionally realized is very difficult, and can make the system instable. The security requirements should then be *integrated at the design stage*, so that they can be identified with the first parts of development process. The posteriori security of critical systems (Firewall, Antivirus, etc.) does not constitute the best security policy. We think that the development of a security policy must be done at the same time than the functional design stage, and that the final model must integrate, at the same time, the functional and security specifications. The security of the critical systems must start with the development of a "model" which would represent: what are the threats? What do we have to protect? Why? This new approach makes the transformation of the security concept from a posteriori vision to a priori vision (at the development process level). "Security concern must inform every phase of software development, from requirements engineering to design, implementation, testing and deployment". [11] This central activity consists in foreseeing the threats and the vulnerabilities induced by the use of the system.

3. UMLsec Profiles

UMLsec is an extension of UML proposed by J.Jürjens that includes profiles for secure systems development. Stereotypes[1] are used to formulate the security requirements. The tables below show some *UMLsec* stereotypes with their labels[2].[9]

TABLE I. UMLLSEC STEREOTYPE

Stereotype	Description	Label
Secure dependency	Package to identify the secure dependency relations in the static models	
Secure links	Package to identify the secure dependency relations between the system's components	
Data security	Package to specify the critical objects and the various properties of security on the data	secrecy, integrity, high, fresh
Fair exchange	Package to represent the fair exchange scenarios in the electronic transactions	start, stop
No down – flow	Package to secure the information flow	high
Provable	Package to express non- repudiation in the electronic transactions	action, cert
Guarded access	Package to control the access to the objects	
Internet	Internet connection	
Encrypted	Encrypted connection	
LAN	Local area network connection	
Secrecy	Confidentiality of dependence	
Integrity	Integrity of dependence	
Guarded	Guarded object	guard
LAN	Local area network node	
Smart card	Smart card node	

TABLE II. UMLSEC LABELS

Label	Description
Secrecy	Data which should be secret
Integrity	Data which should not be modified
Fresh	Data which should not be re-used
Start	Initial state
Stop	Final state
Action	Provable action
Cert	Certificate activity
Guard	To guard an object

4. UML Security Extensions

The extensions which we have just proposed concern different views of the system; the *secure context model*, *security cases* and *critical scenarios* for the specification of the security requirements, the *secure interactions of objects* and the *security*

[1] The stereotypes make it possible to extend the semantics of the modeling elements and to define new UML elements classes.

[2] A label or marked value is a pair (name, value) which adds a new property to UML modeling element.

constraints on the data for the logical view and finally the *protected hardware configuration* for the technical view of the system. As an illustration, examples of the *COMEX* system will be presented.

4.1. Secure Context Model

Many authors, like G.Booch in [8] or more recently P. Roques and F.Vallee in [13], recommended the use of collaboration diagrams to represent, in a synthetic manner, the various functional requirements of a system. After the definition of security conditions, we can present the various security requirements of Web applications on a diagram, which can be called *secure context model*. This model consists in defining the various expected security services of the system considered as a black box. The collaboration diagram is used in the following way:

- The system is represented by a central object; this object is surrounded by other objects symbolizing the various actors.
- The objects are connected by bonds; on each bond are shown output messages which represent the various security services provided by the system.

