Quest Journals
Journal of Software Engineering and Simulation
Volume 7 ~ Issue 9 (2021) pp: 25-37
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

Development of Security Model for an Online Transaction ProcessingSystem Using Mandatory Access Control and OOHDM

[1]AMANZE, BETHRAN. C., [1]AGBASONU VAL. C &[1]AGBAKWURU .A. ONYEKACHI
[1]*Department of Computer Science, Imo State University, Owerri, Nigeria.*

ABSTRACT
The motivation of this paper is due to an attacker to access data that are contrary to the specified access restrictions for that data, an attackers to execute commands as another user and an attacker to conduct a denial of service. The aim of this paper is to develop a security model for an online transaction processing system using MAC mechanism. The objective of this paper is to reduce the growing cross channel fraud, data theft, and ransom ware on e- commerce platforms. Mandatory Access Control (MAC) mechanism was used to enhance the security of the sensitive information/data shared during the online transaction processing. The methodology adopted was Object Oriented Hypermedia and Design Methodology (OOHDM) which is well suited for analyzing and designing objects that make up the new security enhanced system. Microsoft Visual Studio 2010 was used as our development environment. The programming language used was PHP and Java Script, while MySQL Server 2008 was used in the development of the database engine.
KEYWORD: MAC, OOHDM, RBAC, UML

Received 20 September, 2021; Revised: 03 October, 2021; Accepted 05 October, 2021 © The author(s) 2021. Published with open access at www.questjournals.org

I. INTRODUCTION

Many operating systems in use today presents some level of security to protect users, processes resources etc. from various forms of attacks. Although some of these operating system may not be able to defend completely against both internal and external threat as many of them are not well equipped with the necessary tools to monitor and control all aspect of their operations. Linux employs the Discretionary Access Control (DAC) Mechanism to control access to the system. The DAC mechanism is referred to as discretionary because the control of access is based on the owner's discretion. The owner of the object specifies which subjects can access the object (Franco *et al.*, 2008).Most operating systems such as Windows, Linux, and Macintosh and most flavors of Unix are based on the DAC models. In these operating systems, when a file is created, the owner of the file decide what access privileges to give to other users; when they access the file, the operating system will make the access control decisionbased on the access privileges the file owner created (Franco *et al*., 2008).DAC access decisions are only based on user identity and ownership, ignoring other security-relevant information such as the role of the user, the function and trustworthiness of the program or thesensitivity and integrity of the data (Ommeren *et al*., 2014). Many system services and privileged programs run with coarse-grained privileges that far exceed their requirements, so that a flaw in any one of these programs can be exploited to obtain complete system access (Pelc, 2015).There have been a high rate of computer crimes in recent years, every day, we hear of sad news of how organizations are hacked, both big and small organizations alike, we hear of millions of pounds lost to cybercrimes, we also hear of how criminals compromise organizational data either for financial gain, or to prove of how advanced they are in hackertivism, most times, the affected organizations do not disclose such attacks to the public probably for fear of loss of public trust, most time the affected organization do not really recover from the effect of these malicious acts and soon goes bankrupt and subsequently fold their business.

In fact, many small organizations across the world have fallen victim to these and were forced out of business. Making these issues worst is that it is often difficult to trace culprits as they may be tons of miles away, while penetrating peoples network remotely, often times, users do not help the situation. Human beings have been identified as the greatest asset of any organization, and being the greatest assets, they are also the greatest threat. User activities often time enable hackers gain cheap access into the system. Just in the past year (2018), we heard about the boss who granted interview to the press with his password written on postit note and pasted on the wall just behind him in front of the cameras, staff often time due to lack of training and proper awareness are social engineered into divulging sensitive organizational information.All these and many more stimulated the desire in me to design a context-aware system which can take autonomous decisions based on certain environmental factors even without the help of a system administrator who are often cumbered with a load of task of ensuring adequate security in organizations, these administrators are often time engage with securing what organizations termed 'critical' while paying little or no attention to the 'little' things like patching the system, monitoring users activities etc., thereby hackers exploit the loop hole created by these 'little' things to cause catastrophic damage to the organization. Hence, this work will develop trust in e-commerce and online transaction space by allaying the fear of users/customers who are wary of online platforms due to the high rate of cyber-crime. This research work seeks to significantly reduce electronic fraud and encourage the evolving cash-less policy of the Central Bank of Nigeria (CBN). It shall handle and/or checkmate cross channel fraud in e-commerce platforms by building a system that take autonomous decisions based on certain policies and/or rules in Windows operating system. It will not solve all cybercrime issues but it will make users/customers feel protected while purchasing goods and services on e-commerce platforms.

II. REVIEW OF RELATED WORKS

Network security is a systems engineering discipline. The design, functionality, and operational performance of computer architecture with network security are a field that requires a systems perspective and approach. The computer architectures, operating system and platforms need to be thoroughly understood before a layered approach to design is undertaken.Security must be engineered into every aspect of the network design. In order to construct sound network security choices, the kind and type of data to be protected needs to be evaluated for the level of security desired. Government and defense related industries need the most robust cryptography and granular operational appliances that would obviously protect data. A university campus will install protection devices to protect salary and financial records of its employees, but they will not be as secure as patient medical or insurance information located in a hospital computer system. Knowledge of the business environment and where the computer security will be used are items that need to be determined as part of the design process (Bayuk *et al*., 2012).

In addition, the failure of one security appliance should not compromise the entire system. The concept of redundancy and failsafe apparatus must be included in any network security design. There are components in any internal network that have security mechanisms as part of the device. The decision to use integrated functions on a network device as opposed to specialized network appliances need to be made. The weaknesses and strengths of each integrated security appliances should be known and balanced with the success rates of protecting the storage of data (Bottino & Hughes, 2015).Since every network device can be considered a potential target, the danger posed by different vulnerabilities and exploits should be examined. Vulnerabilities can be classified in terms of those that attack hosts, systems, resources and protocols. Assessing the threats from these vulnerabilities is important in that only the proper combination of integrated security appliances will prevent dangerous compromises.

