A New Pattern Template to Support the Design of Security Architectures: A Case Study

Santiago Moral-García[1], Roberto Ortiz[2], Santiago Moral-Rubio[2], Javier Garzás[1, 4] and Eduardo Fernández-Medina[3]

(1) Kybele Group. Dep. of Computer Languages and Systems II.
University Rey Juan Carlos, Madrid, Spain.
{santiago.moral, javier.garzas} @urjc.es
(2) Dep. Information Security. BBVA, Madrid, Spain.
r.ortizpl@gmail.com, santiago.moral@bbva.com
(3) GSyA Research Group. Dep. of Information Technologies and Systems.
University of Castilla-La Mancha, Ciudad Real, Spain.
eduardo.fdezmedina@uclm.es
(4) Kybele Consulting, Madrid, Spain.
javier.garzas@kybeleconsulting.com

***Abstract*—New work paradigms are emerging in the information technology sector, which are causing changes in the technological infrastructures of organizations' information systems. Organizations should adapt to all these changes in order to guarantee the confidentiality, integrity and availability of their information assets. Organizations should therefore seek support from security architectures. A good means to design security architectures is through the use of security patterns. After carrying out a systematic review of security patterns, we observed that the vast majority of current security patterns are oriented towards the production of security mechanisms, such as secure access systems or secure authentication systems. This type of patterns may be extremely useful to those security engineers who work on the production of this type of mechanisms, but they cannot be applied by a wide sector of security engineers who work in the development of security architectures. In a previous work, we proposed a new pattern template in order to complement security patterns and make them more applicable to security architecture design environments. In this paper, which is an evolution of the work mentioned above, we have validated the proposed template with a case study. This case study also provides a new security solution to ensure external accesses to organizations' production environments.**

Keywords-information security engineering; security architectures; security technologies; security patterns; real environments.

I. INTRODUCTION

The globalization of the Information Technology (IT) sector has encouraged the appearance of new paradigms in the traditional role of software development companies. This situation has been motivated by the advancement in communication systems and the need for organizations to reduce costs. These new paradigms are based on outsourcing the role of software developers in areas or regions in which labor costs are cheaper, thereby optimizing the profit margins of those organizations that hire these services.

Although the paradigms within the IT industry are changing and organizations have decided to outsource some of their services, they must ensure the confidentiality, integrity and availability of their information assets [13]. In view of the fact that the realization of this task should consider the constant evolution of the organization's setting [27], we should specifically consider the variation between people, technologies, risks, processes, volumes of information, business strategies, etc. The need therefore exists to adapt the organization to all these changes in order to guarantee the fundamental security properties for their assets [21]. It is not easy for an organization to evaluate its level of risk and adapt itself to permanent changes. It is therefore vital for it to seek support from a security architecture [3] in order to mitigate the impact of these changes and thus minimize the risks associated with each of them.

The concept of security architecture can be defined as complete, structured, coordinated and rigorous designs of information systems that support business processes in order to reduce the risk of confidentiality, integrity and availability when managing its information assets [19]. Security architectures are installed with the intention of minimizing the risks associated with the use of information technologies and optimizing an organization's business process and strategies. If this objective is to be achieved, it is necessary to establish a set of technological infrastructure controls with which to identify the security mechanisms that are needed to define the system's security.

The concept of security mechanism can be defined as artifacts designed to prevent, detect and respond to information security incidents, in order to manage and reduce the confidentiality, integrity and availability of business processes' information risks [26]. A security mechanism cannot be used in isolation to protect a business process, but a

wide set of security mechanisms can reduce security risks when managing information assets in a business process. A security architecture consists of a wide set of security mechanisms, which is complete, structured, coordinated and rigorous.

Security patterns are a good way to design security architectures because they describe a recurring problem, providing a documented and validated solution that can be used multiple times, and they combine experience and good practices in the design of information systems. After carrying out a systematic review of the literature related to security patterns, we discovered that the vast majority of current patterns are focused on supporting the construction of new security mechanisms [12, 28]. These patterns are a useful support for those engineers who work developing security mechanisms, which are the basic elements of an architecture [8, 17, 23]. However, it is difficult to apply most of them to those work environments that are focused on the analysis and design of security architecture, since they do not consider that several security mechanisms must be used to solve a security problem and they do not consider the details of installing the solution in real complex systems [15]. We understand a real complex system to be all those elements that are involved in an organization, i.e., human resources, business processes, technologies, etc.

The lacks detected in current security patterns led us to believe that it was necessary to define a new description template of security pattern with which to resolve these limitations. In order to complement current security patterns, in a previous work [16], we defined a new pattern template with which to define security patterns, characterized by the fact that it includes all the aspects that are necessary for a simple and reusable definition of security architecture designs.

In this paper, which is an evolution of the work mentioned previously [16], we have developed a case study with the aim of checking the pattern template's validity. This case study has also provided us with a security solution to ensure external accesses to organizations' production environments. Finally, the security solution obtained has been deployed in a financial organization, thus helping us to ensure the security of the organization's information assets consisting of outsourced personnel who work outside the organization's security perimeters.

The remainder of this paper is organized as follows. Section II provides a description of the goodness of the security patterns and shows related works in order to represent these patterns. Section III presents a new description template of security patterns. Section IV introduces a case study with which to validate the description template and guarantee the security of external accesses to organizations' production environments. Section V presents the lessons that we have learned after carrying out the case study. Section VI shows our general conclusions with regard to the approach, and presents our future work.

II. SECURITY PATTERNS

A security pattern describes a recurrent security problem, which arises in a specific context, and provides a well tested generic scheme as a solution to that problem [23]. One of the main advantages of patterns is that they combine experience in the design of information system [8], thus making them more efficient. Patterns are a literary format with which to capture the knowledge and experience of security experts, resulting in a structured document in the form of a template to which the security experts' knowledge is transferred [22].

The first authors to propose security patterns were Yoder and Barcalow in 1997 [29]. The number of security patterns which have been published has increased considerably since then [12, 23, 30].

