SECURITY AND COMMUNICATION NETWORKS
*Security Comm. Networks* 2014; **7**:1670–1690
Published online 6 March 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.863

RESEARCH ARTICLE

# Enterprise security pattern: a new type of security pattern

Santiago Moral-García[1,2]*, Santiago Moral-Rubio[3], David G. Rosado[4], Eduardo B. Fernández[5] and Eduardo Fernández-Medina[4]

[1] IT Risk Department, Produban, Boston, MA, U.S.A.
[2] Kybele Research Group, Department of Computer Languages and Systems II, Rey Juan Carlos University, Madrid, Spain
[3] Chief Information Security Officer, BBVA Group, Madrid, Spain
[4] GSyA Research Group, Department of Information Technologies and Systems, University of Castilla-La Mancha, Ciudad Real, Spain
[5] Secure Systems Research Group, Department of Computer & Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, U.S.A.

## ABSTRACT

In recent years, most organizations have suffered attacks against their information systems. For this reason, organizations should seek support from enterprise security architectures (ESAs) in order to secure their information assets. Security patterns can help when building complex ESAs, but they have some limitations that reduce their usability. In this paper, we define the metapattern of a new type of security pattern called Enterprise Security Pattern. This new metapattern provides a model-driven environment and combines all elements that must be considered when designing and building ESAs. We present here a precise meta-model and four diagrams to describe the metapattern of the enterprise security patterns. When avoiding a security problem, organizations could use enterprise security patterns to provide their designers with an optimal and proven security guideline and so standardize the design and building of the ESA for that problem. Enterprise security patterns could also facilitate the selection and tailoring of security policies, patterns, mechanisms, and technologies when a designer is building ESAs. To illustrate our ideas, we present an instance of this new type of pattern, showing how it can be used. Copyright © 2014 John Wiley & Sons, Ltd.

**KEYWORDS**

secure information system; enterprise security architecture; security pattern; enterprise security pattern; threat modeling

***Correspondence**
Santiago Moral-García, IT Risk Department, Produban, Boston, MA, U.S.A.
E-mail: smoralga@produban.us

## 1. INTRODUCTION

In recent years, the vast majority of organizations, regardless of their geographic location or industry, have suffered intentional attacks against their information systems [1]. Most of these attacks are carried out by organized e-crime groups, whose main objective, in most cases, is to obtain or modify sensitive data from organizations [2]. E-crime groups do not choose at random data to attack. The main feature of these data is that they can be monetized, that is, they can produce an economic benefit for the attacker, leading to an economic loss for the organization attacked [3].

For this reason, in recent years, the main objective of organizations, in terms of security, is to ensure the continuity of business operations and to protect the security properties of their information assets (confidentiality, integrity, availability, and auditability). With these purposes in mind, organizations should perform the following: (i) seek support from enterprise security architectures (ESAs); and (ii) use some security methodology.

The objective of ESA is to provide the conceptual design of the system security infrastructure, related security mechanisms, and related security policies and procedures [4]. This conceptual design links the components of security infrastructure as one cohesive unit in order to protect corporate information. To do this, the ESA should determine what information assets must be protected, from what types of attacks, and who (people) or what (system) has access to them.

Because of the fundamental value of information assets to enterprises, a systematic approach is required to build secure systems [5–8]. In our experience, methodologies based on patterns can provide this systematic approach. The use of these kinds of methodologies based on patterns

when deploying new information systems or evolving existing systems organizations is recommended because it leads to a strong ESA.

There are many recurrent problems in software, and patterns try to capture their solution [9]. Within the scope of patterns, we may find several catalogs. Security patterns join the extensive knowledge accumulated about security with the structure provided by patterns. These patterns provide the guidelines to support the construction and evaluation of security mechanisms [10]. The use of security patterns helps to incorporate security principles when building secure systems. However, they have some limitations:

- They are small units of defense [11]. They can only handle one (or a few) threats. Considering the number of threats that current information systems have, a security designer should apply an extensive set of security patterns when building secure systems.
- There are different versions of the same pattern for each architectural level. For example, we may find an abstract access control pattern, an access control pattern for distributed systems, and an access control pattern for web services. As the building of secure systems needs an extensive set of security patterns, this fact increases the complexity when a security designer is trying to select a pattern.
- Several instantiations of a pattern may have common aspects, but the designer has to find them. For example, within a design, the designer may need an access control pattern to restrict the access to the customers' account and another access control pattern to restrict the access to the customers' stocks. In this case, the designer is using two instantiations of the same pattern, but it is possible that some aspects of the instantiations are common. Unnecessary redundancies may result.

Because of these limitations, in a previous work [12], we defined a new type of pattern to support the design of ESAs. In this paper, we have refined that approach to define a new type of security pattern called the Enterprise Security Pattern (this expression was used in [13], where the authors use this expression to describe and to identify a set of existing security patterns focused on a specific environment such as enterprise environments). We adopt this name, because the objective of these patterns is to provide a top-down strategy based on models for defining ESAs in different levels of abstraction, including their technological implementation. These patterns are not intended to replace security patterns. They use and incorporate them in a more comprehensive pattern that can handle more threats. An enterprise security pattern combines a wide range of items describing generic ESAs that protect a set of information assets in a specific context. We describe a precise meta-model complemented with specific diagrams to represent their solution. We also show a detailed example related to the external access to a production environment. In another work [14], we also provide an example of enterprise security pattern related to secure Software as a service.

When avoiding a security problem, organizations could use enterprise security patterns in order to select a global security strategy, providing their designers with an optimal and proven set of security guidelines, and so standardize the design and building of the ESA for that problem. Using enterprise security patterns security engineers could, on the one hand, manage separately the security elements included in the different abstraction models and, on the other hand, perform automatic transformations between them. This fact would facilitate the designer in the selection and tailoring of security policies, patterns, mechanisms, and technologies when they are building ESAs.

As the use of a template to define patterns is well known and widespread [12,15–18], we define a specific template to define and to document enterprise security patterns, which is associated with the ESA elements and the meta-model described in this work. The template presented here is the main focus for defining enterprise security patterns, and therefore, the meta-model is defined according to this template, and the sections are defined with the aim of incorporating the ESAs elements, such as assets, context, threats, security technologies, and stakeholders.

The remainder of this paper is organized as follows. Section 2 provides a brief description of ESAs and the elements included in them. Section 3 defines the template used to document enterprise security patterns and the relationship with the ESA elements. Section 4 provides a precise meta-model of these patterns. Section 5 presents a new set of diagrams to graphically represent their context and solution. Section 6 shows an example of an enterprise security pattern. Section 7 discusses related work. Finally, Section 8 presents some conclusions and future work.

## 2. ENTERPRISE SECURITY ARCHITECTURES

A subset of software architectures are the architectures that provide security for enterprise information systems. From that, the concept of "enterprise security architecture" was born. As with the conventional architecture of buildings, the enterprise security architect must take into account the following [19]:

- The *goal* that one wants to achieve with the architecture.
- The *environment* in which the architecture will be built and used.
- The *threats* that the architecture may suffer in this environment.
- The *technical capabilities* needed to construct and operate the architecture.
- *Who* (or *what*) will build, use, or maintain the architecture?

Figure 1 shows ESAs elements associated with each of the considerations that the security architect must take into account when building the architecture. We discuss in the preceding texts each of the elements included in ESAs.

### 2.1. Information assets

Information assets can be defined as information items stored in the systems of the organization, which are recognized as valuable. When we refer to systems of the organization, we are also including employees' computers, laptops, and smart phones. Depending on the size or the industry of the organization, this list of assets may vary significantly. If the number of information assets is very high, the complexity and cost of protecting them may increase.

### 2.2. Context

All security architectures start with defining the business context that being the balance of business drivers and acceptable risk. This business context is the result of decisions made from the analysis of internal and external factors. Security policies are the guidelines for this business context. The resulting architecture is a functional combination of process and technology to achieve the business goal within the boundaries of the business context. The architecture must fit this business context for the enterprise to achieve security and to provide legal and regulatory compliance [20].

Policies are management directives indicating a predetermined course of action, or a way to handle a problem or situation [21]. Without policies, it is impossible to build secure systems; we would not know what we should protect or how much effort we should put into this protection [22]. A specific system uses a combination of security policies according to its goals and environment. When building secure systems, designers have to consider many security policies of different types, such as confidentiality policies, integrity policies, and availability policies.

The context or environment for ESAs is composed of security realms (SRs) and is associated with sensitivity records, which provide a set of security policies. These sub-elements are explained in the next section.

### 2.3. Threats

When protecting a specific information asset, a security designer should consider the methods or ways that e-crime groups and insiders use to breach the security defenses. Depending on the information asset's sensitivity for the organization and the security properties that the threats associated with this asset would violate, it may or may not be required that the ESA needs to protect the asset. For this reason, when building secure systems, organizations should consider all threats associated to the information assets to be protected.

### 2.4. Stakeholders and systems

A stakeholder can be defined as an individual, a team, or an organization with an interest in, or concerns relative to, a system [23]. When building secure systems, the designer should take into account the interests and concerns of the stakeholders that will build, use, and maintain the security technologies involved in the ESA. Considering that stakeholders often have different backgrounds, the designer should be able to explain the advantages and disadvantages of the design decisions. Depending on the methodology used in the building of the system, the stakeholders could change. But, in most cases, the designer should take into account the interests and concerns of the following stakeholders: *systems administrator*, *security administrator*, *log administrator*, *security developer's team*, *technical users*, and *end users*.

In addition, the designer should also take into account the systems and applications that will interact and use the security technologies to be deployed. The goal of these considerations is to avoid coupling problems between the systems/applications and security technologies in the later stages of the development lifecycle (implementation or testing stages).