In addition, each security vulnerability does not carry the same risk and the risk associated with vulnerability does change over time (Bottino & Hughes, 2015).Another design consideration that can contribute to secure construction of the computer architecture is resource separation. Different security technologies can be combined to separate resources on various defensive layers. These defensive layers all share the principle of security zones. Within these zones logical groupings of systems, networks or processes are collected that have the same degree of acceptable risk (Bayuk *et.al*, 2012).The installation of particular security appliances used in different security technologies can influence operational performance and network reliability in an adverse manner. With the addition of certain appliances in any computer architecture, steps can be taken to compensate for performance degradation and elements of unreliability. Firewall architectures, IDS sensors and VPNs with security encryption will require a certain amount of bandwidth and affect overall network bandwidth. Using hardware accelerators to improve performance is one technique that can be used to counteract negative effects on performance, reliability and network latency. This is successfully accomplished since hardware accelerators are designed to perform the mathematical operations that are required by cryptography. Also, cryptographic operations can be performed more efficiently with accelerators than with general purpose computers (Bottino & Hughes, 2015).

III. MANAGEMENT OF A PROTECTED ENVIRONMENT

The management of a protected computer environment entails balancing protection with the containment of integrity. After an evaluation of computer architecture, the formula for secure computer architecture should include the proper combination of security devices and security technologies. Most components of a computer network already have some security mechanisms inherent in the software of the device. As an example, network routers have been designed with many built-in security features such as packet filters, statefull firewalls features and VPN support. However, these security mechanisms alone will never protect data; provide an adequate defense against different malicious threats, or alert network users of reconnaissance. The network security appliances selected can have an operational effect on the performance of the network. Additional bandwidth will be used and throughput will be increased, however with some modifications in network design, losses of data can be prevented and compromises in network performance can be avoided (Geers, 2011).

Amurthy &Redddy, (2012),developed an embedded fingerprint system, which was used for ATM security applications. In their system, bankers collect customers’ finger prints and mobile numbers while opening accounts, then customer only access ATM Machine. The working of the ATM machine is such that when a customer place a finger on the finger print module it automatically generates every time different 4-digit code as a message to the mobile of the authorized customer through GSM modem connected to the microcontroller. The code received by the customer is entered into the ATM machine by pressing the keys on the touch screen. After entering it checks whether it is a valid one or not and allows the customer further access. From our point of view, though the system developed is an aspect of cyber security system but it is limited to ATM machines and does not provide a technique or process that protects a system’s information assets from threats to confidentiality, integrity, or availability.Over the last few years, researchers have carried out researches to identify the best approach to providing adequate security of information systems; this follows the increase in the number of attempted cyber-attacks. These attacks when successful have caused serious damages to organizational reputations, huge financial loss and breach of public trust. Many obstacles face organizations, agencies, government or even individuals when attempting to achieve maximum security of information systems; among them are the existence of dependence of security systems management on human intervention, which is a continuousprocess that increases the level of difficulty. Another example of obstacles is that attacks on computer/information systems are becoming increasingly sophisticated and there are several deficiencies in current security systems. Thus, the problem of security management is becoming more complex and it is therefore pertinent to use resources offered by policy-based computing. From the forgoing, it is obvious that none of the authors considered developing a system that would resist suspicious activities by making autonomic decision(s) based on in-built policies thereby enhancing the security of the system as well as make the user feel protected. Basically, policy-based systems were designed to support run-time reconfiguration ability of systems decision making logic. Policy-based computing describes a methodology for embedding dynamic behavior into software components and thismakes them more reliable. Thus, an enhanced security model that implements MAC mechanism was proposed.

IV. MATERIALS AND METHODS

The methodology adopted is the object oriented hypermedia and design methodology (OOHDM).In OOHDM, a hypermedia application is built in four step process, supporting an incremental or prototype process model. Each step focuses on a particular design concern, and an Object-Oriented model is built. Classification, aggregation and generation/specification are used throughout the process to enhance abstraction power and reuse opportunities. These phases are summarized below.

1.Requirement Gathering:
In this phase the UML provide some abstraction mechanisms for this purpose, as the Use Case Diagram, but other diagrammatic tools as the Interaction Diagram (Sequence or Collaboration) or the state chart Diagram can also be used to gather requirements even they are usually used only at design time.

2.Domain/Conceptual Analysis:
In this phase a conceptual model of the application domain is built using well-known object-oriented modeling principles augmented with some primitives such as attribute perspective. Conceptual classes may be built using aggregation and generalization/specialization hierarchies.

3.Navigational Design:
Here we describe the navigational structure of a hypermedia application in terms of navigational contexts, which are induce from navigational classes such as nodes, links, indices, and guided tours. Navigational contexts and classes take into account the type of intended users and their access level.

4.Abstract Interface Design:

The abstract interface model is built by defining perceptible objects (e.g., a picture, a city map, and so forth) in terms of interface classes. Interface classes are defined as aggregations of primitive classes (such as text fields and buttons) and recursively of interface classes. Interface objects map to navigational objects, providing a perceptible appearance. Interface behavior is declared by specifying how to handle external and user-generated events and how communication takes place between interface and navigational objects.

5. Implementation:

Implementation maps interface objects to implementation objects and may involve elaborated architecture (e.g., client-server), in which applications are clients to a shared database server containing the conceptual objects.

The OOHDM approach is motivated by the kind of system we desire to develop. We desire to build a usable and evolvable hypermedia application. Good web applications should be good hypermedia applications. The very nature of the proposed system, in which navigation is combined with the inherent difficulties of dealing with multimedia data, needs an OOHDM approach. Traditional methodologies do not contain useful abstractions to deal with the hypertext metaphor. They do not provide the notion of linking. The interface of Web apps is more complex than in traditional software systems, navigation and functionality should be seamlessly integrated and the navigational structure should be decoupled from the domain model of the app, therefore OOHDM was chosen for its functionalities, in that it allows object oriented abstractions for analysis and design of information-intensive web applications. Besides the modeling abstractions, it also provides a methodology which guides a developer through different activities in the web application development.

Use Case Diagram

When creating a use case model, individual functions to be provided by the system isrepresented as use case and the presence outside the system to interact with use case is represented as an actor. Using the mandatory access control (MAC) mechanism, there are four actors in the online transaction processing system:

1. the credit card issuer assigned private role
2. The merchant assigned protected role
3. The customers gets the default role
4. Site visitors gets public role

The figures bellow shows creating a use case model for online transaction processing system.