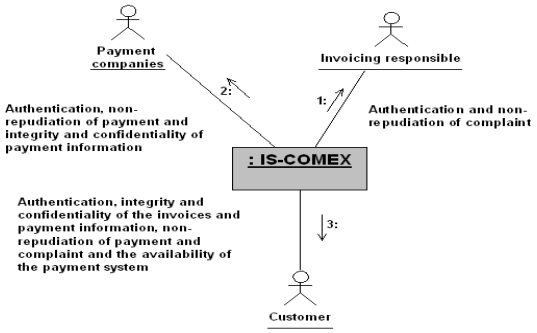

Figure 1. Example of a secure context model

4.2. Security Cases Model

In this model, we are interested in the specification of the Web applications requirements in terms of security. To do that, we use the *use cases* in a different manner by introducing the concepts of *security cases* and *security cases model*. The *security cases model* is used to structure the security services provided by the system (always considered as a black box) for the various actors as a set of *security cases*. A *security case* represents a security service returned by the system for one or more actors. For example: to verify the identity of user, to ensure the integrity and the secrecy of the exchanged information, to ensure the non-repudiation of transactions, etc. A *security case* specifies an awaited system behavior to meet security needs without imposing the realization mode of this behavior. It makes it possible to describe what the future system will have to do in terms of computer security without defining how to do it. *Security cases* are distinct from *use cases;* they do not produce a functional added value but they indeed cover all security services that a user needs.

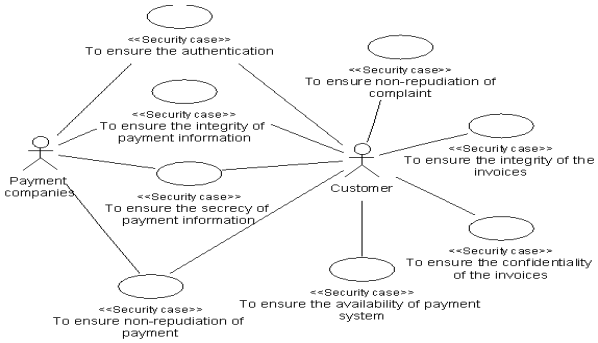

Figure 2. Example of a security cases model

4.3. Critical Scenarios

The critical scenarios consist in describing and representing the critical interactions or actions using the various services of security specified by the *security cases*. A critical scenario represents a particular succession of sequences (interactions between the actors and the system) which involves a risk in terms of computer security. To underline this risk, we will associate the various constraints of security on the interactions between the system considered as a black box and the various actors. For example: the scenarios which ensure the non-repudiation in the electronic transactions, the scenarios which specify the interactions with exchange of critical information, etc. We used the sequence diagram which makes it possible to better visualize the interactions.

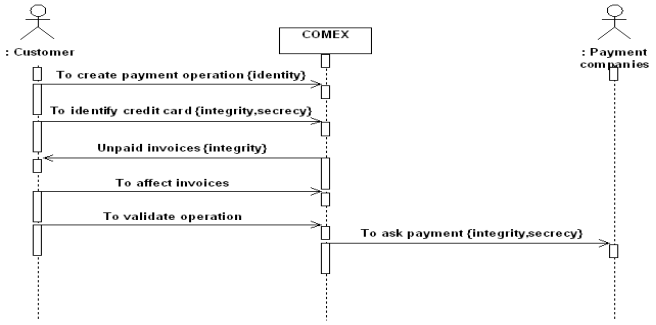

Figure 3. Example of critical scenarios model

We used three constraints[3] for the interactions between system and actors:

- The {secrecy} constraint to ensure the secrecy of the interactions.
- The {integrity} constraint to ensure the integrity of the interactions.
- The {identity} constraint to ensure the identity of the parties during the execution of interaction action between an actor and the system.

[3]A constraint is a semantic relation between UML modeling elements. Each constraint is indicated between braces and is placed close to the element (stereotyped or not).

4.4. Secure Interactions of Objects

After the identification of the classes and objects of the system (the Static Model), we now replace the system by a collaboration of objects. A scenario of secure interactions of objects represents an ordered set of messages exchanged between objects (instances of classes and actors) with the specification of the security constraints on these messages. A message represents the specification of a one-way communication between objects which transports information and whose goal is to generate a reaction from the receiver. It can include parameters which transfer values from the transmitter to the receiver. [15] For the representation of secure interactions of objects, we used the sequence and the collaboration diagrams of the UML.

- The {secrecy} constraint to ensure the secrecy of the messages;
- The {integrity} constraint to ensure the integrity of the messages;
- The {identity} constraint to ensure the identity of the transaction parties.