### 2.5. Security technologies

These are technologies that assist in the protection or mitigation of the effects of attack methods used against information assets, the assessment of the damage provoked

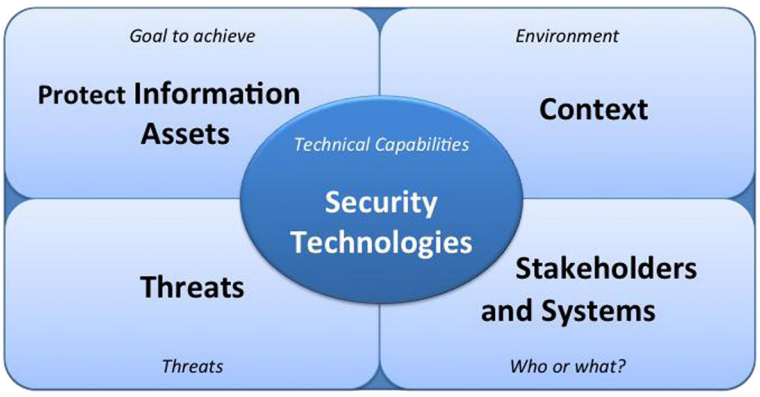


**Figure 1.** Enterprise security architecture elements.


DOI: 10.1002/sec

by these attacks, or the response management for them. One of the features of information systems is that they could operate without the use of security technologies. Security technologies are the tools to provide the desired level of security; they do not add to the system's functionality.

## 3. TEMPLATE FOR ENTERPRISE SECURITY PATTERNS RELATED TO ENTERPRISE SECURITY ARCHITECTURE ELEMENTS

We have defined, in the previous section, a set of elements to take into account when we build ESAs. We want to define a security pattern to help us to incorporate these security elements when building secure systems for an enterprise environment. This pattern will be defined by using a template whose elements will be related to elements of the ESA. In this way, when we are using enterprise security patterns in a security methodology, we are incorporating the security elements of an ESA within the development process.

The elements of the pattern correspond to sections of the template used to document it. This template includes sections of the template provided by Buschmann *et al*. [24] and some new sections that we consider necessary when designing ESAs, such as *Intent*, *known incidents*, and *considerations*. All the sections are described in the following texts:

- ]Name: The pattern's name should represent the problem that it is attempting to solve. This name must also be unique within the scope of this type of pattern.
- Intent: This section provides a short description of the intended purpose of the pattern.
- Context: This section describes the generic environment under which the enterprise security pattern should be applied. The context may include the following: (i) the type of information assets to protect (data, applications, and code and configuration); (ii) the SRs where the assets are stored; and (iii) the general features of who (customers, employees, or technical users) or what (systems) will access the assets. The context should be specified by using the context diagram shown in the Section 6.
- Problem: This section describes the situation that has led to the necessity to apply a series of security mechanisms, including the threats that cause the situation and the forces that guide the solution. The problem section should also consider the information assets, because they will affect the security mechanisms of the solution.
- Known incidents: This section describes real cases of known security incidents related to the problem. These incidents can be found in specialized websites, such as [25], which collect this type of events and specify when they occurred, how they occurred, and what their impact was.
- Solution: This section describes how the ESA could handle the threats associated with the information assets to protect. The solution must be expressed in the four different models: Computation Independent Model (CIM), Platform Independent Model (PIM), Platform Specific Model (PSM), and Product Dependent Model (PDM). Each of these models should be specified by using the diagram model associated with it (Section 5).
- Considerations: This section presents a qualitative analysis of the PDM of the solution in relation to its performance overhead (storage, primary memory, processor, and bandwidth), installation cost, complexity of massive expansion, and the complexity for the stakeholders (security administrator, log administrator, end user, and system administrator) who will build, use, and maintain the technologies involved in the solution. This analysis should also show if the ESA would need complementary measures to attain its objective, that is, the residual risk of the solution.
- Consequences: This section discusses the benefits and drawbacks of the solution in relation to the forces found in the problem. The consequences should also discuss what threats are prevented or not when implementing the solution in a real system. The enumeration of consequences should match the forces of the problem, but there may be consequences that do not correspond to any force.
- Known uses: This section describes existing ESAs where the solution provided in the PSM of the pattern has been used. For solutions where the pattern has not yet been deployed, specific contexts where the pattern could be deployed are enough.
- Related patterns: This section gives references to enterprise security patterns that solve similar problems, consider similar contexts, or complement this pattern.

Figure 2 presents a Unified Modeling Language (UML) meta-model that defines both the elements of an enterprise security pattern (white rectangle with *) and the elements of an ESA (shaded rectangle), as well as the relationships between them. As we can see in Figure 2, all the elements of our pattern are related to some element of the ESA, so that we have defined the template of the security pattern to include security principles, aspects, considerations, and issues of the ESAs. For example, the "solution" element of the template is associated with the "context" and "security technologies" elements of the ESA indicating that as a solution to protect the information assets for a similar context, we need specific security technologies that provide us with the security mechanisms necessary to protect these assets.

## 4. A META-MODEL FOR ENTERPRISE SECURITY PATTERNS

An enterprise security pattern combines a wide range of items describing generic ESAs that provide some security properties for a set of information assets in a specific context. To do this, enterprise security patterns combine in one cohesive pattern, and using a template (defined in the previous section), all elements included in the ESAs are as follows: (i) the information assets to be protected; (ii) the context in which these assets

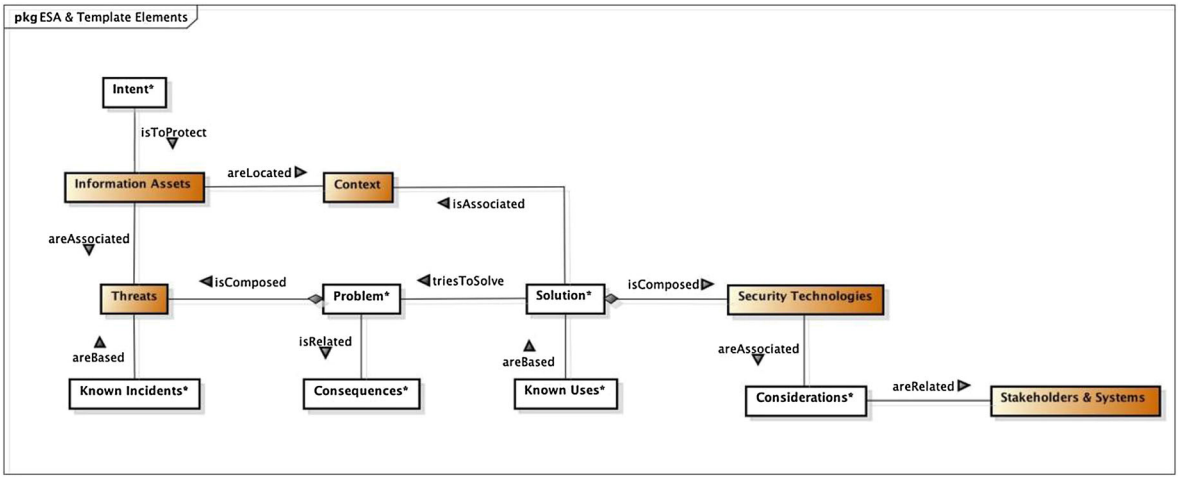


**Figure 2.** Unified Modeling Language meta-model of the enterprise security architecture elements and enterprise security pattern template.

are found; (iii) the threats associated with the assets; (iv) the security policies, patterns, mechanisms, and technologies used to stop these threats; and (v) the stakeholders and systems involved in the solution.

We will explain here the details of each one of the ESA elements used to define the enterprise security patterns through the template that helps us to document enterprise security patterns thanks to the relationships defined in Figure 2, using UML diagrams.

### 4.1. A meta-model for information assets

When building secure systems, organizations should use an information assets classification, in order to facilitate the security designer's work. The information assets should be classified into groups, according to their sensitivity record, which indicates the importance that those assets have for the organization and the protection level that has to be applied to protect them from threats and attacks. The process of creating information asset profiles helps organizations to develop an inventory of their information assets and to describe them in sufficient detail to convey their value. This value may depend on several aspects or factors. For this reason, when classifying assets, the organizations should seek support from a risk analysis methodology.

The identification of assets and their sensitivity record will facilitate the establishment of cost-effective policies to preserve these assets. For example, the brochure with the organizations' new products will need security policies related to its integrity and availability. However, the organizational information related to the purchase of a product from its competitor will need additional security policies related to its confidentiality.

As we can observe in Figure 3, the organizations' information assets may be classified into three large groups: *data*, *applications*, and *code and configuration*. Table I shows some examples of information assets in each group.

### 4.2. A meta-model for context

Considering the elements included in the context of ESAs, we define here the model of SRs and sensitivity record used in enterprise security patterns. Figure 4 presents a UML meta-model that includes these elements and the relationships between them.

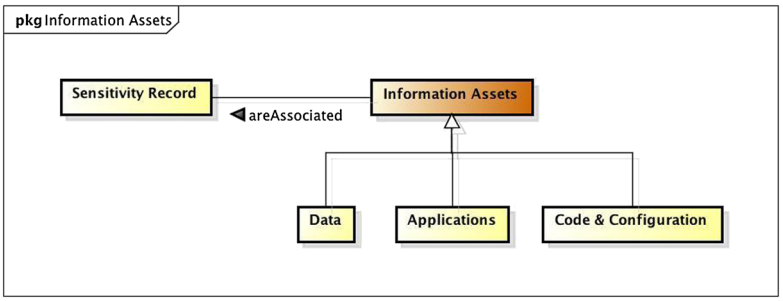


**Figure 3.** Unified Modeling Language meta-model of the information assets.


DOI: 10.1002/sec

**Table I.** Groups of information assets.

| Groups | Information assets | |
|---|---|---|
| Data | Customers | Name |
| | | Account number |
| | | — |
| | Employees | Address |
| | | Rank |
| | | — |
| | Systems | Passwords |
| | | Keys |
| | | — |
| | Organizational information | Budget |
| | | Business plans |
| | | — |
| Applications | Customers | Purchasing |
| | | Item management |
| | | — |
| | Employees | Mailbox |
| | | Payroll visualization |
| | | — |
| Code and configuration | Systems | Operating systems |
| | | Firewalls |
| | | Web servers |
| | | — |

#### 4.2.1. Security realms model

Other works [4,19] use the security domain concept to refer to this term. However, the term security domain has been adapted with different meanings in different areas, such as physical security and JBoss. We have thus decided to call it a SR in order not to confuse the reader.