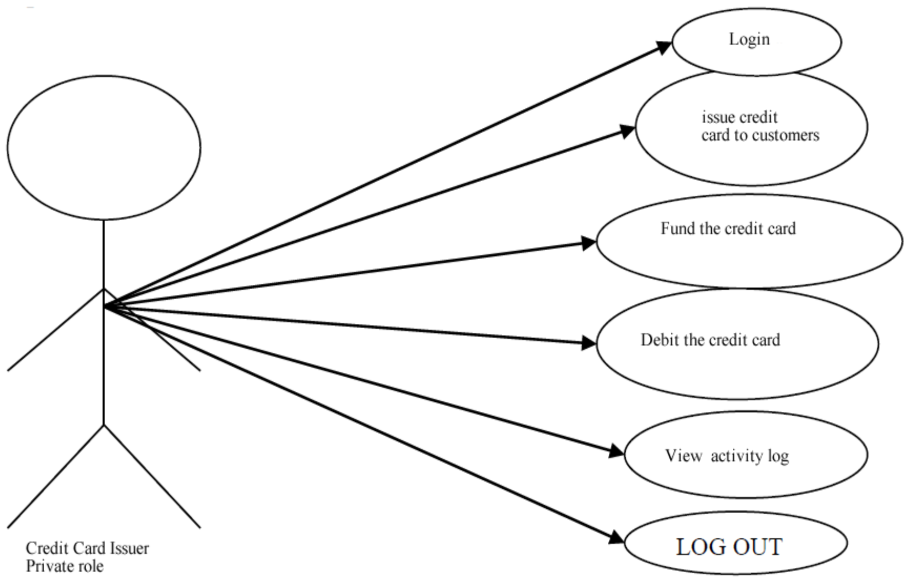

Figure 1: Use Case Boundary of credit card issuer

The use case as shown in figure 1 has private role assigned to the credit card issuer. The access is granted to only the licensed issuing agent who in turns issue credit card to customers, credit or debit the card and also view the activity log of the credit card.

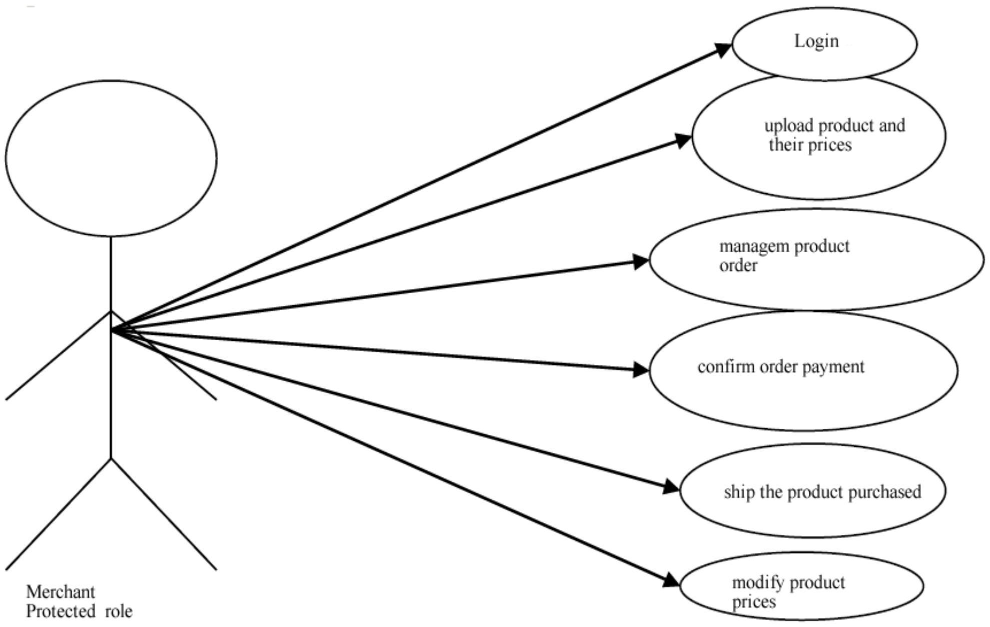

Figure 2: Use Case Boundary of merchant

The merchant use case diagram was assigned protected role as shown in figure 2. The functions includes uploading product prices on the web platform, modifying the prices, managing customers purchase order list, confirmation of payment and delivery of the product ordered.

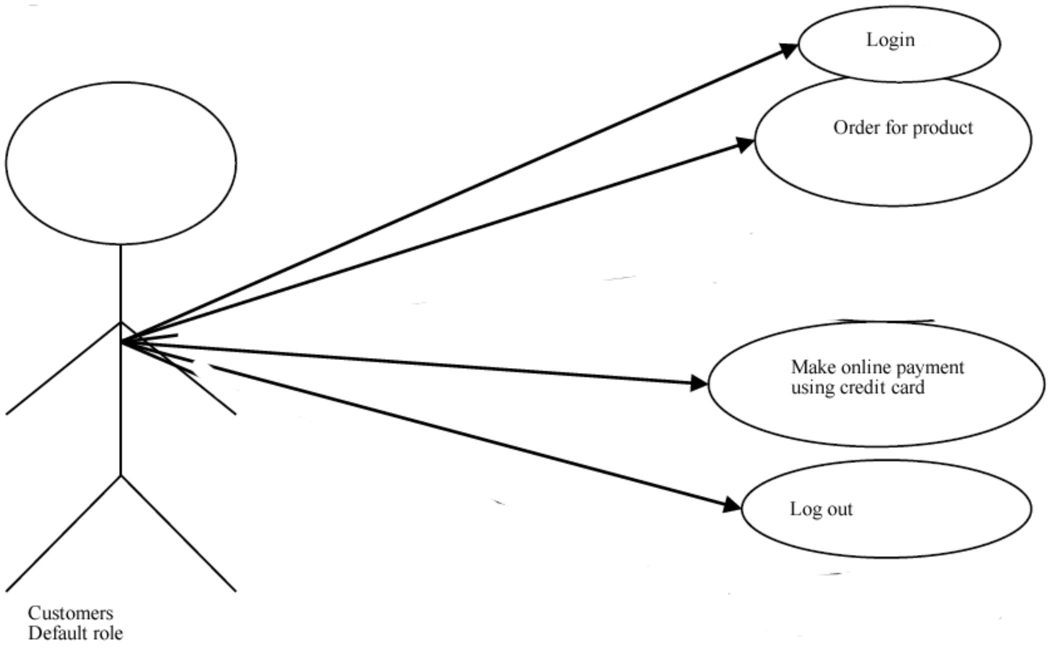

Figure 3: Use Case Boundary of Customers

The customers has default role as shown in figure 3 which enable them log on to the web site, select products for ordering, place and order, make payment for the products.