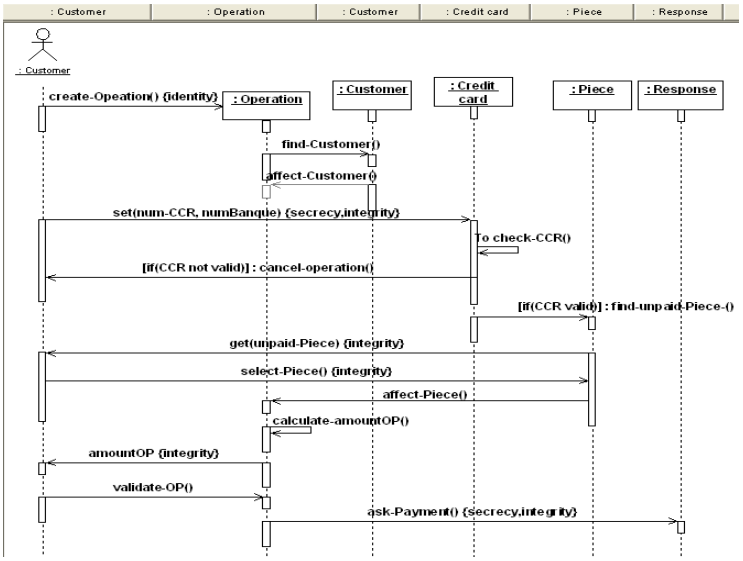

Figure 4. Example of the secure interactions of objects model

4.5. Data Security

The set of security cases discovered through the specification of security constraints guides all the dynamic views, by representing the critical scenarios, the collaborations and the interactions of objects with the sequence diagrams. In order to benefit from the security analysis phase, it is necessary to update the class diagram by adding security constraints on the data. The class diagram is viewed as the most important diagram in the object methods. After having developed the class diagram, we will define security constraints on the attributes and the operations starting from the critical scenarios represented on message flows between objects. The {secrecy} constraint specifies the data being confidential, the {integrity} constraint is used to ensure the integrity of the data and the {identity} constraint indicates that only the authorized parts can reach the data.

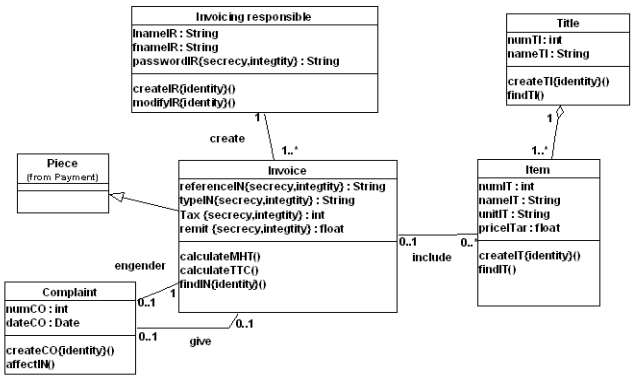

Figure 5. Example of security constraints on the data

4.6. Protected Hardware Configuration

The protected hardware configuration model consists in expressing the implementation constraints at the physical level represented by nodes and connections, which are the various types of machine connected by various means with the integration of the prevention tools (Firewall, IDS, etc) to implement the security constraints. This model also allows representing the types of connections (LAN, VPN, etc) between the various nodes. The deployment models and hardware configuration models are both expressed by using a deployment diagram. However, they do not quite express the same description level. The hardware configuration model is used to express the constraints of implementation at the physical level; it consists of the nodes and the physical connections of the system. On the other hand, the deployment model expresses the physical distribution of the system's functions and permits to justify the localization of the data bases and working environments.