A great heterogeneity exists between the different descriptions in each of the security patterns published [2, 9, 11, 22]. This is because the authors who describe the security patterns that have been discovered have historically used different description templates to represent them. The most frequently used templates are those proposed by the Gang of Four [10], which have been adapted to describe security patterns, the template proposed by Buschmann et al. [4], the template proposed in the SERENITY project [24], and that proposed by Alexander [1]. Apart from these, other templates for the description of patterns have also been published, but their use is not yet massively widespread. One example of these is that proposed in [25], in which the security patterns are represented as calculation events. Recent years have seen proposals of other types of more specific security patterns, such as attack patterns [7] or misuse patterns [9].

Although the various authors who describe security patterns do not use a standardized description template, the majority of the description templates of these patterns have the following trio of elements in common: the context in which the pattern has been discovered; the security problem that it attempts to resolve within the context put forward; and the forces that affect the solution. The solution is conditioned by the associated forces, and these are expressed through UML diagrams which model this solution [9].

In order to resolve the lacks detected in current security patterns and to thus support information security engineers when analyzing and designing organizations' security architectures, we propose a new description template of security patterns. The template proposed below is intended to be an easy-to-use guideline, which will allow both experts and non-experts in security to access a structured and methodical document with which to resolve security problems in the real complex systems of the organizations in which they work.

III. A NEW DESCRIPTION TEMPLATE OF SECURITY PATTERNS

In this section, we shall set out the new description template of security pattern, explaining its characteristics and the contribution that it will make to the scientific community in the field of security. We shall then go on to enumerate and detail

each of the description elements of the proposed template.

A security pattern focused on the development of security architectures describes a valid generic path that assists security engineers to make analysis and design decisions when confronting the development of a secure architecture, which will resolve a real security deficiency in an information system. In order to obtain the maximum applicability within an organization, the proposed solution is oriented towards the architecture and technology that must be used in that organization in order to guarantee the security of the information assets associated with the deficiencies that they intend to resolve.

The new template will be specified with the description elements from the description template proposed by Buschmann et al. [4] and the template proposed in the SERENITY project, used in [5], together with the new description elements that are necessary to provide security experts and non-experts with a template to support the design of security architectures.

One of the principal contributions of this approach is that the proposed solution provides the security engineer with three complementary levels or *viewpoints:* the platform independent level, the platform specific level and the product dependant level. This solution model manages to separate the implementation of the system's functionality specification over a platform in a specific technology. This allows us to differentiate the functionality that the system must satisfy and the technologies that could be implemented to develop the solution. The security engineer can also visualize the evolution of the solution from abstract models to real implementations in the complete system.

Figure 1 shows a graphic representation of the solution levels.

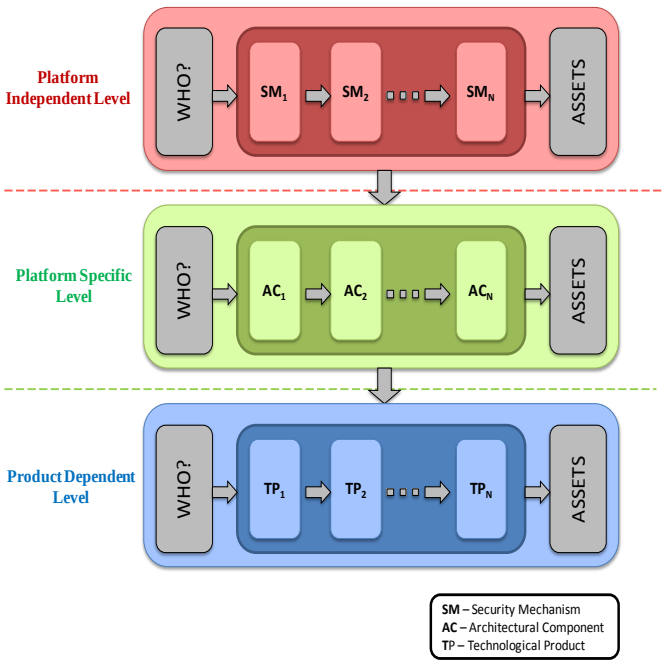

Figure 1. Abstraction levels of the solution.

As the figure above shows, all security systems must consider which information assets they intend to protect and who will have access to them.

We shall now provide a short description of each of the abstraction levels shown in Figure 1, and how the transformations needed to move from one level to the following should be carried out, illustrating which new elements should be incorporated or considered.

Platform Independent Level: this level provides a description of the security functionalities that the system should have, independently of its technological characteristics and implementation details. More specifically, a conceptual description of the security mechanisms that should be incorporated into the system is provided, along with the type of relationships that exist among them. The elements that should appear at this level are security patterns which are oriented towards the development of security mechanisms. A good guideline which can be used as a basis for discovering the type of patterns that are necessary is the guideline developed by Schumacher et al. in [23].

Platform Specific Level: the solution should be defined in this level, detailing the architecture or platform to which it will be applied. It is also necessary to set out how the necessary security mechanisms should be situated, through the presentation of an optimum security architecture with which to resolve the problem, independently of the technology used to protect the organization's systems. Given that security problems have repercussions on specific technological architectures, the same platform independent model can be instantiated N times, since it corresponds with different technological architectures. The security mechanisms described in the independent level become architectural components in this level.

Product Dependent Level: it is necessary to install the platform specific model in a specific architecture in this level, in order to implement it with technological products that are already available. Each of the architectural components can, therefore, be transformed into N technological products. The technological products must be valid products made by known manufacturers in the security industry. The final solution may vary significantly depending on the technologies used. This level should be independent of the information system's technological conditions. This view of the solution is very practical since it shows the user the different technologies that already exist on the market and that are oriented towards resolving the given problem.

This manner of structuring the solution provides a clear example of the steps that must be followed to implement the pattern, signifying that both experts and non-experts can understand the solution and know how to deploy it in a real system.