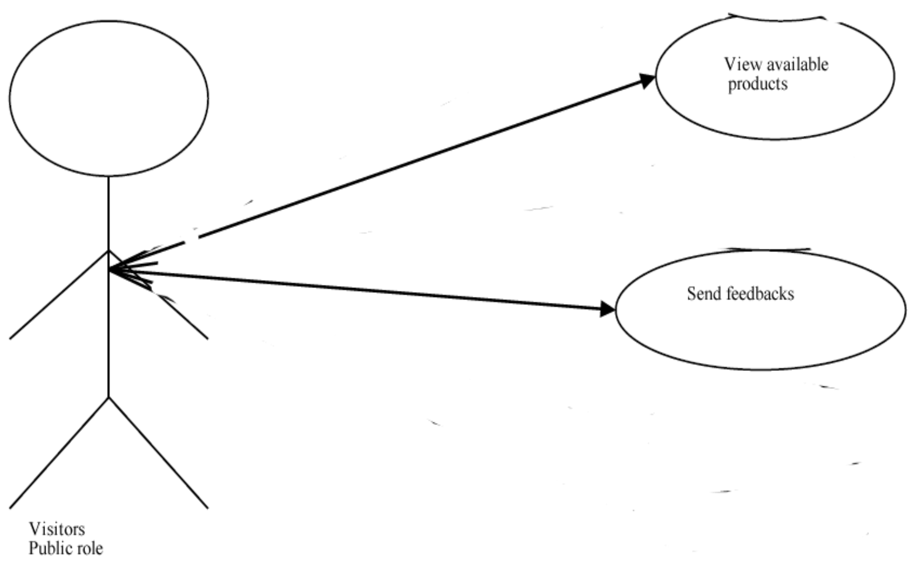

Figure 4: Use Case Boundary of Visitors

As shown in figure 4, they have a public role that can enable them view the products on the website and also send feedback messages.

Class Diagram

Class diagrams are one of the most useful types of diagrams in UML as they clearly map out the structure of a particular system by modeling its classes, attributes, operations, and relationships between objects. The standard class diagram is composed of three sections:

- **Upper section:** Contains the name of the class. This section is always required, whether you are talking about the classifier or an object.
- **Middle section:** Contains the attributes of the class. Use this section to describe the qualities of the class. This is only required when describing a specific instance of a class.
- **Bottom section:** Includes class operations (methods). Displayed in list format, each operation takes up its own line. The operations describe how a class interacts with data.

Customers can select and purchase different products for a web shop. The shopping process musthave well defined steps. This is necessary because we need to show the customer where he is in the process. Figure 5 shows the classdiagram for the Shopping Cart pattern. The ShoppingCart class collects information about all the products a customer has selected. The CartItem object indicates the quantity and the product selected by a customer. A customer can query the products in his cart and remove products from the cart. The Customer class indicates the customer responsible for a shopping cart. When the shopping cart is checked out, an order and an invoice will be generated (the Order and Invoice classes).

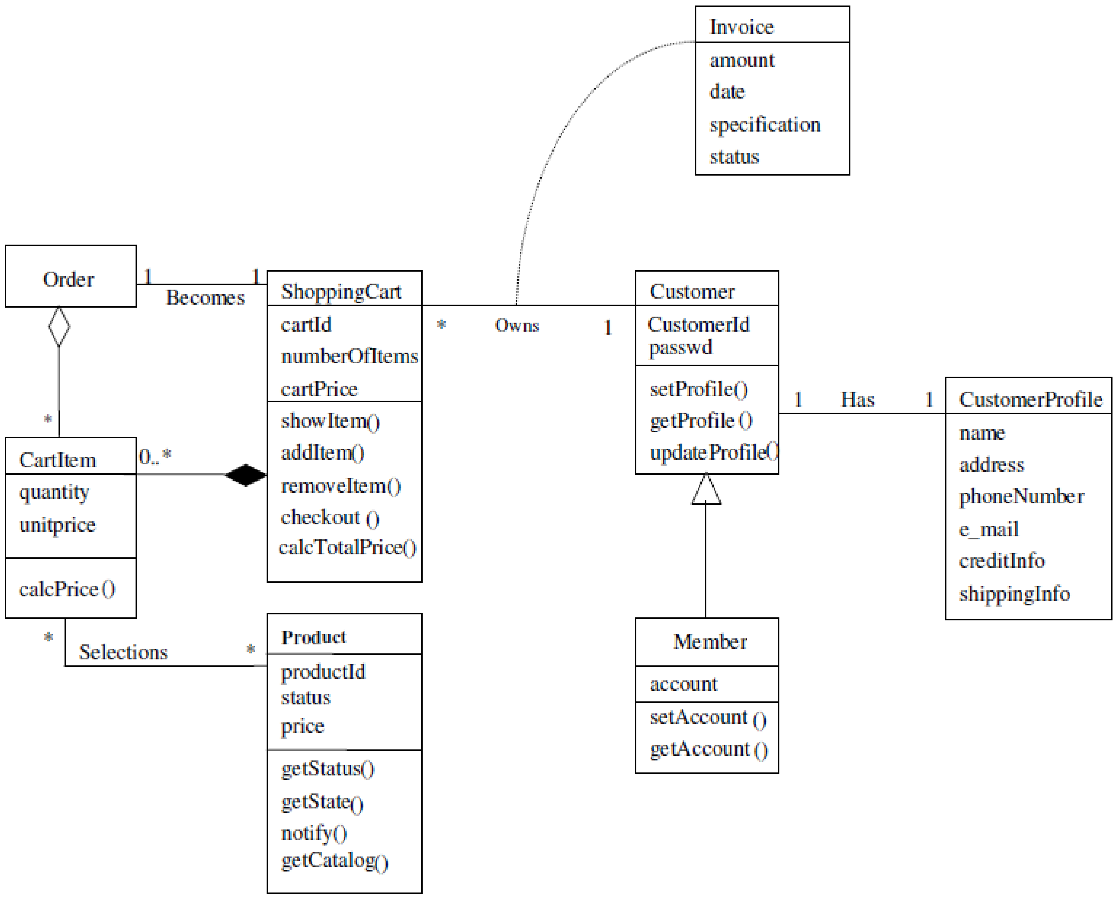

Figure 5: Class diagram for Shopping Cart

There are many systems where we need to combine the functions of creating and preparing aninvoice, and paying that invoice, including the corresponding validations. The solution is affected by the following forces:

- The creation, preparation, and validation of an invoice requires specific actors, actions inspecific sequences, and must follow specific rules.
- Preparation and validation should be done by different people (separation of duty).
- There should be flexibility about who is responsible for a payment and how the payments will be made.
- The system and the client need a convenient way of keeping track of the payments made.
- Validation of every prepared invoice and every received payment has to be made to ensure that the client's information is correct and in accordance to the requirements and regulations of each system.
- We need to keep track of who created an invoice, who validated it, and who validated apayment.

Figure 6 is the class diagram for the Invoice pattern. Class **InvoiceCreator** defines an interfacefor creating an invoice.

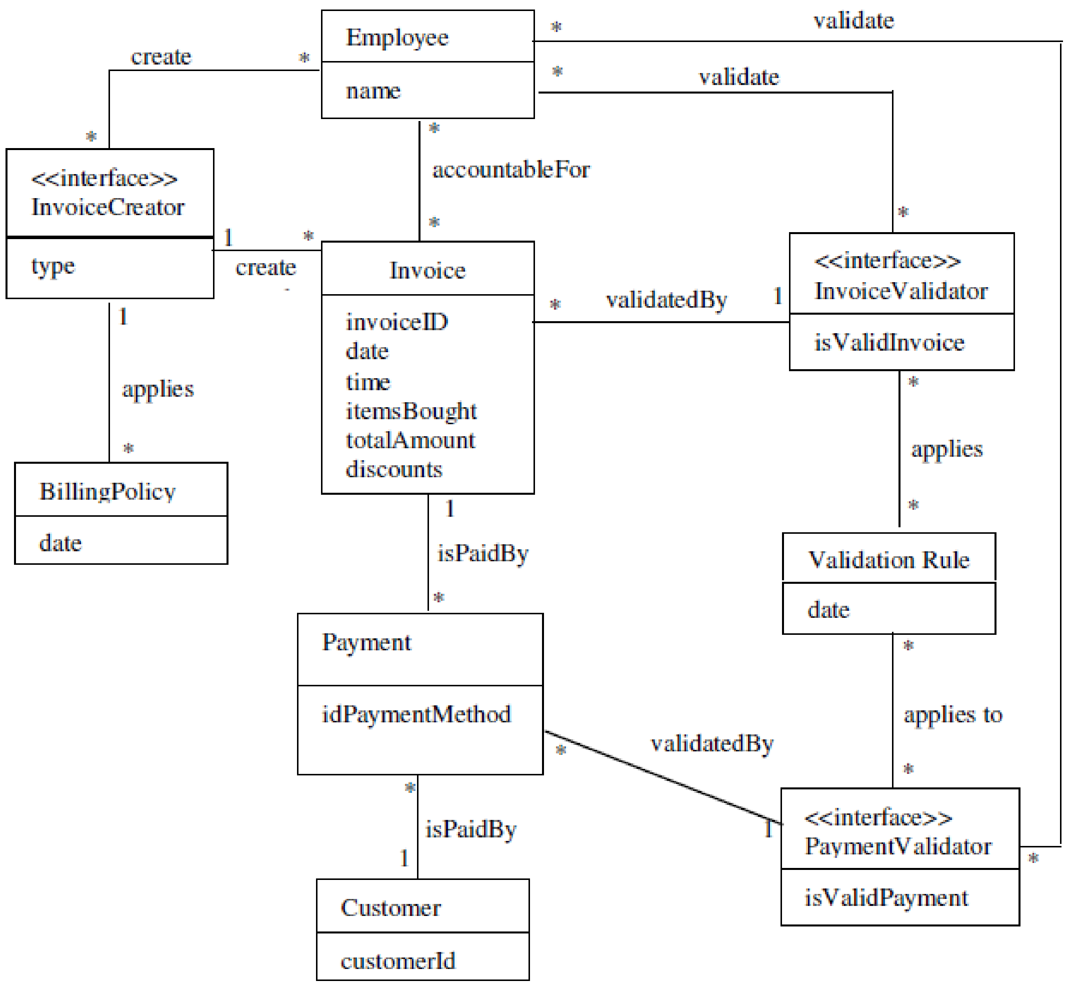

Figure 6: Class diagram for Invoice

It also provides a way of preparing the invoice by adding or deletingitems from it, specifying different properties, which are used to derive the final scope of theinvoice. Class **Invoice** represents the document in which all the goods or services areincorporated together with the nature of each item. Class **InvoiceValidator** is used to ensure thatthe invoice that resulted from the steps described above is in a consistent form that complies withthe trade usage. Classes **BillingPolicy** and **ValidationRule** include business policies that apply tothe preparation and validation of invoices, and validation of payment.The **Payment** classrepresents the payment made by the client for the products and/or services incorporated in theinvoice. Class **PaymentValidator** is used to validate payments according to validation rules.
Class **Employee** keeps track of who validated a payment, and class **Customer** represents thecustomer that makes payments for the given invoice.

Sequence Diagram
Figure 7 shows the sequence diagram for ordering for product. Figure 10 shows the sequence diagram for creating, preparing and validating an invoice. Figure 7 shows the sequences diagram for payment of an invoice.