Security realms can be defined as logical and discrete entities that partition the enterprise network. The main purpose of these realms is to standardize enterprise security in order to reduce the cost, users' delay, and administrative overhead of redundant security procedures. The main characteristic of SRs is that each of them has the same security policies in common. Therefore, the enterprise network can be composed of a set of SRs (sub-networks) and for each one of them different security policies can be defined.

When classifying the realms, we consider their trust level (TL) for the organization, because, depending on who is responsible for security, the security policies to apply to the realm could change. We provide a brief description of the TLs included in our model in the following texts:

- Managed (M): If the realm is managed by the security department of the organization, we have the ability to design and to implement security mechanisms within the realm.
- Externally managed (EM): If the realm is managed by another organization or partner, we can presume that this realm has reasonable security levels, but we do not have the ability to inspect them.
- Public (P): If the realm is not managed by any organization, we cannot be certain that this realm has appropriate security levels. We neither have the ability to design nor to implement security mechanisms within the realm.

When classifying the realms, we also define a set of types of realms (TR) that can be found in an enterprise network. Figure 4 shows this set of SRs that we have considered in our proposal. We provide a brief description of them in the succeeding texts. These types are based on the classification found in [4]:

- The user realm is composed of the following:
  - The customer (C) realm consists of a customer or a group of customers with the same purpose. Customers typically have permissions for reading and modifying their own data. The reading and changes that they make on the data are usually performed through specific applications. An example of this realm is a customer accessing the website from his/her smartphone.
  - The employee (E) realm consists of an employee or a group of employees with the same purpose. Employees tend to have permissions for reading and modifying their data and their clients' data. As customers, the reading and changes that they make on the data are usually performed through specific applications. An example of this realm is the staff of a bank working in the office.
  - The technical user (TU) realm consists of a technical user or a group of technical users with the same purpose. Technical users tend to have permissions for reading and modifying the applications and the code and configuration of the systems. The reading and changes that they make on the organization's information assets are usually performed directly on the asset. An example of this realm is the developer's team working in the organization's building or a developer working from an outsourced partner's building.
- The development (De) realm consists of a group of applications that are under development. These applications are only accessed by technical users. Within this realm, there is no data. An example of this realm is an application server used to perform the development and evaluation of new systems.
- The data (Da) realm consists of a group of applications and data that are being used by customers, employees, and technical users. These applications and data are in operation. An example of this data realm is a mainframe.
- The bastion (B) or gateway realm consists of a group of technologies used to separate the public realms of the managed or externally managed realms. An example of this realm is the application gateway accessed by a customer.
- The transport (T) realm consists of the parts of the enterprise network used to provide connectivity between realms. An example of a transport realm is

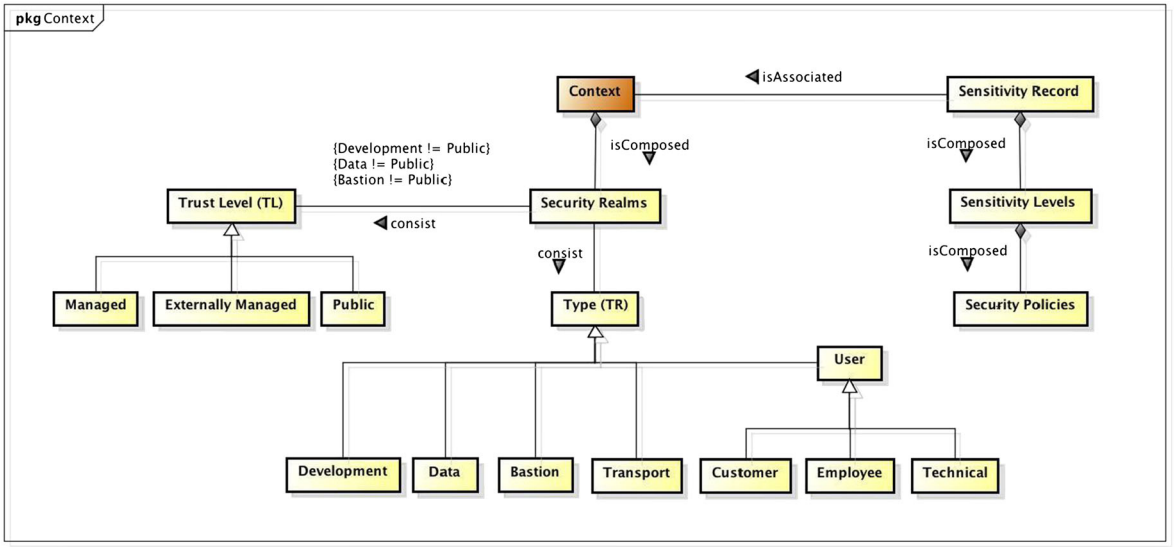


**Figure 4.** Unified Modeling Language meta-model of the context.

the system of routers that exist between the customer's router and the first router of the organization's gateway or the system of routers that exists to connect two managed data realms in different buildings of the same organization.

When we classify SRs, we must take into account the TR that can be found in an enterprise network, and who manages each of those realms, that is, their TL, because of the fact that some of the types of SRs can be managed, externally managed and public (customer, employee, technical user, and transport), whereas others can only be managed or externally managed (bastion, development, and data), that is, they must be managed by some organization. The classification of SRs that we propose here can be defined as SR: TR × TL. The specific realms can be adjusted to fit different types of applications; what matters here is that we use a classification of this type. Table II shows with a "✓" each of the 18 SRs provided in our classification.

### 4.2.2. Sensitivity Record Model

A common characteristic of all information assets is that they have to be stored in and may be transported through SRs. In terms of security, the sensitivity levels (SLs) that should be applied to all SRs included in a specific context form the sensitivity record of an information asset. The SLs can therefore be used to determine the criticality of information assets in each security realm. This SL is defined by the security policies for a specific realm and by the set of security mechanisms and techniques used to protect it. The security policies applied in each realm can vary, but we need to preserve all the required security attributes of assets (confidentiality, integrity, availability, and auditability) when they are handled or transferred within a realm. We have defined a group of security policies associated to the confidentiality that the enterprise security patterns will use to define the SL of an information asset. This set of security policies is made up of the following:

- Secure channel (SC). Policies defined to secure the transport channel.
- Clear channel (CC). Policies defined to ensure that the transport channel is clear.
- Blocked channel. Policies defined to block the transport channel.
- Hidden storage. Policies defined to secure the information stored.

**Table II.** Classification of security realms.

| | | Trust level | | |
|---|---|---|---|---|
| | | Managed | Externally managed | Public |
| Types of realms | Customer | ✓ | ✓ | ✓ |
| | Employee | ✓ | ✓ | ✓ |
| | Technical user | ✓ | ✓ | ✓ |
| | Development | ✓ | ✓ | — |
| | Data | ✓ | ✓ | — |
| | Bastion | ✓ | ✓ | — |
| | Transport | ✓ | ✓ | ✓ |

- Clear storage. Policies defined to store the information of clear way.
- Blocked storage. Policies defined to block the storage system.

To establish the sensitivity record of an information asset in a context, the security engineers should obtain the SL of that asset for all SRs included in the context. To obtain the SL of that asset in each security realm, the security engineers should respond to four dependent questions related to the following security aspects: access authorization, encryption, and storage authorization, which are related with the two possible states that an information asset can be (in transit or stored). The four questions are listed in the following texts:

1. Can the information asset A be transported by using the security realm SR?
2. If so, should A be encrypted?
3. Can A be stored in SR?
4. If so, should A be stored in a hidden form?

According to the answers to these questions, the security polices will be assigned, as shown in Table III. In this table, we can see in the first column a number that denotes the SL provided by each set of security policies (1 is the lowest and 6 the highest).

As we can see in Table III, we obtain seven different SLs, but one of them is not applicable (CC & hidden storage,, the shadowed row) because it is not a usual find security policies in which the information assets require encrypted storage and may be transported in a clear way. When protecting the information assets, enterprise security patterns will use additional security policies, such as integrity policies, availability policies, and auditability policies.

The output of the Sensitivity Record Model is a set of numbers that represent the SL of an information asset for each of the SR included in the context. These numbers will help security engineers when designing the solutions of the enterprise security patterns. From this set of numbers, we can decide on the security policies that will be applied to each security realm as a solution in our enterprise security pattern.

Before using enterprise security patterns, organizations' security engineers should create information asset profiles and respond to the four questions listed in the previous texts for each security realm included in the context. Several organizations could apply different sensitivity records to the same asset. For example, when classifying the customers' account, a food industry organization could decide to apply low or medium SLs in all its SRs. However, a banking organization could decide to apply high or very high SLs. Because of this, enterprise security patterns do not try to protect single information assets. They intend to protect information assets with the same SL in a particular context.

### 4.3. A meta-model for threats

Figure 5 shows a UML meta-model for the threats element. As we can see, the problem that an enterprise security pattern attempts to solve considers the threats associated with the information assets and the forces that enable those threats. The forces should clarify the intricacies of the problem and make explicit the kinds of trade-offs that must be considered. The enumeration of consequences should match the forces identified, but there may be consequences that do not correspond to any force.

By using the sensitivity record of the information assets and the threats associated with these assets, security designers could verify if the security properties (confidentiality, integrity, availability, and auditability) of a specific information asset needs to be protected or not, depending on its sensitivity record and its threats. For example, a designer knows that an information asset A is susceptible to a sniffing attack Sa in a security realm R. Sa could violate the confidentiality of A using the communication channels. If the sensitivity record of A does not require securing the communication channels in R, the designer does not need to protect A from Sa. However, the security designer should protect the asset in all cases if the threats associated with the information asset violate security properties required by its sensitivity record.