A further implicit property of this description template is its associated *decision path*. This element is of great assistance when selecting the most appropriate pattern with which to resolve a determined problem. The following five levels have

been proposed in the decision path in order to classify the patterns that are associated with a discovered security deficiency:

1) What is the state of the information, programs or configurations that need to be protected? The possible states are the following:

a) Stored: These are found in a data base.

b) Transit: Through a transfer to another company or service. There is a movement of information.

c) Accessed: The information is being accessed.

2) Who accesses the information that we wish to protect? The people who can access the information are:

a) The organization's internal users.

b) External users or customers.

c) Computing personnel during their work. This type of user is special since s/he can access data, applications and systems without using the security mechanisms which have been designed in the applications utilized by the final users.

3) How is the information accessed? or *What is the means of access?* In short, the information can be accessed in the following manners:

a) Directly: By accessing the data directly without any limitations to the use that is made of them.

b) Through an application: By applying business logic to the use, through which the information is shown.

4) Where is the information accessed from? It is basically accessed from two places:

a) From within the organization, i.e., all the technological spheres that are governed by the same security policies.

b) Outside the organization: where it is not possible to ensure the fulfillment of the same security policies that appear in the organization in which the assets are located.

5) Who manages the means used to access the information that needs to be protected?

a) The person responsible for security, who will use the pattern and will be legally authorized to manage the systems' security.

b) Any other person who does not belong to the organization or does not have legal authorization to manage the system's security.

This *decision path* can be used to verify what type of problem, in general terms, will be resolved with the pattern discovered, i.e., two security patterns that respond identically to the same path resolve problems of the same nature, and could thus be alternatives to the same problem.

With regard to the elements described in the template, it is also necessary to emphasize that they do not describe the security vulnerabilities that may affect the information system in which the solution is installed. This is owing to the fact that new vulnerabilities frequently appear and the pattern must constantly be modified. We consider that it is the technologies themselves that should be updated each time a new vulnerability is encountered, and that in this case it should be the manufacturer who updates them, or the security administrator who incorporates new rules into the security technologies used, if the impact of these vulnerabilities is to be minimized. This new template of security patterns therefore considers that vulnerabilities appear in all technologies on a permanent basis, and this concept forms a part of the pattern's considerations. The greater a technology's exposure to public networks, the higher its level of weakness. All security architectures will therefore be designed by bearing in mind that critical vulnerabilities repeatedly appear in all technologies.

The template proposed for the description of security patterns focused on the design of security architectures will be shown as follows. We must emphasize that this template is used to evolve existing security patterns, since it maintains the same base structure as their description, and it is only necessary to add the new elements that are proposed. The template that is proposed consists of the following elements:

A. Name

The pattern's name should represent the problem that it is attempting to resolve. This name must also be unique within the sphere of this type of patterns.

B. Context

The context provides a generic description of the setting, at both user level and system level, and includes the conditions in which the described pattern should be applied.

C. Problem

This describes the situation which has led to the necessity to apply a series of security mechanisms in order to obtain an optimum solution, and basically describes the reasons for the problem. It should also indicate the following questions:

- Which assets need to be protected? Information, programs and/or configurations.
- What are we protecting ourselves from? Information leaks, massive attacks, etc.
- Which security properties do we intend to conserve? Confidentiality, integrity, availability, auditability and/or non-repudiation.

D. Known incidents

A description of real cases of known security incidents, in relation to the problem posed that the implementation of the pattern intends to resolve. These incidents can be easily located on the Internet on specialized sites [6], which collect this type of events and specify when they occurred, how they occurred and what their impact was.

E. Decision Path

This element should describe all the general levels of the state of the assets that need to be protected (described previously). This will make it possible to determine which pattern should be used to resolve a specific security problem. The objective of this descriptive element is to be able to develop a methodology based on security patterns, on the basis that the definition of the pattern in itself develops its own path in the decision tree.

F. Solution

This element describes the solution in accordance with the scenario and the problem being considered. This solution must be expressed in three different abstraction levels, as has been shown previously. It is first necessary to set out the solution for a platform independent level, showing the security mechanisms that must be used and the relationships that exist among them. This first level is then transformed into a second level, called a platform specific level, which refers to the technological architecture proposed to resolve the given problem. The second level is finally transformed into a third level, called the product dependant level, which shows a proposal for the technologies that can be used to implement the solution proposed by the pattern described. The Security Engineering sector must consider these technologies to be trustworthy.

G. Considerations

It is necessary to carry out a qualitative analysis of the solution in relation to the critical parameters found in the real complex system: a) storage; b) memory consumed; c) frequency with which the systems, technologies and applications are patched up; d) process capacity; e) complexity for final user; f) complexity for security/system administrator; g) complexity of log management; h) broadband consumed; i) complexity for massive use of the solution; j) cost of installing solution; and k) solution fulfillment guarantees. It is necessary to decide whether each of these aspects is qualitatively altered in a Null (0), Low (1), Medium (2) or High (3) manner when deploying the solution in a real information system.

These decisions will assist when evaluating whether or not the implantation of the solution is appropriate with regard to the organization's current situation. This is particularly true when considering the cost parameters and fulfillment conditions, since excessive costs and an inability to ensure the fulfillment of the solution might be the principal cause of a solution being rejected.

H. Rules and Regulations

If the adoption of a predefined solution in the form of a pattern in a real environment is desired, it is necessary to consider the regulations of the country in which the solution is intended to be installed, with regard to the information activities that need to be protected. We must also bear in mind the rules associated with these regulations which must be fulfilled in the proposed solution in order for them to be correct from both a juridical and legal standpoint. For example, Argentina does not permit the movement of information related to people who reside in that country, and a solution which does not fulfill this regulation could not, therefore, be installed.

I. Benefits

A short description of a solution's goodness with regard to the sphere and specific context in which the pattern is developed.

J. Consequences

This element describes the consequences of adopting a pattern as a solution in a real information system. An analysis of the risks that the organization will run if it does not adopt this solution must also be carried out. To do this, it is necessary to describe the following consequences:

- Negative consequences of tackling the solution.
- Consequences of not tackling the solution.