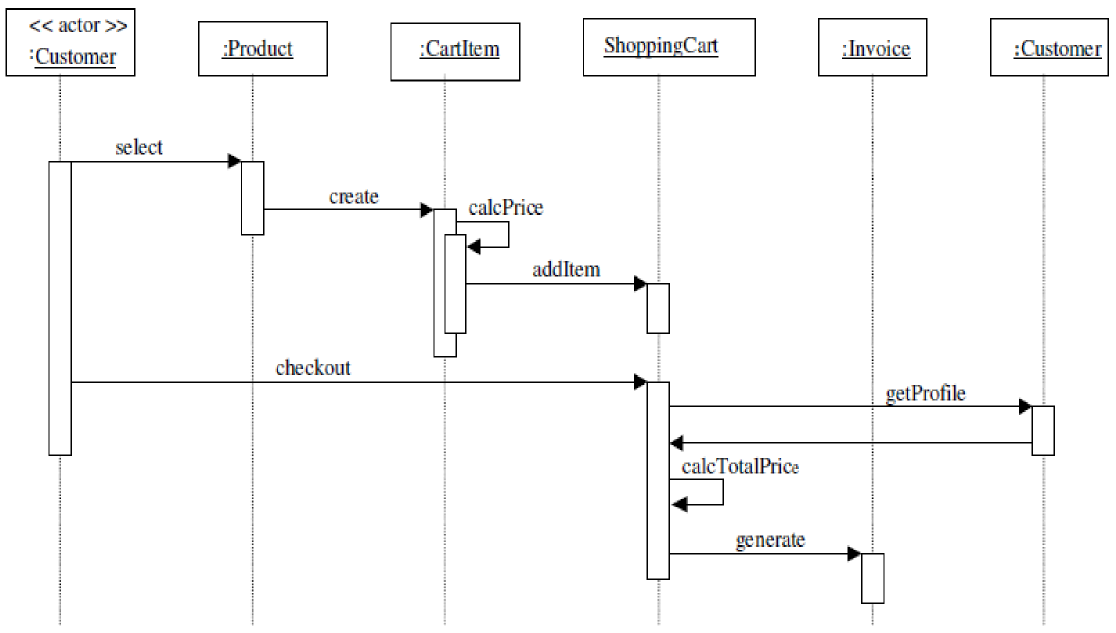

Figure 8: Sequence diagram for buying and checking out a product

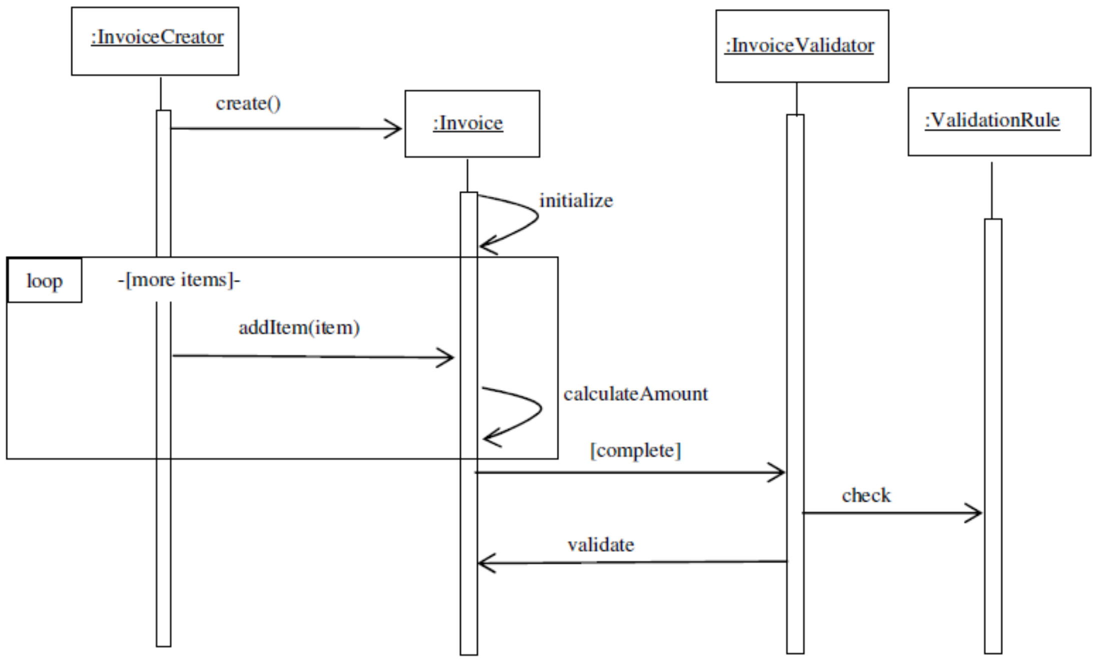

Figure 9: Sequence diagram for creating, preparing and validating an invoice

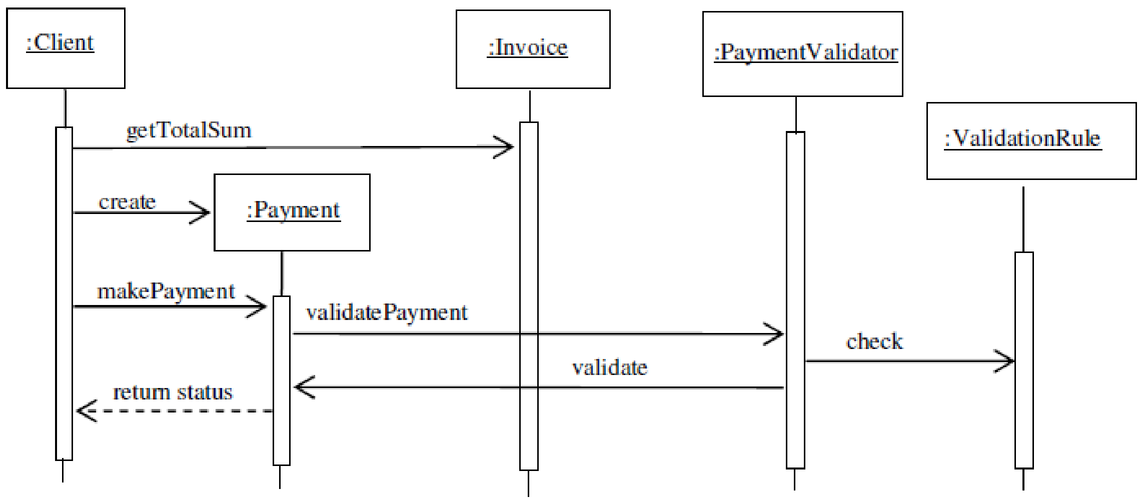

Figure 10: Sequence diagram for invoice payment

Activity Diagram

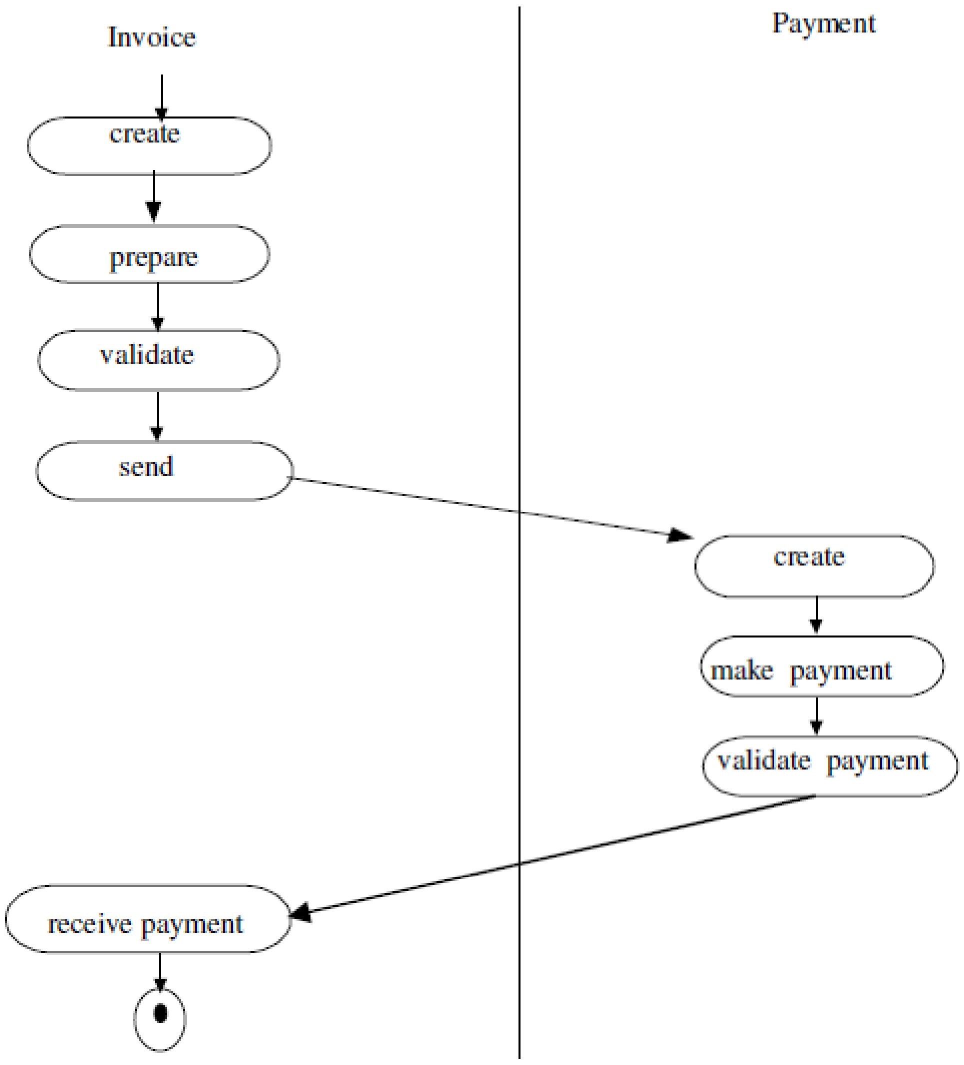

Figure 11: Activity diagram for invoice creation and payment

This pattern describes an abstract invoice preparation process that can be tailored todifferent specific situations. We can keep track of who prepared and who validated an invoice, as well as whovalidated payments. We can apply different validation rules to an invoice and a payment.

Secure Online Transaction Processing Collaboration Diagram

Security constraints can be added to each of the component patterns to produce a domain modelfor secure e-commerce. We demonstrate here how to add security constraints by instantiating asecurity pattern, that is, Role-Based Access Control (RBAC) pattern. In the RBAC pattern,users are assigned to the roles according to their tasks or jobs and rights are assigned to the roles.