### 4.4. A meta-model for security technologies

Model-driven architecture (MDA) [26] is the approach defined by the Object Management Group for software development under the model-driven engineering framework. MDA defines three viewpoints of a system, which are modeled with specific models: (i) the CIM, which is used by the business analyst and is focused on the context and requirements of the system without

**Table III.** Security policies of the sensitivity level.

| | | Answers combinations | | | |
|---|---|---|---|---|---|
| SL | Security policies | 1 | 2 | 3 | 4 |
| 4 | Secure channel and hidden storage | Yes | Yes | Yes | Yes |
| 3 | Secure channel and clear storage | Yes | Yes | Yes | No |
| 5 | Secure channel and blocked storage | Yes | Yes | No | — |
| — | Clear channel and hidden storage | Yes | No | Yes | Yes |
| 1 | Clear channel and clear storage | Yes | No | Yes | No |
| 2 | Clear channel and blocked storage | Yes | No | No | - |
| 6 | Blocked channel | No | — | — | — |

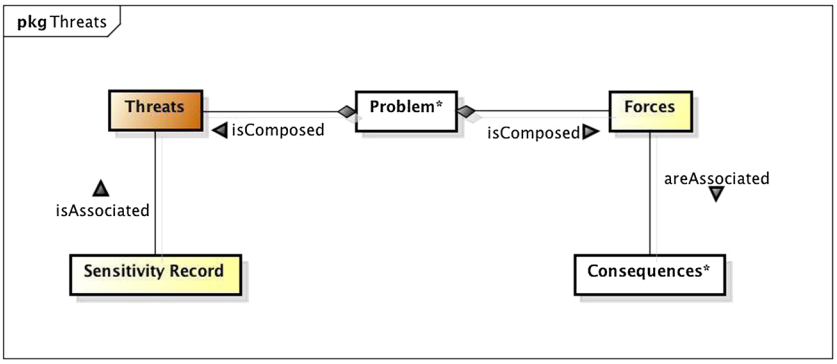


**Figure 5.** Unified Modeling Language meta-model of the threats.

considering its structure or processing; (ii) the PIM, which is used by software architects and designers and is focused on the operational capabilities of a system outside the context of a specific platform; and (iii) the PSM, which is used by software developers and programmers and includes details related to the system for a specific platform [27].

To describe the solution of an enterprise security pattern, we have based it on a MDA but adapting the architecture to the security context and the enterprise environment where the technological part has a greater importance. As we can observe in the UML meta-model for the solution element, we define four viewpoints, three of them fit in CIM, PIM, and PSM and the fourth viewpoint, the PDM, which is related to the technological environment, that is, the security technologies provided by the solution (Figure 6).

This architecture proposes not only a set of models that represent the system at different abstraction levels but also a software development lifecycle [28] with which to perform the following: (i) capture requirements in a CIM; (ii) create one or more PIMs (it is sometimes possible for part of the PIM to be obtained from the CIM); (iii) transform the PIM into one or more PSMs, adding platform specific rules that the transformation did not provide; and (iv) transform the PSM into one or more PDMs, adding different existing technological products in the security industry. We discuss in the following texts the four models included in the solution of these patterns:

CIM: This model provides a description of the security policies that the system should enforce independent of its functional and technological characteristics. The security policies defined in the sensitivity record of the information assets (Section 4.2.2) should be applied to the SR included in the context. When building secure systems, the CIM could help us to define the security requirements of the systems to be protected.

PIM: This model provides a conceptual description of the security mechanisms that should be incorporated into the system and the relationships that exist between them, independent of its technological characteristics and implementation detail. The same CIM could be instantiated N times in this model, because a security policy may correspond to different security patterns. A good guideline that can be used as a basis to select the security patterns needed are the guidelines developed by Schumacher *et al.* in [29] or Fernandez in [30]. When building secure systems, the PIM could help us to use the enterprise security patterns in the analysis stages of the security methodologies.

PSM: This model defines the architectural components included in the ESA, independently of the technology used to solve the problem. The PSM should take into account how to place the security mechanisms within the architecture. The same PIM can be instantiated N times in this model, because a security mechanism may be placed in different architectural components. The security patterns described in the PIM are included within the architectural security components. Two good guidelines that can be used as a basis to select the architectural component are the ISO/IEC-27000-series [31] and the IT Baseline Protection Manual [32]. The PSM could help us to use the enterprise security patterns in the design stages of the security methodologies.

PDM: It is necessary to install the PSM in a specific technological architecture. The same PSM could be instantiated N times, because the same architectural component may correspond to different technological products. The technological products must be reputable products made by known manufacturers in the security industry. The final solution may vary significantly depending on the technologies used.

As we can also see in Figure 6, the four models of the solution have some elements in common. As we showed previously in Section 3, the solution of these patterns tries to solve a problem in a specific context. This means that the four models build their solution based on the same set of SR and take into account the set of threats included in the problem of the

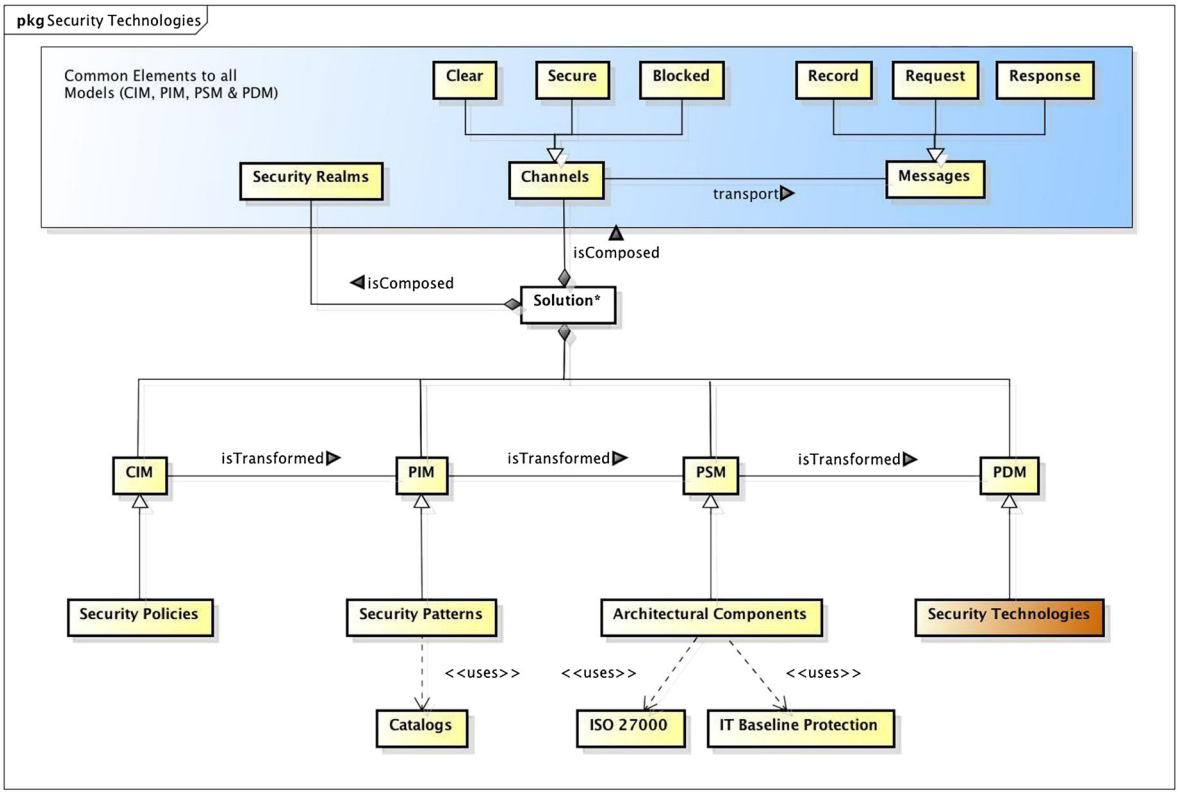


**Figure 6.** Unified Modeling Language meta-model of security technologies.

pattern. In addition, all models used to build the solution use a set of channels or communicators. These channels link users, model components, and information assets. Each of the channels has a sender and a receiver. The types of channels that we may find within the solutions of enterprise security patterns are CC, SC, and Blocked Channel. In order to show a logical representation of the type of message that they could transport, we define below three types of messages that could be sent through the CC and SC:

- *Request message*: A sender transmits a request to a receiver through the channel.
- *Response message*: A receiver responds to a request sent by a sender through the channel.
- *Record message*: A sender transmits relevant information to a receiver through the channel. The receiver must record this information.

## 4.5. A meta-model for stakeholders and systems

Figure 7 shows a UML meta-model for stakeholders and systems. As we can see in this figure, enterprise security patterns present a qualitative analysis (or set of considerations) of the PDM of the solution. The set of considerations is divided into two parts:

1. The qualitative analysis on the systems or technologies involved in the solution. This analysis is related to the following:
   - Performance overhead (storage, primary memory, processor, and bandwidth) of the solution.
   - Installation cost of the solution.
   - Complexity of massive expansion of the solution.
   - Residual risk of the solution, that is, if the ESA would need complementary measures to attain its objective.
2. The qualitative analysis on the complexity of the solution for the following stakeholders: security administrator, log administrator, end user, and systems administrator.

When carrying out the analyses, we consider if the deployment of the solution alters qualitatively each of the aspects listed above in a null (0), low (1), medium (2), or high (3) manner.

As we have seen in this section, the considerations provided by these patterns may support means to carry out an analysis of cost, performance, and complexity of the technological solution. For this reason, enterprise security patterns could also be used in the initial stages (the inception or feasibility stage) of the security methodologies as a way to estimate costs and efforts.

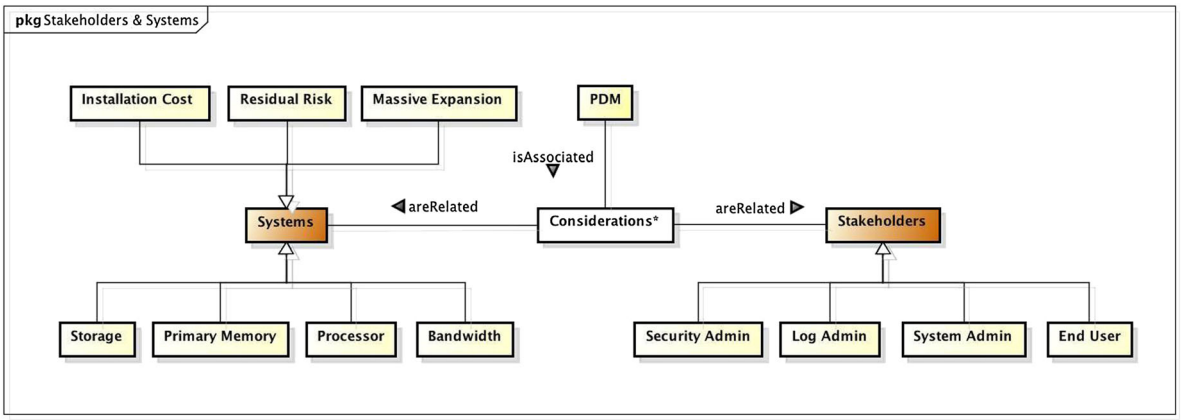


**Figure 7.** Unified Modeling Language meta-model of stakeholders and systems.

## 5. A SET OF DIAGRAMS TO REPRESENT ENTERPRISE SECURITY PATTERNS

We define here five new diagrams in order to graphically represent the context and the four solution models of enterprise security patterns. Table IV shows the icons library that these patterns will use when representing their context and solution. We discuss in the succeeding texts what elements should be included in each of the diagrams.