K. Alternatives

The majority of security deficiencies can be resolved in different ways, and this section should therefore describe other solutions that can be used to resolve the problem considered. These alternatives may differ from the pattern described in the technological level, in the architectural level or even in the security mechanisms used to guarantee the information assets that are at risk.

IV. A CASE STUDY

In this section, we present a summary of a case study that was carried out in an organization in the banking sector with the objective of validating the template proposed in the previous section. To do this, we have followed all the elements included in the pattern template. The other objective of this case study is to help us to resolve a security lack related to the accesses of personnel who work outside the organization's security perimeter in production environments, such as external personnel dedicated to software maintenance.

Within the sphere of Information Technologies, an organization's production area is of maximum criticality owing to the fact that it is here where the information and data directly used by the customers and end users is kept. The extraction of this information or the malicious modification of the programs that access it may cause great losses in the organization. We have therefore carried out research to discover a security pattern that will resolve this lack, such that any security engineer who confronts this problem will be able to use this solution as a basis to guarantee the security of the information assets of the organization to which s/he belongs. The elements included in this security pattern are described as follows:

A. Name

Security Pattern for External Access to Productive Environments.

B. Context

The evolution of technology, and in order to reduce costs in infrastructures and installations, both on the part of an organization and on that of the suppliers in charge of maintaining the applications, signify that it is possible to locate the work carried out by these maintenance companies outside organizations' internal networks. This work is, in some cases, currently developed in the organization's installations with the objective of locating it in infrastructures belonging to the suppliers so that the maintenance work is carried out outside

the organization's installations. A supplier's design and development personnel who work for the organization therefore need to access the production environments from outside the internal security network.

C. Problem

The people who access the production information are situated in locations which are not controlled by the same person, from the point of view of security, i.e., they are not under the same security management as the systems in which the asset to be protected is located. It is also necessary to be able to download the information onto the maintenance personnel's computers to allow them to deal with it and reestablish normality in the system, signifying that these people can download information and remove it from the organization's internal network. It is therefore vital to provide any accesses that occur with security as regards the organization's assets in order to avoid undesirable situations such as misuse, illegitimate copying or any manipulation of these assets that may affect output and the organization's image.

- *Which assets need to be protected?* The principal asset to be protected is the organization's information to which the company personnel outside the organization's security limits have access.
- *What are we protecting ourselves from?* Asset leaks, i.e., leaks of the information to which the company's personnel have access.
- *Which security properties do we intend to conserve?* Confidentiality and auditability.

D. Known Incidents

Two known incidents of the theft of information or money from large organizations will be shown as follows. These thefts took place in companies that were carrying out an external service for other organizations which were the real victims of the theft.

The first incident is related to the theft of a large amount of money ($2 million) from Citibank [20]. Russian hackers accessed critical customer information from Citibank via an SQL injection vulnerability that they found on the Website of the American chain store 7-Eleven. At the time of the information theft there were 5,500 Citibank-branded ATMs at 7-Eleven. This means that for two weeks in September 2007, anyone who entered their PIN number in one of these ATMs was exposed to this fraud. As soon as the hackers had obtained duplicate bank cards and their associated PIN numbers, they began to withdraw money and to pay by stolen credit card.

The second incident is related to the theft of information from Epsilon [14], the World's Largest Permission Based Email Marketing Services Company. Epsilon sends over 40 billion emails annually and has over 2,500 clients, including 7 of the Fortune 10 to build and host their customer databases. Security Week has been able to confirm that the customer names and email addresses, and in a few cases other pieces of information, were compromised at several major companies, including the following: Kroger, TiVo, US Bank, JPMorgan Chase, Capital One, Citibank, Ameriprise Financial, Lacoste, Hilton Honors Program and Marks & Spencer, among many others.

This type of harvested data can be categorized as a minor threat, but having access to customer lists opens the opportunity for targeted phishing attacks against customers who expect communications from these companies. Attackers can use this type of data to send a targeted phishing message to a bank customer and personally address them by name. This type of attack will certainly result in a much higher success rate than a typical spamming campaign. Having access to this information will therefore simply help phishing attacks to achieve a higher success rate.

E. Decision Path

The following questions will assist in the classification of this pattern in the general context of solutions that can be found within this type of security patterns.

1) What is the state of the information, programs or configurations that needs to be protected? The state of assets is *accessed.*

2) Who accesses the information that we wish to protect? The people who access the information that we wish to protect are computing personnel during their work.

3) How is the information accessed? or *What is the means of access?* The information is accessed *directly*.

4) Where is the information accessed from? The maintenance team will access it directly from outside the organization's security perimeter.

5) Who manages the means used to access the information that needs to be protected? The security of the installations in which the maintenance work on the applications is carried out is not under the security management of the organization that requires this work.

F. Solution

The solution to the problem proposed will be set out at different abstraction levels, from the platform independent level to the product dependent level. We shall first show the platform independent level model, and shall then go on to transform this level in order to develop a model that is specific to the platform. Finally, we shall transform the previous level in order to show the solution from a product independent level.

Platform Independent Level: As is shown in Figure 2, the people in charge of the organization's software maintenance gain access via the organization's Internet in order to modify defective data and/or programs.

The security mechanisms that must be implanted in the organization's internal network to develop the desired solution are extracted from the following security patterns, which are described in greater detail in [23].