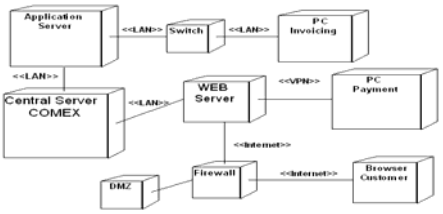

Figure 6. Example of a protected hardware configuration model

5. Conclusion

In this paper, we have tackled the highly vast subject of computer *security*, while concentrating on security of Web applications at the model level. It is a transverse approach, where the security concept is being included in the modeling of Web applications, and where the UML extensions are able to help master the control of security. The security model is a representation of security derived from a "vision of the world". The model defines what must be defended (information flow), against what (threats)

and why (sensitivity of information). It can be more or less complete, but in all cases it emphasizes just about the risks from where we can deduct a security policy. [6]

UML is not a closed notation: it is generic, extensible and configurable by the user. Where necessary, we can use extension mechanisms. This paper presents new profiles of UML for the modeling of security aspects. The *secure context model* and the *security cases model* for the specification of the security needs, the *critical scenarios model* consist in describing the interactions or the actions which involve a risk and the *secure interactions of objects model* for the specification of the security constraints on the messages exchanged by objects. In the analysis model, we defined security properties on the data. At last, for the modeling architecture, the *protected hardware configuration model* allows to express the implementation constraints at the physical level with the integration of the prevention tools in order to fulfill the security requirements. The important points which remain to be developed are: the realization of attack simulations on protected UML models in order to validate these models and to correct the security weaknesses found, and the integration of these extensions in a development process.

References

[1] M. LOULOU ALOULOU, « Approche Formelle pour la Spécification, la Vérification et le Déploiement des Politiques de Sécurité Dynamiques dans les Systèmes à base d'Agents Mobiles », Thesis of doctorate, University of Bordeaux 1, (2010).

[2] A.Abou el Kalam, « Modèles et politiques de Sécurité pour les Domaines de la Santé et des Affaires Sociales », Doctorate Thesis, Institut National Polytechnique of Toulouse ,(December 2003).

[3] Réda KADRI, « Une approche pour la Modélisation d'Applications Web à base de Composants Logiciels », Thèse de Doctorat, Université de Bretagne Sud, (2009).

[4] http://www.cert.org/stats.

[5] C.Larman, « UML et les Design Patterns », Campus Press, (2002).

[6] CNRS, "Computer security: number 31,… , 35", Site: http://www.cnrs.fr/Infosecu, (2001).

[7] E.Maiwald, « Sécurité des réseaux », Campus Press, (2001).

[8] G.Booch,"Object Solutions:Managing the Object-Oriented Project",Addison Wesley, (1996).

[9] J. Jurjens, « Secure Systems Development with UML: a Foundation », Thesis of doctorate, Munich University of Technology, (2003).

[10] N. Mayer and J Humbert. «La gestion des risques pour les systèmes d'information ». Magazine MISC n°24. ISSN: 1631-9036. (April-May 2006).

[11] P. Devanbu, «Software Engineering for Security: A Roadmap », (2000).

[12] P. Mell, K. Scarfone, S. Romanosky, « A complete guide to the Common Vulnerability Scoring System, version 2.0 », Forum of Incident Response and Security Team, (2007).

[13] P. Roques et F. Vallee, « UML en action », Eyrolles, (2002).

[14] P. Roques, « Modéliser un site e-commerce», Eyrolles, (2002).

[15] P.Roques, « UML par la Pratique », Eyrolles, second edition (2003).

[16] Robert Longeon et Jean-Luc Archimbaud , « Guide de la sécurité des systèmes d'information a l'usage des directeurs », CNRS, Site : http://www.cnrs.fr/Infosecu.

[17] S. Meng , « Security Requirements Analysis and Modeling of Distributed Systems », Thesis of Master, Munich University of Technology, (2004).

[18] BLOCH, Laurent, WOLFHUGEL, Christophe. « Sécurité informatique : principes et méthodes »,Eyrolles, (2007).