### 5.1. Context diagram

The context diagram of the enterprise security patterns should include the SR in which the information assets to be protected are stored, from which the users access them, and through which the information assets are transported. The icons and acronyms to represent the type of realm should appear at the top, whereas a name to identify the realm should appear at the bottom. The users, information assets, and channels between them should also be represented. As we can see in Table IV, the types of users and information assets may vary. In this diagram, the channels between users and information assets should not represent the type of channel that they are, or the type of message that they convey. Figure 8, in Section 6, shows an example of this type of diagram.

Within the scope of enterprise security patterns, each context is unique. When using enterprise security patterns to define and to build the security architecture of a new system, security engineers should find a context diagram according to their initial requirements. If there is, they should also review the remaining diagrams to know if the pattern can be useful (more detail in the following sections). If not, a new enterprise security pattern should be discovered. A methodology to discover new enterprise security patterns is proposed as future work, in order to create a catalog of this kind of patterns and facilitate the work of organizations' security engineers when designing new architectures.

### 5.2. Computation Independent Model diagram

The CIM diagram of the enterprise security patterns should show all elements included in the context diagram, except the name that identifies each SR. This name should be replaced by the security policies included in the SL of the information assets to protect. For this reason, the icons of security policies and the number that denotes the SL to apply in each realm should appear at the bottom. Figure 9, in Section 6, shows an example of this type of diagram.

As we said previously, a context is associated with a problem, and so with a set of threats. Threats that must be mitigated in a specific context depend on the SL of the information assets to be protected. As different information assets may have different SLs, the same context diagram may be transformed into multiple CIM diagrams.

For this reason, although security engineers find a context diagram according to their initial requirements, it does not mean that the enterprise security pattern may be useful for them. They should also find a CIM diagram according to the SL of the information assets they intend to protect. If they do, they could review the remaining diagrams to analyze, to design and to implement the new security architecture. If not, a new solution (CIM, PIM, PSM, and PDM diagram) should be discovered for that enterprise security pattern. Because of this, an enterprise security pattern could have more than a solution.

### 5.3. Platform Independent Model diagram

The PIM diagram of the enterprise security patterns should show all elements included in the CIM diagram and the security patterns used to protect the information assets of the threats found in the problem. Only threats that may endanger the SL of the information assets should be mitigated. The communications channels between users, information assets, and security patterns should also be represented. As we can see in Table IV, the types of channels are: clear, secure and blocked.

**Table IV.** Icons Library.

| | | Element | Icon |
|---|---|---|---|
| | Information assets | Data |  |
| | | Applications |  |
| | | Code and configuration |  |
| Security realms | Types | Customer |  |
| | | Employee |  |
| | | Technical user |  |
| | | Data |  |
| | | Development |  |
| | | Bastion |  |
| | | Transport |  |
| | Trust level | Managed |  |
| | | Externally managed |  |
| | | Public |  |
| Model components | Security policies | Clear channel |  |
| | | Secure channel |  |
| | | Blocked channel |  |
| | | Clear storage |  |
| | | Hidden storage |  |
| | | Blocked storage |  |
| Security patterns | | |  |
| Architectural components | | |  |
| Technological products | | |  |
| | Channels | Clear |  |
| | | Secure |  |
| | | Blocked |  |
| | Messages | Request or response |  |
| | | Request and response |  |
| | | Record |  |

The types of messages that channels can convey are: request, response and record. The data flow should also be denoted including sequential numbers in the security patterns used. Figure 10, in Section 6, shows an example of this diagram.

Considering that a CIM diagram could be transformed into multiple PIM diagrams, when using enterprise security patterns, security engineers should find a PIM diagram that meets the objectives defined in their system requirements. If found, they could analyze the security of the new system based on this diagram and review the remaining diagrams to design and implement the security architecture. If not, they could modify the security patterns included in the PIM diagram in order to tailor their own solution, but they should consider two possible consequences associated with this change. On the one hand, they should ensure that changes introduced do not compromise the information assets to be protected. On the other hand, they should take into account that a change in the PIM diagram will cause changes in the remaining diagrams (PSM and PDM diagrams).

### 5.4. Platform Specific Model diagram

The PSM diagram of the enterprise security patterns should show all the elements included in the PIM diagram and the architectural security components that should be deployed to protect the information assets. Each of the architectural security components should consist of one or more security patterns. Figure 11, in Section 6, shows an example of this type of diagram.

As in previous diagrams, a PIM diagram could be transformed into multiple PSM diagrams. For this reason, when using these types of patterns, security engineers should find a PSM diagram that meets the design requirements defined for the new system. If there is such a diagram, they could base the security design of the new architecture on this diagram and review the PDM diagram in order to implement it. If not, they could modify the architectural security components provided by the pattern to design their own solution. As in the aforementioned diagram, they should ensure that changes introduced do not compromise the information assets to be protected and take into account that a change in the PSM diagram will cause changes in the PDM diagram.

### 5.5. Product Dependent Model diagram

The PDM diagram of the enterprise security patterns should show all elements included in the PIM diagram and the technological products that an enterprise could purchase and deploy to protect the information assets. The architectural security components of the PSM diagram should be transformed in technological products. Figure 12, in Section 6, shows an example of this type of diagram.

In the security industry, it is common to find different technologies that perform similar work. The main difference between them is usually related to the precision and performance that they offer. When using enterprise security patterns, security engineers could implement the security technologies provided by the pattern or find a set of security technologies that do similar work. It is important that all security technologies implemented have been developed by well-known companies in the security industry and have been tested long enough to show that they are reliable.

## 6. AN ENTERPRISE SECURITY PATTERN: SECURE EXTERNAL ACCESS TO A PRODUCTION ENVIRONMENT

To make our ideas clearer, we present here an enterprise security pattern instance or example. This pattern could be used by organizations of different sectors or industries. We discuss in the succeeding texts each of sections included in the pattern template.

### 6.1. Intent

This pattern attempts to protect the data accessed by the applications developed and maintained by outsourced companies. The data and applications are stored within the data center of the organization.

### 6.2. Context

Figure 8 shows a diagram of the context of this pattern. As we can see in this figure, a group of technical users perform the development and maintenance of applications of an organization from an outsourced company (*externally managed technical user realm*, (*EM-TU*)) using a browser. Two technical user icons are shown to represent plurality, but the number of technical users could be higher, or even lower. Internet Explorer, Firefox, and Chrome icons are shown to represent the most important browsers, but technical users could use

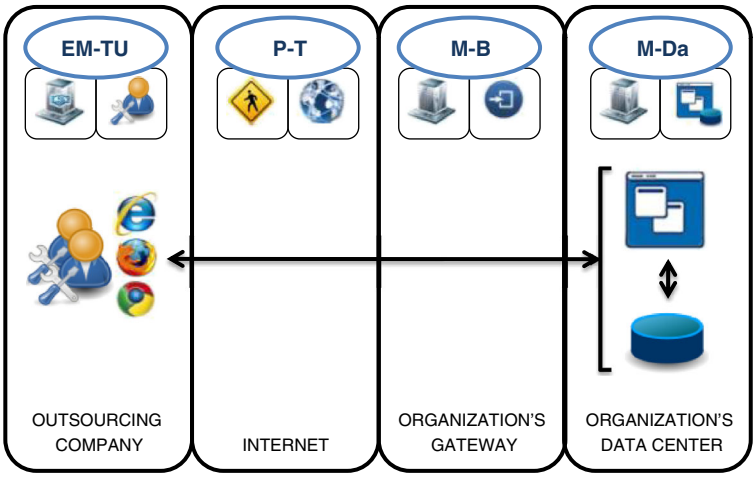


**Figure 8.** Context diagram.


*Security Comm. Networks* 2014; **7**:1670–1690 © 2014 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