Each of the security mechanisms used to develop the solution in the platform independent level is described as follows:

- *Identification & Authentication:* Security patterns such as "I&A Requirements", "Automated I&A Design Alternatives" and "Password Design and Use" can be

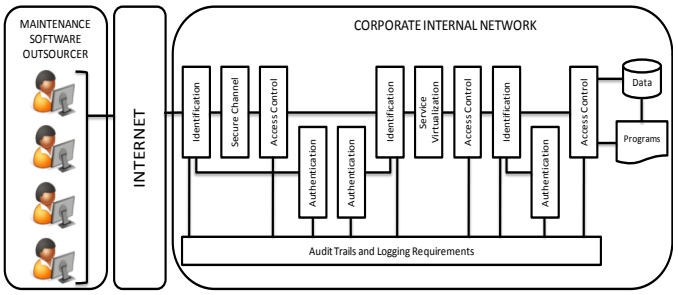

Figure 2. Platform Independent Level

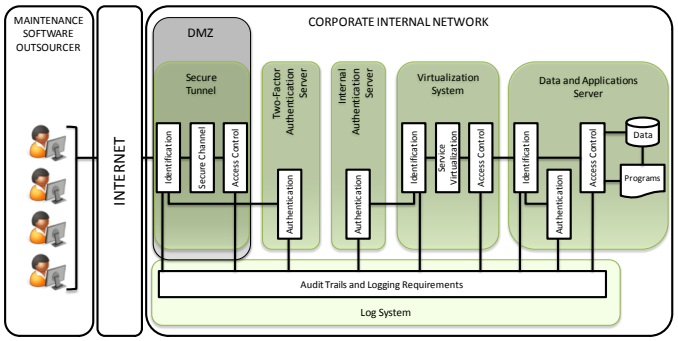

Figure 3. Platform Specific Level

used and combined for this solution. These security patterns can be used when an actor (person, process or other entity) intends to interact with an organization's system, and it must recognize that the actor is interacting with the system. Although these mechanisms appear separately in the previous figure, they are in fact complementary systems, since the whole system identification must validate the credentials via an authentication system.

- *Secure Channel:* This security mechanism is incorporated to avoid the situation of an attacker intercepting messages exchanged between the maintenance personnel and the organization on the Internet. This mechanism is able to code the channel that the information, which is in many cases sensitive, travels along.
- *Access Control:* A multitude of security patterns that expose various access control mechanisms currently exist. Those such as: "Authorization", "RBAC", "Multilevel Security" and "Role Rights Definition" can be used and combined, if considered necessary, in this solution. This type of security patterns defines security restrictions, i.e., they define the rules that permit access to the zones, resources and other aspects that strengthen the organization's security.
- *Audit Trails and Logging Requirements:* This security mechanism allows the registers that are carried out with regard to events and activities to be captured and audited, such as the identification and authorization of a resource in the organization.
- *Service Virtualization:* This security mechanism is in charge of creating a virtual version of a device or resource: a server, a storage device, a network, an operative system etc. This mechanism also permits the handling, management and provision of a computer's four main resources (CPU, memory, network and storage), thus allowing the dynamic sharing of these resources among the virtual machines defined in the central computer.

Platform Specific Level: As has already been explained in the previous section, this level is a transformation from the platform independent level. This transformation includes the architectonic components that are necessary to provide the solution. As Figure 3 shows, each of the architectonic components is composed of one or more of the security mechanisms detailed in the previous level.

Each of the architectonic components used to develop the solution in the platform specific level is described as follows:

- *Demilitarized zone (DMZ):* The intention of this security zone is to isolate the organization from potential attackers by separating the access to the various applications and services (the organization's public zone) from the different servers (the organization's private zone).
- *Secure Tunnel:* This is situated in the organization's internal network DMZ and is in charge of establishing the first filter between the Internet and the users who attempt to access it via the Internet. This security measure establishes a secure communication tunnel via the Internet in order to ensure that access to the organization's systems is exclusive to the personnel who are registered in the organization's two-factor authentication server. This tunnel consists of the following security mechanisms: (i) an identifier to allow the users to introduce their credentials. These credentials will be passed to the two-factor authentication server to verify whether they are correct. If the user has access permission then (ii) a secure tunnel (SSL) will be established from the software provider's installations which are located outside the organization's perimeter and installations. In parallel to this, (iii) the access control will manage the permissions concerning the user's resources in order to ensure that the user can only access the resources facilitated.
- *Two-Factor Authentication Server:* The external credentials of the users who wish to access the systems of the organization's internal network are checked in this server, which controls the login, the password and a two-factor to make the user access mechanism more robust.
- *Internal Authentication Server:* This authentication server will check the internal credentials which can be used to access the organization's systems. In this case they will be different credentials to those requested by the two-factor authentication server.
- *Virtualization System:* This system is in charge of creating a virtual version of the organization's systems. It must also contain (i) an identifier which is in charge of being the interface into which the users introduce their credentials. Once the credentials have been checked against the internal authentication server, (ii) a virtualized version of the system is

created which (iii) maintains an access control to establish the permissions needed with regard to the resources available.

- *Data and Applications Server:* This system is in charge of storing both the organization's data and the applications. In order to make the access to this system, which contains the resources, more robust it is necessary to install a series of security mechanisms which carry out the tasks of: (i) identification, (ii) authentication, and (iii) access control, as in the other previously explained areas.
- *Log System:* This architectonic element is in charge of collecting all the activities that are relevant to identification, control and access control of the various mechanisms. The information is gathered in the form of a log.

Product Dependent Level: As is shown in Figure 4, in this level the architectonic elements detailed in the previous level are transformed into specific technological products of concrete manufacturers. The technological products to be installed in the organization's internal network must be products which have been validated in the company's security environment.

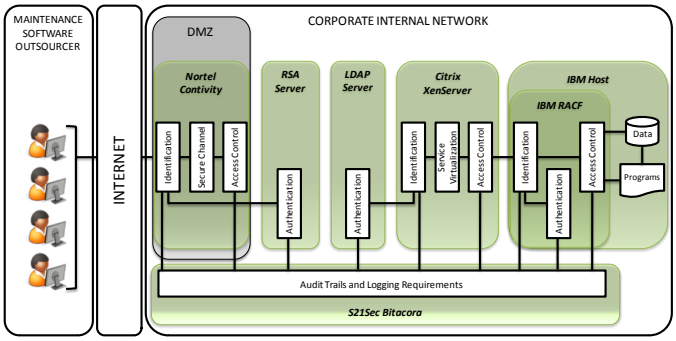

Figure 4. Product Dependent Level

As is shown in the previous figure, the specific technological products correspond with one or more of the architectonic components defined in the platform independent level. Table 1 shows the correspondence between the architectonic components and the specific technological products of this solution:

TABLE 1. TECHNOLOGICAL PRODUCTS

Architectonic Components	Technological Products
Security Tunnel	Nortel Contivity
Two- Factor Authentication Server	RSA Server
Internal Authentication Server	LDAP Server
Virtualization System	Citrix XenServer
Data and Applications Server	IBM Host
	IBM RACF
Log System	S21Sec Bitacora

G. Considerations

It must be possible to answer whether each of the technological aspects related to the system in which the solution is integrated are qualitatively altered in any of the following manners: null (0), low (1), medium (2), or high (3). The first column in Table 2 shows the considerations that must be taken into account, while the second column provides a brief description of the means used to analyze each consideration. Finally, the third column shows the results of the qualitative analysis of this solution.