In this way, a need-to-know policy can be applied, where roles get only the rights they need toperform their tasks. Figure 13 shows how to add security constraints to the Shopping Cart patternby applying instances of the RBAC pattern.

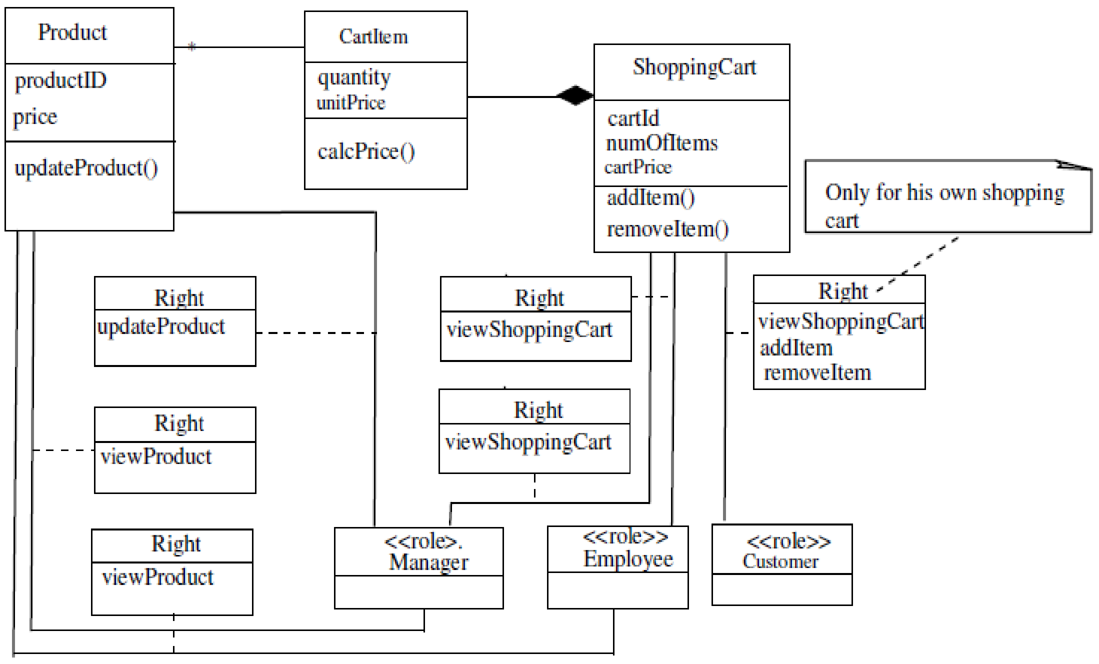

Figure 12: Adding security constraint to Shopping Cart

5.0 Analysis of the New Model Architecture

The new model, as illustrated in Figure 13, is described as follows: threats target electronic transaction processing systems (assets). The threat actor(s) gain access to the assets via attack vectors and vulnerabilities present in the technology components that house or provide direct access to the targeted assets. Security controls are applied to the technology components with the intent to counter or mitigate the vulnerabilities and/or attack vectors used by the threat actors, thereby protecting the assets.