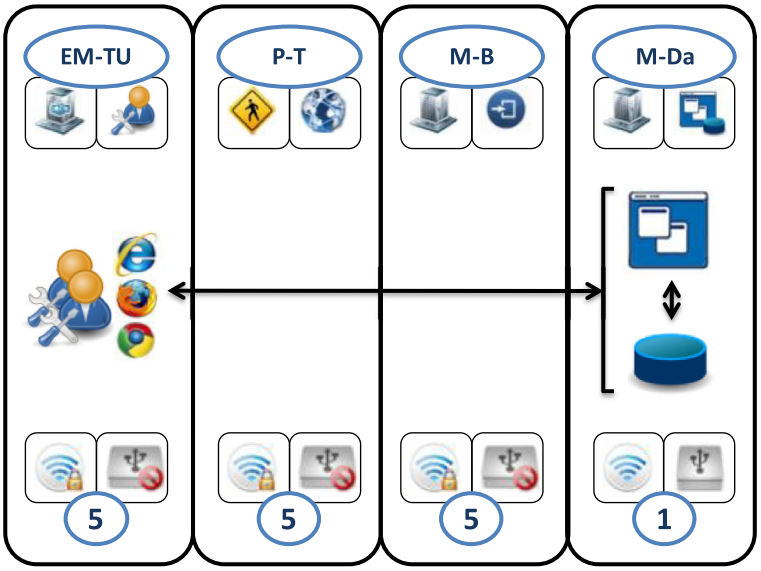


**Figure 9.** Computation Independent Model diagram.

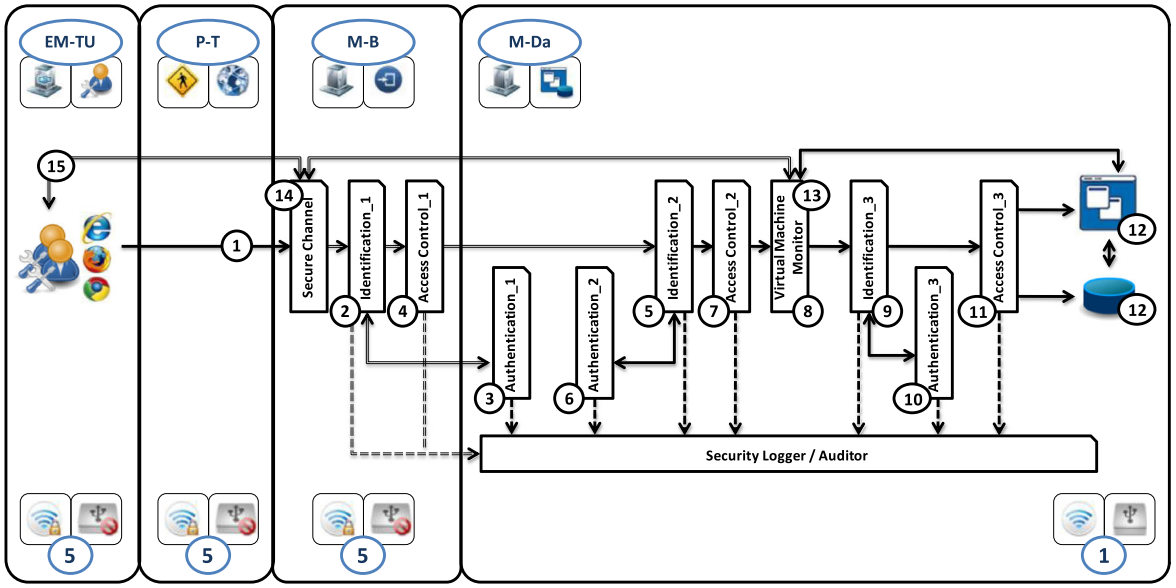


**Figure 10.** Platform Independent Model diagram.

any browser that complies with international standards marked by the *World Wide Web Consortium*. The applications are placed in one of the organization's data centers (*managed data realm*, (*M-Da*)). The outsourced company accesses the organization via the Internet (*Public Transport realm*, (*P-T*)). The organization has an applications gateway (*managed bastion realm*, *M-B*) between the Internet and the data center.

The sensitivity record (Section 4.2.2) of the information assets or data that this pattern attempts to protect is shown in Table V.

As shown in that table, the data accessed by applications should only be stored in the organization's data center (M-Da). The data could leave the organization, but the communication channels of realms through which they have to be transported should be secure, excepting the channels within M-Da. This pattern should be used when the organization needs to ensure that we only need to protect the information assets that meet this SL. If the information assets to be protected have other SLs, it is possible that this solution needs more or fewer defense mechanisms.

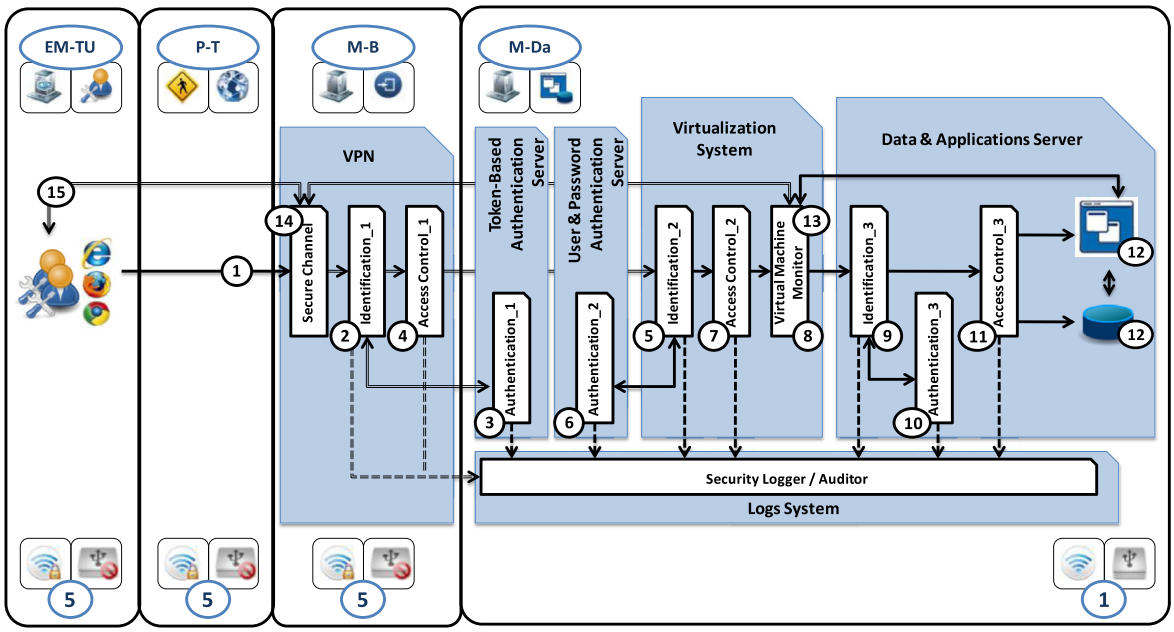


**Figure 11.** Platform Specific Model diagram.

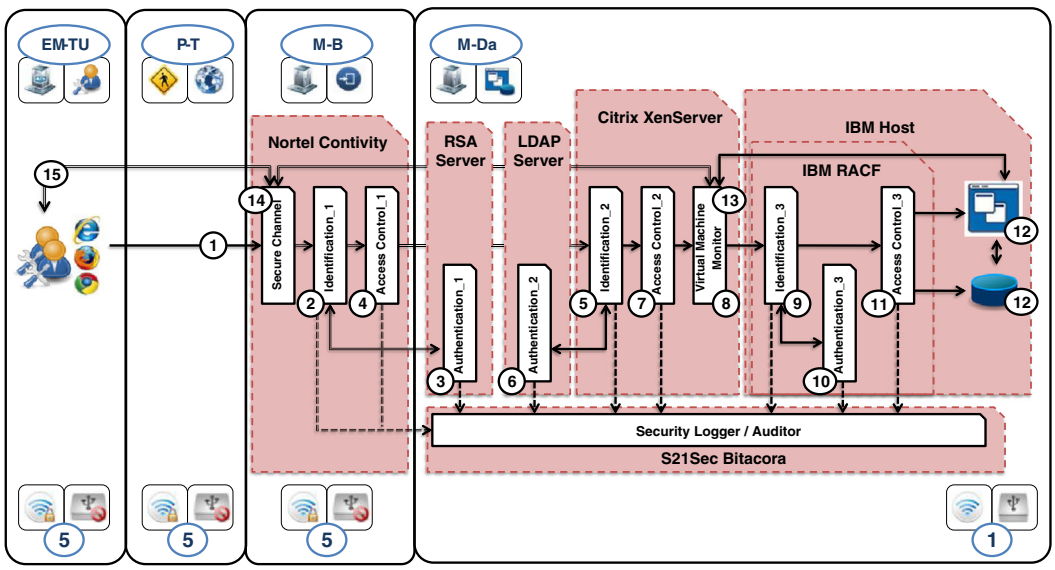


**Figure 12.** Product Dependent Model diagram.

## 6.3. Problem

In the past, the technical users of organizations developed and maintained applications from a network directly connected to the data center. The user network and the data center were not connected to the Internet. This context provoked a set of threats. Apart from threats that should also be handled related to the integrity and availability, some threats related to the confidentiality of data in this environment are the following:

- An external attacker may acquire applications' data via the Internet. To prevent this, organizations have to prevent external communication channels transport data in a clear way.
- An unauthorized user may access the applications source code or the data accessed by them. To prevent this, organizations need to ensure that only authorized users may access their data and applications.
- An unauthorized user with physical access to the user network may copy data or the access credentials for

**Table V.** Sensitivity record of the information assets.

| Security realms | Security policies | SL |
|---|---|---|
| Externally managed technical user | Secure channel and blocked storage | 5 |
| Public transport | Secure channel and blocked storage | 5 |
| Managed bastion | Secure channel and blocked storage | 5 |
| Managed data | Clear channel and clear storage | 1 |

acquiring applications' data. To prevent this, the organizations need to prevent unauthorized users from intercepting information of the legitimate users and data network.

- An authorized user may provide an external attacker or an unauthorized user data accessed by the applications. Organizations could reduce the risk of this threat by being very restrictive, that is, by avoiding external storage devices on users' computers (USB, CD, DVD, etc.). However, they cannot prevent it. A user may still record the data with a video camera or smart phone from the screen of its computer.

Once the distributed systems paradigm was born, the context for developing or maintaining an organization's applications has changed significantly. In addition, many companies have as a main objective application development and maintenance for other organizations. The threats that the organizations have to handle in these contexts are similar to those listed previously, but the method of handling them is different.

## 6.4. Known incidents

Two known incidents of information asset theft from large organizations are shown the succeeding texts. 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 was the theft of a large amount of money ($2m) from Citibank [33]. Russian hackers accessed critical customer information from Citibank via an Structured Query Language (SQL) injection attack to the website of the American chain store 7-Eleven. At the time of the information theft, there were 5500 Citibank-branded ATMs at 7-Eleven stores. As soon as the hackers had obtained duplicate bank cards and their associated personal identification numbers, they began to withdraw money and pay for goods using the duplicated credit cards.

Knowing that the hackers used a Structured Query Language injection attack of the website, one hypothesis of how they stole the information is to assume that the ATMs were also connected to the internal network of 7-Eleven. In this way, the hackers would be able to monitor the transactions of Citibank's ATMs. If this hypothesis is true, Citibank could have prevented the theft using the pattern that we are describing here (by making the ATMs the technical users).

The second incident is the theft of information from Epsilon [34], the world's largest permission-based e-mail marketing services company. Epsilon sends over 40 billion emails annually and has over 2500 clients, including seven of the Fortune 10 for which it builds and hosts 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: Kroger, TiVo, US Bank, JPMorgan Chase, Capital One, Citibank, Ameriprise Financial, Lacoste, Hilton Honors Program, and Marks & Spencer.