TABLE 2. CONSIDERATIONS

Aspects to consider	Description	Analysis
Storage	It is necessary to identify specific causes resulting from an elevated consumption of storage in comparison to other existing solutions. It is principally necessary to estimate the economic impact that the adoption of this solution will have on the organization.	1
Memory Consumed	It is necessary to identify specific causes resulting from high memory consumption in comparison with other existing solutions.	3
Frequency of Patching	It is necessary to estimate specific causes resulting from a rise in the frequency of patching up the solution proposed. This evaluation must be carried out in both the economic plan and the risk plan associated with vulnerabilities.	0
Process Consumption	It is necessary to identify specific causes resulting from the elevated consumption of the process capacity in comparison with other existing solutions.	3
Broadband	It is necessary to estimate the technological aspects needed to identify specific causes resulting from a high consumption of broadband in comparison to other existing solutions.	2
Installation Cost	It is necessary to evaluate the global cost of the installation of the solution in the organization's real environment in comparison with other existing solutions.	3

Complexity for Security Administrator	It is necessary to evaluate whether the installation of the solution requires an increase in the time needed by the person in charge of adapting it and maintaining in the organization's installations.	0
Complexity of Log Management	It is necessary to evaluate whether the installation of the solution requires an increase in the time needed by the personnel who manage and research the logs collected.	2
Complexity of Use for End User	It is necessary to evaluate whether the installation of the solution requires an increase in the time needed by the end user when using the systems in which the solution has been applied.	0
Complexity of Massive Expansion	It is necessary to evaluate whether the installation of the solution requires an increase in the time needed by those in charge of the massive installation of the solution in various points of the organization.	0
Complexity for System Administrator	It is necessary to evaluate whether the installation of the solution requires an increase in the time needed by the administrators of the other systems, and the impact on their daily work.	0
Residual Risk	It is necessary to evaluate whether the pattern, once it has been installed and is functioning correctly, needs complementary measures to attain its initial objective.	0
Capacity to Ensure Fulfillment	It is necessary to evaluate whether it is possible to display the necessary measures which allow us to verify the correct functioning of the pattern, and whether all the participants will be sufficiently motivated to fulfill it.	3

H. Rules and Regulations:

In order to adopt the proposed pattern it is necessary to bear certain considerations in mind with regard to the location of the organization, the location of the software maintenance factories and the flow of information between the organization and the factories. It is therefore necessary to bear the following considerations in mind:

- Privacy laws are different in the different countries involved in adopting the solution.
- Make an inventory of the data that will be part of the normal flow of the solution in order to adopt the necessary measures relative to each country's treatment of personal data.
- The local restrictions of each country with regard to the treatment, quality and robustness of the passwords used, and the minimum security measures of the organizations residing in each country.
- Consult those responsible for security in the various headquarters of the organization so that they can evaluate the risks involved in not complying with the laws associated with the country in which they are.

I. Benefits:

If the pattern shown in this paper is adopted to provide security in productive environments outside the perimeter of the organization, then the following benefits will be obtained:

- The simplification of all externalization processes, in which externalization signifies the location of personnel outside the organization's perimeter, since the technological complexity associated with these processes is reduced.
- The architecture is reusable in similar situations such as the remote administration of systems, remote access to different environments to carry out tele-maintenance tasks, etc.
- The structural solution for an organization, since it is a robust solution that will last.
- Great savings in the organization's infrastructures, since the maintenance work on the applications is now carried out in the suppliers' installations rather than in those of the organization.

It can be deduced that this system is valid and has good behavior whenever it is necessary to make the security conditions of the system used independent in order to carry out different maintenance tasks on the systems in which these tasks must take place.

J. Consequences

The negative consequences of adopting this pattern as a solution, and the risks that the organization may run if it does not adopt this solution are the following:

- ***Negative consequences of tackling the solution:*** The adoption of this solution requires a great investment in the virtualization system, since the entire process moves from being carried out on the organization's computers to being carried out in a virtualized system. A high economic investment is also necessary to resize the infrastructures of the organization's systems in order to adapt them to the technological aspects needed to tackle the solution.
- ***Consequences of not tackling the solution:*** Organizations will run the risk of suffering situations of misuse, copying, illicit distribution and theft of the data to which the maintenance factory's personnel has access, thus making an impact on both the organization's image and exposing it to potential attackers.

K. Alternatives:

The different security alternatives with which to solve this aforementioned problem are as follows:

- ***Alternative 1:*** The first security alternative to the situation of the externalization of maintenance tasks is to centralize the security management in the same circumstances, conditions and restrictions as those used in the organization. This occurs by passing the security control of the factory's installations over to those responsible for security in the organization that requires the maintenance work to be done. It would thus be possible to ensure that the tasks carried out would be controlled exactly as the organization that had contracted the service wished. The fulfillment of this alternative is relatively low, since each supplier has its own security department and would find it difficult to adopt this measure.
- ***Alternative 2:*** Commitment on the part of the company supplying the service to adopt the security measures required by the organization that contracts them. These security measures consist of dedicating installations exclusively to the realization of maintenance work, the dedication of lines of communication, the adoption of the same security measures as the organization with regard to personnel, etc. This measure requires the signing of contracts containing all the aspects mentioned, confidentiality clauses, the periodical auditing of the company providing the service, etc.