The published news does not give the details of how the information was stolen. Therefore, it is not possible to determine the security measures that Epsilon should have taken to prevent the attack. However, by applying the pattern that we are describing here, Epsilon could have prevented the threats listed in the problem. Although it had assumed some threat, for example, the authorized users could record the customers' information with a video camera or smart phone from the screen of their computer, a massive data theft is not possible using this technique.

## 6.5. Solution

The solution implies four abstraction models: the CIM, the PIM, the PSM, and the PDM. The CIM applies the sensitivity record of information assets on the SRs. The PIM enforces the CIM, refining the security policies included in the sensitivity record as security patterns. The PSM transforms the security patterns of the PIM into architectural security components. Finally, the PDM transforms the architectural security components into technological products.

### 6.5.1. Computation Independent Model

We need to apply here the security policies included in the sensitivity record of the information assets. As shown in the diagram of the CIM, we could prevent an external attacker viewing the applications' data by encrypting the channels and not allowing data to be stored in EM-TU, P-T, and M-B (Figure 9). To prevent an external attacker acquiring the applications' data in realms where the information assets can be stored, that is, M-Da, we will use some of the security mechanisms included in the PIM of the solution.

### 6.5.2. Platform Independent Model

We define here the security policies of the CIM as security patterns. All security patterns included in the PIM diagram are described in [29], except *virtual*

*machine monitor* and *security logger/auditor* described in [35] and [17], respectively (Figure 10). Instantiations of the same pattern P are denoted as P_1, P_2, and so on. The types of channels that it is a possible find within the PIM are CC (single line) and SC (double line). What is more, these channels show a logical representation of the type of message that they transport. The types of messages that could be transported are request or response message (solid line) and record message (dashed line). The numbers (1–15) in the center of the diagram represent the sequence of actions that organizations should deploy in their ESAs in order to prevent or mitigate some of the threats identified in the problem. An alteration of the sequence of actions could reduce the effectiveness of the pattern. We show in the succeeding texts how each security pattern or set of security patterns included in the PIM diagram help to prevent or mitigate these threats.

In order to assure communication channels between the organization's gateway (M-B) and the outsourced company (EM-TU), we have included within M-B the security pattern SC.

In order to prevent unauthorized users or external attackers accessing the systems of the data center (M-Da), we have included the security patterns *Identification_1*, *Authentication_1*, and *Access Control_1*. In this case, Access Control_1 should only check if the Authentication_1 is valid or not. If not valid, the access of the technical user should be denied.

In order to ensure that the information assets cannot be stored in EM-TU, P-T, and M-B, we have included the security patterns *Identification_2*, *Authentication_2*, *Access Control_2*, and virtual machine monitor. In this case, Access Control_2 should check if Authentication_2 is valid or not, and if valid, the group to which the technical user belongs. Virtual machine monitor should show the technical user a virtual desktop depending on their group through a SC. Using these four security patterns, we could reduce the risk to data accessed by the applications, which a technical user provides to an external attacker or to an unauthorized user. To do this, the virtual desktops should not allow external storage (USB, CD, DVD, etc.) or Internet access. As in centralized contexts, we cannot prevent a technical user from recording the data from the screen of their computer with a video camera or smart phone.

In order to prevent the technical users from reading or modifying an applications source code, or data accessed by applications for which they have no permissions, we have included the security patterns *Identification_3*, *Authentication_3*, and *Access Control_3*. In this case, Access Control_3 should check if the Authentication_3 is valid or not, and if valid, the role to which the technical user belongs. The permissions to read and to modify the data and applications will depend on her role. We have instantiated the authentication pattern three times, because each of them uses a different authentication mechanism (more detail in the PSM).

In order to audit possible attacks, the patterns of identification, authentication, and access control should record all activity using the security pattern security logger/auditor.

The pattern includes three identifications, but the technical user should only present the access credentials once. With the aim of reducing the complexity of the solution, the credentials should travel seamlessly through the components of the architecture. Once the user has been successfully validated by the three access controls, the data and applications requests would pass only through the SCs and virtual machine monitor. The same should occur with the data and applications responses. These facts are shown with three bidirectional arrows that link technical users SCs virtual machine monitor data and application*s* (at the top of the pattern).

### 6.5.3. Platform Specific Model

We transform here the security patterns of the PIM into architectural components. As shown in the diagram of the PSM, the security pattern instantiations SC, Identification_1, and Access Control_1 form a *virtual private network* within the organization's gateway (M-B) (Figure 11).

In order to prevent an unauthorized user from accessing applications' data using the access credentials of an authorized user, the ESA must ensure that the technical user is who she claims to be. To do this, an authentication system with high security level should be used (something stronger than passwords). In the solution of the pattern, we have decided to include *token-based authentication*, because its use is currently more widespread, but we could also have used biometric authentication or some other kind of strong authentication. It is important that the token-based authentication server is placed within the data center. This is because the authentication systems should never be accessed directly from public networks.

The authentication system associated with the *virtualization system* does not need to be based on tokens. This is because the architecture is already sure that the user is who they claim to be. However, it is still necessary to know who is accessing the virtualization system. For this reason, an *authentication server* based on *user and password* is sufficient in this case.

Finally, we can see in Figure 11 that all architectural security components have to register their activity in the *logs system*, and the authentication system associated to the *data and applications server* is included within the server. As we said before, this authorization system should be based on roles (role-based access control).

### 6.5.4. Product Dependent Model

We transform here the architectural security components in technological products. The diagram of the PDM, Figure 12, shows the technological products that we have decided should be included in the solution. We have selected these security technologies because we consider them reliable and currently used by many

**Table VI.** Considerations.

| | | Aspects to consider | Analysis |
|---|---|---|---|
| Systems | Performance overhead | Storage | 1 |
| | | Primary memory | 3 |
| | | Processor | 3 |
| | | Bandwidth | 2 |
| | | Installation cost | 3 |
| | | Massive expansion | 0 |
| | | Residual risk | 0 |
| | Stakeholders | Security administrator | 0 |
| | | Log administrator | 2 |
| | | End user | 0 |
| | | System administrator | 0 |

organizations. But we could have selected another set of similar security technologies.

### 6.6. Considerations

The considerations of the pattern show a qualitative analysis of the selected technologies in the PDM of the solution. Another set of technologies could change this analysis. Table VI shows the result of the analysis for each of the relevant aspects (Section 4.5).

### 6.7. Consequences

With the four models of the solution, we have tried to prevent or to reduce the risk of the threats found in the problem. As we have said previously, these threats are related to the confidentiality of the assets. Integrity and availability threats could be handled in similar ways. We discuss in the succeeding texts the security mechanisms that we have included in the pattern, in order to prevent or reduce the risk of the identified threats:

- An external attacker may acquire the applications' data in transit. To prevent this, we include a virtual private network in the organization's gateway.
- An unauthorized user may access the applications source code or the data accessed by them. To prevent this, we include an authentication and access control system before accessing data and applications.
- An unauthorized user with physical access to the user network may copy the access credentials for the applications and then access the data. To prevent this, we include a token-based authentication.
- An authorized user may provide data from an application to an external attacker or an unauthorized user. We could reduce the risk of this threat, by avoiding the users' virtual machines having external storage (local HD, local USB, local CD, local DVD, etc.) and Internet access. However, users may record with a video camera or smart phone the data from the screen of their computers, regardless of whether they are in a managed or externally managed network.

This pattern would also be applicable in a context where the technical users perform applications development and maintenance from their home (Public Technical User realm) rather than from the outsourced company.

### 6.8. Known uses

Banco Bilbao Vizcaya Argentaria Group is currently using the solution offered in this pattern to reduce the risk of information leakage, when the technical users of outsourced companies access development and production environments from outside the organization in order to perform the construction and maintenance of corporate information systems.

## 7. RELATED WORK

Security patterns share the advantages of design patterns in that they allow the re-use of the knowledge and experience of many designers. However, as indicated in Section 1, they have some limitations of which the most important is that it is not easy for inexperienced designers to apply them. In addition to this type of security pattern, other researchers have proposed variations of the concept:

1. Kienzle *et al*. [18] described 26 patterns and 3 mini-patterns. The authors defined the scope of the problems their patterns address, by focusing on the domain of web application security. The patterns are divided between structural patterns and procedural patterns. They follow a simpler template than that was used in [29,30], and the solutions are presented in words, and may even including a block diagram, and they do not take into account the stakeholders involved in the design. Our proposal follows a more complete template, and the solutions are presented in both a textual and graphical manner.
2. Fernández *et al*. presented an approach with which to obtain "secured" middleware patterns [36], for example, a secure broker or secure pipes and filters. Whole systems of patterns can thus be secured by adding patterns to cover possible threats. Our approach follows

this idea of building whole systems and adding enterprise security patterns defined with an extended template into which elements of ESA are incorporated, showing possible solutions and technologies that can be applied to solve a problem or security threat.

3. Mouratidis presented Secure Tropos [7], which uses agent-oriented patterns with a template and representation that are different from those of standard patterns. Secure Tropos has its own notation and does not use UML. The patterns themselves are mostly intended to define semantic business constraints without defining software aspects in detail. Our approach also defines its own graphical notation, but it is focused on software aspects and technological environments, and many details are added to certain elements of the pattern such as an extended MDA model.
4. Georg *et al*. used a template-pattern that is instantiated as a security aspect in [6]. The definition of the pattern itself is a template in the C++ sense, signifying that the pattern is not a guideline but an instantiable class. This approach requires the pattern to fit exactly in the requirements of the problem, and its use is therefore more constrained than the use of standard security patterns. It does not take into account the stakeholders involved in the design of the system. Our proposal serves as a guideline for designing and building ESAs with the use of standard security patterns adapted to enterprises.
5. Jackson's problem frames [37] are converted into security problem frames in [8]. Schmidt *et al*. [8] presented a security engineering process based on security problem frames and concretized security problem frames. Both kinds of frames constitute patterns with which to analyze security problems and associated solution approaches. They are arranged in a pattern system that makes the dependencies between them explicit. The authors provide a step-by-step description of how the pattern system can be used to analyze a given security problem and how solution approaches can be found. Jackson's [37] approach is useful to describe contexts and requirements, but it is not sufficiently detailed to build security mechanisms. Our proposal offers recommendations about certain security mechanisms to be used in the security pattern in addition to other aspects such as solutions, threats, or known incidents.