V. LESSONS LEARNED

Our real-life experience of tackling an externalization project for the software suppliers in an organization in the banking sector is summarized as follows. We show the advantages, disadvantages and the lessons learned.

Before formalizing a solution using the previously explained pattern template, we carried out a security analysis, which is our usual course of action whenever we are confronted with a new project that affects the systems of the organization in which we work. The principal tasks carried out in this analysis can be seen in [18].

The initial requirement is clear: we must reduce costs in the organization, and one of the main reductions is achieved by externalizing the personnel that carry out the software development and maintenance tasks. To do this, it was necessary to design a solution so that the work of these groups of people was not affected, as far as possible, and so that they would continue working outside the organization's installations in as similar a way as possible.

One of the organization's main handicaps was that the information that these development factories accessed, when they were still at the organization's headquarters, was sensitive, and could have affected the business if it had leaked from the installations. In this new work model, they therefore had to access this information in a regular manner in order to continue functioning as normal.

After designing the solution, we decided to express it as a security pattern, since this type of situations is very common and recurrent in organizations, either for cost reduction or because the software developers cannot constantly displace themselves every time an incident occurs in the systems of the organization that they serve.

After analyzing and implementing the solution presented in the case study shown in the previous section, the principal advantages were the following:

In the first place, we attained the objective pursued, i.e., the reduction of costs. This objective was attained because in addition to obtaining a more accessible workforce, we also reduced investment in the organization's installations, such as electricity, gas, jobs, computational material, etc. With regard to what affects the information systems, and particularly the security environment of the information, one of the most notable advantages which had not initially been contemplated was that we obtained order and coherence in the access to productive environments on the part of the software development or maintenance teams when they were consulting, modifying or eliminating information that was, in some cases, critical. After deciding to externalize the software factories, the analysis concentrated on protecting the assets that would be accessed from outside the organization. This was done by designing the communications between the headquarters of the software factories and the organization's information systems. This allowed us to, on the one hand eliminate all the access routes that had previously existed in the organization and, on the other, to correctly censor all the accesses to the productive environments, analyzing each of the casuistries that the software developers requested to carry out their maintenance tasks. We thus managed to eliminate undesired accesses that might have been occurring. This resulted in a) a greater control of the actions carried out by the externalized software maintenance teams, without organization's information systems, b) a refinement of the entity's technological systems, c) a considerable increase in the security of the information systems that contained data that was critical to the organization, and d) we obtained an exhaustive census of all the accesses that were produced, from the software factories to the information systems, so that if necessary we could carry out a forensic analysis of hypothetical information leaks or violations of the service conditions on the part of the company providing the service.

On the other hand, the disadvantages discovered concerned incidents provoked, because in some cases, when the personnel in charge of the maintenance or installation of the software developed arrived at their new location they could not access the information systems with the same privileges as they had had when they were in the organization's systems, and this limited their actions when working at remote form. These work teams have therefore had to adjust their customs to the new form of work designed to carry out their function. One example of this type of cases is the following: given that the system which is prepared to access the production environment

is virtualized and watertight, i.e., no information can be moved from that system to others, one of the problems was that it was impossible to print out or download any type of document, because if this were permitted it might lead to uncontrolled information leaks. This was an habitual practice for the software maintenance teams, since when cataloging and resolving incidents they download the information to their computers to then substitute the modified code without having to always be connected to the organization's network. This situation obliged each of the developers who resolved and cataloged incidents in the organization's productive environments to always have an Internet connection at their disposal.

Leaving aside the advantages and disadvantages discovered after implementing the previously proposed solution, we consider that the structure of this solution in the form of a pattern will help to optimize the time effort and cost needed to analyze this type of problems, because it reduces the majority of similar cases to one specific case, which is the access to critical data from outside the installations of the organizations that own them.

With regard to the pattern template proposed, the analysis of this type of real cases has provided us with a huge amount of feedback with which to refine the proposal. We therefore consider the section in which the information assets that must be protected are cataloged to be primordial. This is owing to the fact that if there is an exhaustive cataloging of the assets that are accessed, in addition to obtaining the locations from which they can be accessed, the security measures to be applied are very specific, i.e., depending on the criticality of the data to be protected and from where they are accessed, the mechanisms used to guarantee the security of the information assets can either be very relaxed or very robust.

VI. Conclusion and Future Work

In this paper, we have presented a security pattern description template which had previously been published, and which has now been completed with a practical case, which validates its use in a specific problem in a real and complex organization.

After carrying out a systematic review of the state-of-the art of those works that present security patterns, we detected a series of lacks, which we have attempted to solve with this new template. Of the most outstanding lacks we can highlight: the proposals analyzed are oriented towards the construction of security mechanisms, and not towards the construction of security architectures in information systems; they do not contemplate the impact that the implementation of the security pattern will make on an information system; they do not carry out a classification on the basis of the criticality of the information assets to be protected, and do not therefore specify the appropriate security measures to be applied; and they do not take into consideration the regulations or laws that may apply to the solution in the sector or country in which the organization operates.

All these lacks have motivated our research, which has led us to design a template for the description of security patterns oriented towards the construction of secure architectures, which collects each of the lacks detected in the aforementioned works.

In this work, we have not only presented the new template with which to describe security patterns, but we have also shown a case study extracted from a real and recurrent problem faced by technological organizations, which is access to productive environments from outside the organization's security perimeter. With this proposal we intend to validate the previously defined template, with a real case in a complex organization.

This security problem is very common in any large organization and requires an exhaustive analysis to be mitigate as much as possible the leaking of the organization's information assets. We have therefore decided to describe this problem with a solution in the form of a pattern, which will thus serve to assist information security engineers to resolve problems of this type in an agile and effective manner, whilst maintaining the homogeneity in each of the systems in which it is implemented.

This exercise of adapting a real problem to the form of the security pattern template proposed has helped us to validate and refine the template on which we are working. For example, we have realized the importance of the template section in which the security assets to be protected on the basis of their criticality are classified. This is therefore one of the fundamental parts to which most attention is paid when designing security architectures in the form of a security pattern.

We are currently working on the implementation of new practical cases following the template, which will allow us to refine and validate it. We are also working on the modeling of a framework based on security pattern mining, whose principal objective is to discover, design and document security patterns that concentrate on supporting the design of security architectures.

Another of the lines on which we are working is the definition of a secure information system development methodology based on security patterns, which will guide information security engineers when systematically and homogonously resolving security problems in real complex organizations.

Acknowledgements

This research has been carried out in the framework of the following projects: MODEL-CAOS (TIN2008-03582/TIN) financed by the Spanish Ministry of Education and Science, BUSINESS (PET2008-0136) financed by the Ministry of Science and Innovation, and SISTEMAS (PII2I09-0150-3135) and SERENIDAD (PEII11-0327-7035), all financed by the Local Government of Castilla-La Mancha, in Spain.

REFERENCES

[1] C. Alexander, S. Ishikawa, and M. Silverstein "A Pattern Language: Towns, Buildings, Constructions" Oxford University Press, 1977.

[2] Z. Anwar, W. Yurcik, R. E. Johnson, M. Hafiz, and R. H. Campbell "Multiple design patterns for voice over IP (VoIP) security" in Performance, Computing, and Communications Conference (IPCCC 2006). 25th IEEE International, 2006.

[3] A. Barth, C. Jackson, and C. Reis "The Security Architecture of the Chromium Browser", Technical Report 2008.

[4] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal. "Pattern-oriented software architecture: A system of patterns" Wiley, 1996.

[5] A. Cuevas, P. El Khoury, L. Gomez, and A. Laube "Security Patterns for Capturing Encryption-Based Access Control to Sensor Data" in SECURWARE '08. Second International Conference on Emerging Security Information, Systems and Technologies, 2008, pp. 62-67.

[6] "DATALOSS db - Open Security Fundation", http://datalossdb.org/, retrieved: January, 2012.

[7] E. Fernandez, J. Pelaez, and M. Larrondo-Petrie "Attack Patterns: A New Forensic and Design Tool" in Advances in Digital Forensics III, 2007, pp. 345-357.

[8] E. B. Fernández "Security patterns and secure systems design" ACM Southeast Regional Conference 2007.

[9] E. B. Fernandez, N. Yoshioka, and H. Washizaki "Modeling Misuse Patterns" in ARES '09. International Conference on Availability, Reliability and Security, 2009, pp. 566-571.

[10] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides "Design Patterns: Elements of Reusable Object Oriented Software" Addison Wesley, 1995.

[11] J. Garzás and M. Piattini "Object Oriented Microarchitectural Design Knowledge" IEEE Software, pp. 28-33, 2005.

[12] M. Hafiz, P. Adamczyk, and R. E. Johnson "Organizing Security Patterns" Software, IEEE, pp. 52-60, 2007.

[13] D. M. Kienzle, M. C. Elder, D. Tyree, and J. Edwards-Hewitt "Security patterns repository, version 1.0" 2006.

[14] M. Lennon, "Massive Breach at Epsilon Compromises Customer Lists of Major Brands", http://www.securityweek.com/massive-breach-epsilon-compromises-customer-lists-major-brands, retrieved: January, 2012.

[15] S. Moral-García, S. Moral-Rubio, and E. Fernández-Medina "Security Pattern Mining: Systematic Review and Proposal" in WOSIS '11. 8th International Workshop on Security in Information Systems, 2011, pp. 13-24.

[16] S. Moral-García, R. Ortiz, S. Moral-Rubio, B. Vela, J. Garzás, and E. Fernández-Medina "A new Pattern Template to Support the Design of Security Architectures" in PATTERNS 2010. 2nd International Conference on Pervasive Patterns and Applications, 2010, pp. 66-71.

[17] T. Okubo and H. Tanaka "Web security patterns for analysis and design" in Proceedings of the 15th Conference on Pattern Languages of Programs, Nashville, Tennessee, 2008.

[18] R. Ortiz, S. Moral-Rubio, J. Garzás, and E. Fernández-Medina "Towards a Pattern-Based Security Methodology to Build Secure Information Systems" in WOSIS '11. 8th International Workshop on Security in Information Systems 2011, pp. 59-69.

[19] OSA, "Open Security Architecture", http://www.opensecurityarchitecture.org/cms/index.php, retrieved: January, 2012.

[20] K. Poulsen, "7-Eleven Hack From Russia Led to ATM Looting in New York", http://www.wired.com/threatlevel/2009/12/seven-eleven/, retrieved: January, 2012.

[21] D. G. Rosado, C. Gutiérrez, E. Fernández-Medina, and M. Piattini "Security patterns and requirements for internet-based applications" Internet Research: Electronic Networking Applications and Policy, pp. 519-536, 2006.

[22] M. Schumacher "B. Example Security Patterns and Annotations" in Security Engineering with Patterns, 2003, pp. 171-178.

[23] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad "Security Patterns: Integrating Security and Systems Engineering" Wiley, 2006.

[24] "Serenity Project - System Engineering for Security & Dependability", www.serenity-project.org, retrieved: January, 2012.

[25] G. Spanoudakis, C. Kloukinas, and K. Androutsopoulos "Towards security monitoring patterns" in Proceedings of the 2007 ACM symposium on Applied computing, Seoul, Korea, 2007.

[26] W. Stallings "Network security essentials: applications and standards", Prentice Hall, 2007.

[27] C. Steel, R. Nagappan, and R. Lai "Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management", Prentice Hall ed., 2005.

[28] H. Washizaki, E. B. Fernandez, K. Maruyama, A. Kubo, and N. Yoshioka "Improving the Classification of Security Patterns" in DEXA '09. 20th International Workshop on Database and Expert Systems Application, 2009, pp. 165-170.

[29] J. Yoder and J. Barcalow "Architectural Patterns for Enabling Application Security" in Fourth Conference on Patterns Languages of Programs (PLoP'97), 1997.

[30] K. Yskout, T. Heyman, R. Scandariato, and W. Joosen " An inventory of security patterns" Katholieke Universiteit Leuven, Department of Computer Science, 2006.