Although conventional security patterns can be made easier to use by adding classifications and tools, we do not believe that the other variety of security patterns are appropriate for use with complex systems. They are too limited to be useful in practice.

The proposals analyzed are focused on defining security patterns or methodologies into which the security patterns are incorporated and used to develop secure software, and the selection of the most appropriate patterns is not solved for a specific context. Moreover, none of the proposals analyzed have considered all the elements of an ESA, as we can see in Table VII. The definition of an enterprise security pattern that encompasses all elements of an ESA therefore helps us to design and to develop secure software for business environments. This is because the enterprise security pattern used incorporates the most important security issues (refer to the last row of Table VII), and it is not therefore necessary to combine and to select a set of existing security patterns and to tailor them to our business environment.

## 8. CONCLUSIONS AND FUTURE WORK

Security patterns are not applied when building ESAs as much as they could be because designers have problems in selecting and applying them in the right places. Enterprise security patterns could improve the application of the patterns by incorporating them in a more comprehensive pattern that may handle more threats and could be easier to select because of their smaller number. In addition, enterprise security patterns combine, in one cohesive pattern, all the elements that must be considered when designing ESAs.

There are several approaches that define security patterns with different templates and elements in different ways and for different stages of the development life cycle, and none of them have considered all the elements of an ESA for business environments. In this paper, we have defined a security pattern related to ESA in which the security elements of this architecture are incorporated into the template of the pattern with the aim of building secure systems for an enterprise environment. The designers therefore have a more reduced set of security patterns to select from, in addition to a guideline of how to apply this pattern to the design, and all

**Table VII.** Comparison between security patterns and methodologies proposals with elements of an enterprise security architecture.

| ESA elements Proposals | Goal | Context | Threats | Stakeholders | Security technologies |
|---|---|---|---|---|---|
| Kienzle | | X | X | | X |
| Fernandez | | X | X | | X |
| Mouratidis | X | X | X | X | |
| Georg | X | X | X | | X |
| Schmidt | | X | X | | |
| Our proposal | X | X | X | X | X |

of this is always focused on enterprise environments. This enterprise security pattern and others that are being defined are applied, checked, and validated in a real case in the banking sector with the aim of justifying their validity and usefulness in a real business environment. This validation allows us to improve, update, and add new aspects, relationships, and elements of the template to cover all the needs and constraints of enterprise environments.

The template defined considers aspects such as the context of the pattern, the description of the problem to be solved together with the threats, possible solutions, technological considerations, or examples of known incidents. It is clear that our pattern cannot deal with unknown threats or incidents introduced into our system because we do not have information about this threat, about what asset acts, which security mechanisms it is necessary to use to protect us from this unknown threat, what its impact is, and so on. Once we have more information about an unknown threat, we can define a new pattern or update an existing pattern and add this information about this unknown threat that is known once the threat has been introduced into our system, and we have seen the means of acting and how it can be solved by using our security pattern.

As future work, a security methodology based on this type of templates could be defined. It would not be possible to use this methodology without an extensive catalog of patterns and a framework with which to discover new enterprise security patterns should therefore also be defined. In order to facilitate the definition and the use of these patterns, a set of tools to support the design and construction of secure information systems could also be built. The new patterns could also be incorporated into an existing methodology such as [5], in which they could complement the use of individual patterns.

## ACKNOWLEDGEMENTS

This research has been hosted at Florida Atlantic University (FAU) and carried out in the framework of the following projects: MASAI (TIN2011-22618), financed by the Spanish Ministry of Education and Science; SERENIDAD (PEII11-037-7035), financed by the "Viceconsejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha" (Spain); and FEDER, SIGMA-CC (TIN2012-36904), and GEODAS (TIN2012-37493-C03-01), financed by the "Ministerio de Economía y Competitividad" (Spain).

## REFERENCES

1. KPMG. *The e-Crime Report 2011: Managing Risk in a Changing Business and Technology Environment*. e-Crime Congress, 2011.
2. Zhang Y, Xiao Y, Ghaboosi K, Zhang J, Deng H. A survey of cyber crimes. *Security and Communication Networks Journal, Wiley* 2012; **5**(4):422–437.
3. IC3. 2010 Internet crime report. Internet Crime Complaint Center, 2010.
4. Arconati N. *One Approach to Enterprise Security Architecture*. SANS Institute: USA, 2002.
5. Fernandez EB, Larrondo-Petrie MM, Sorgente T, Vanhilst M. A methodology to develop secure systems using patterns. In *Integrating Security and Software Engineering: Advances and Future Vision*. IGI Global: Pennsylvania, USA, 2006; 107–126.
6. Georg G, Ray I, Anastasakis K. An aspect-oriented methodology for designing secure applications. *Information and Software Technology* 2009; **51** (5):846–864.
7. Mouratidis H. Secure software systems engineering: the Secure Tropos approach. *Journal of Software* 2011; **6**(3):331–339.
8. Schmidt H, Hatebur D, Heisel M. A pattern-based method to develop secure software. In *Software Engineering for Secure Systems: Industrial and Research Perspectives*. IGI Global: Pennsylvania, USA, 2011.
9. Alexander C, Ishikawa S, Silverstein M. *A Pattern Language: Towns, Buildings, Constructions*. Oxford University Press: Oxford, United Kingdom, 1977.
10. Fernandez EB, Washizaki H, Yoshioka N, Kubo A, Fukazawa Y. Classifying security patterns. In *Asia-Pacific Web Conference*, 2008; 342–347.
11. Pelaez J, Fernandez EB, Larrondo-Petrie MM. Misuse patterns in VoIP. *Security and Communication Networks Journal, Wiley* 2009; **2**(6):635–653.
12. Moral-García S, Ortiz R, Moral-Rubio S, Vela B, Garzás J, Fernández-Medina E. A new pattern template to support the design of security architectures. In *The International Conferences on Pervasive Patterns and Applications. PATTERNS*, 2010; 66–71.
13. Romanosky S. Enterprise security patterns. *Information Systems Security Association Journal* 2003.
14. Moral-García S, Moral-Rubio S, Fernández EB, Fernández-Medina E. A New enterprise security pattern: secure software as a service (SaaS). In *9th International Workshop on Security in Information Systems (WOSIS)*. Wroclaw (Poland), 2012; 14–26.
15. AGCS. AG Communication Systems Template, *The Patterns Handbook: Techniques, Strategies, and Applications*. edited by Linda Rising, pp. 85, Cambridge University Press: NY, 1998.
16. Gamma E, Helm R, Johnson R, Vlissides J (eds). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley: Boston, USA, 1994.
17. Fernandez EB, Mujica S, Valenzuela F. Two security patterns: least privilege and security logger/auditor. In *Asian PLoP*. 2nd Asian Conference on Pattern Languages of Programs (Asian PLoP 2011): Tokyo, Japan, 2011.

18. Kienzle DM, Elder MC, Tyree D, Edwards-Hewitt J. *Security Patterns Repository Version 1.0*. Available from: www.scrypt.net, 2002.
19. Sherwood J, Clark A, Lynas D. Enterprise security architecture. SABSA White Paper, 2009; 25.
20. IBM. *Enterprise Security Architecture. Using IBM Tivoli Security Solutions*. International Technical Support Organization, IBM Redbooks: New York, USA, 2007.
21. Wood CC. *Information Security Policies Made Easy*. Version 7, Baseline Software: New York, USA, 2000.
22. Fernandez EB, Gudes E, Olivier M. Policies and models. In *The Design of Secure Systems*. Under contract with Addison-Wesley: Boston, USA.
23. Lankhorst M. *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer: New York, USA, 2009.
24. Buschmann F, Meunier R, Rohnert H, Sommerlad P, Stal M. *Pattern-Oriented Software Architecture: A System of Patterns*. Wiley: New Jersey, USA, 1996.
25. OSF. DATALOSS db - open security foundation. Available from: http://datalossdb.org/ [retrieved on October, 2012].
26. The Object Management Group (OMG). *Model-Driven Architecture Guide Version 1.0.1*. The Object Management Group (OMG): Massachusetts, USA, 2003.
27. Harmon P. *The OMG's Model Driven Architecture and BPM*. Newsletter of Business Process Trends, 2004; **2**(5).
28. Meservy TO, Fenstermacher KD. Transforming software development: an MDA road map. *Computer* 2005; **9**:52–58.
29. Schumacher M, Fernandez-Buglioni E, Hybertson D, Buschmann F, Sommerlad P. *Security Patterns: Integrating Security and Systems Engineering*. Wiley: New Jersey, USA., 2006.
30. Fernandez EB. *Security Patterns in Practice: Building Secure Architectures Using Software Patterns*. To appear in the Wiley Series on Software Design Patterns: Under contract with J. Wiley: New Jersey, USA.
31. ISO. International organization for standardization. Available from: http://www.iso.org [retrieved on October, 2012].
32. BSI. *IT Baseline Protection Manual*. Federal Agency for Security in Information Technology: Germany, 2000.
33. Poulsen K. 7-eleven hack from Russia Led to ATM looting in New York. Available from: http://www.wired.com/threatlevel/2009/12/seven-eleven/ [retrieved on October, 2012].
34. Lennon M. Massive breach at epsilon compromises customer lists of major brands. Available from: http://www.securityweek.com/massive-breach-epsilon-compromises-customer-lists-major-brands [retrieved on October, 2012].
35. Fernandez EB, Sorgente T. A pattern language for secure operating system architectures. In Proceedings *of the Latin American PLoP*. : Brazil, 2005; 68–88.
36. Fernandez EB, Larrondo-Petrie MM. Securing design patterns for distributed systems. In *Security in Distributed, Grid,and Pervasive Computing*. CRC Press: United States, 2007; 53–66.
37. Jackson M. *Problem Frames: Analyzing and Structuring Software Development Problems*. Addison-Wesley: Boston, USA, 2001.