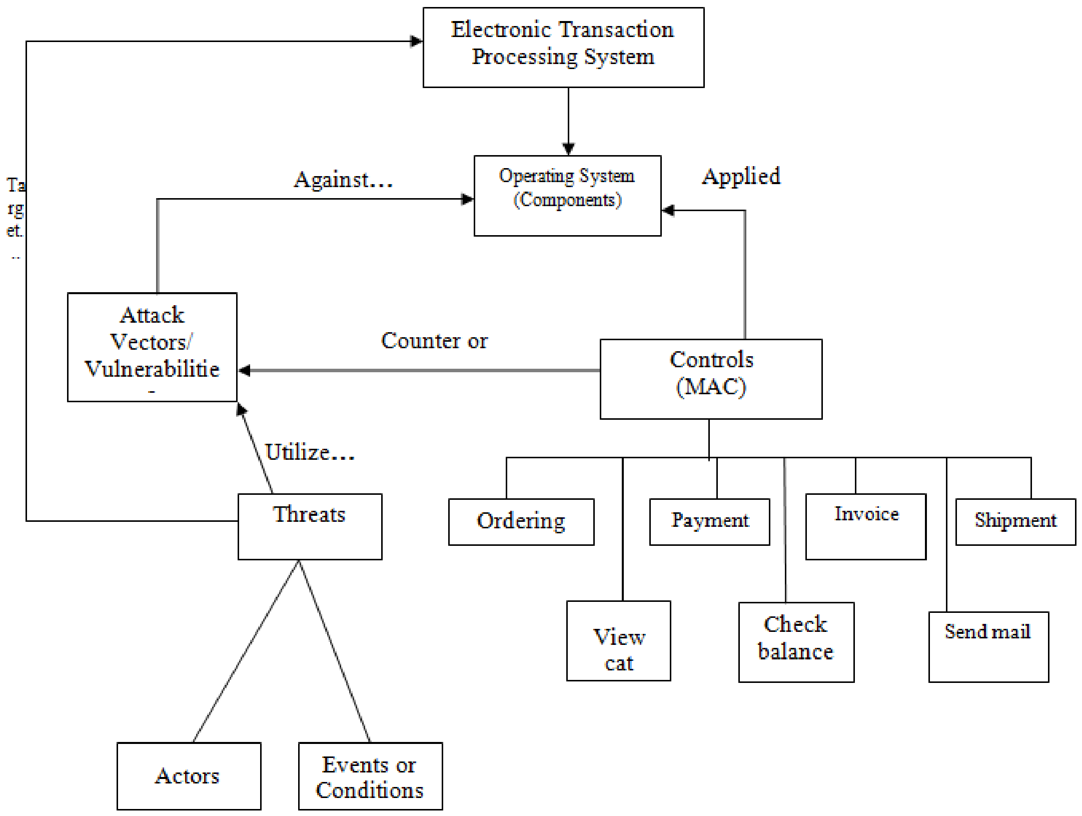

Figure 13: New Model Architecture

V. CONCLUSION

This research presents a set of patterns appropriate for securing information in an online transaction processing system using OOHDM. The research work also shows how security constraintscan be added to the domain model by using mandatory access control (MAC). The system developed created three different access levels which include e-platform administrator, the payment gateway administrator and the customers. Each of the users on the platform has limited access areas and this was achieved by applying mandatory access control technique. Security constraints were added to each of the component patterns to produce a domain modelfor secure e-commerce. We demonstrate here how to add security constraints by instantiating asecurity pattern, that is, Role-Based Access Control (RBAC) pattern. In the RBAC pattern,users are assigned to the roles according to their tasks or jobs and rights are assigned to the roles.In this way, a need-to-know policy can be applied, where roles get only the rights they need toperform their tasks. The software developed utilized the mandatory access control (MAC) as a security mechanism for the online transaction processing which involves product ordering, payment using credit card, and product information management. The system is very robust and MySQL database was used at the back-end.

REFERENCES

[1]. Amurthy, P. K. & Redddy, M. S. (2012). Implementation of ATM Security by Using Fingerprint Recognition and GSM. International Journal of Electronics Communication and Computer Engineering, 3 (1), 83-86.

[2]. Bayuk, J. L.; Healey, J.; Rohmeyer, P.; Sachs, M. H.; Schmidt, J. & Weiss, J. (2012). Cyber Security Policy Guidebook. New Jersey: John Wiley & Sons, Inc., 1056-1088.

[3]. Bottino, J. & Hughes, V. (2015). Understanding and Managing Cybercrime. Boston: Allyn & Bacon, 202-244.

[4]. Chen, D.; Cong, J.; Gurumani, S.; Hwu, W.; Rupnow, K. & Zhang, Z. (2016). Cyber-Physical Systems: Theory & Applications. Journal of the Institution of Engineering and Technology, 1 (1), 70-77.

[5]. Franco, L.; Sahama, T. & Croll, P. (2008). Security Enhanced Linux to Enforce Mandatory Access Control in Health Information Systems. Proceeding of 2nd Australasian Workshop on Health Data and Knowledge Management (HDKM 2008), Wollongong, Australia, 144-250.

[6]. Fulghum, D. A.; Wall, R. & Butter, A. (2007). Cyber-Combat's First Shot. Aviation Week & Space Technology, 167 (21), 28.

[7]. Geers, K. (2010). Live Fire Exercise: Preparing for Cyber War. Journal of Homeland Security and Emergency Management, 7 (1), 1-16.

[8]. Geers, K. (2011). From Cambridge to Lisbon: the quest for strategic cyber defense. Journal of Homeland Security and Emergency Management, 8 (1), 1-16.

[9]. Geers, K. (2011). Strategic Cyber Security. Estonia: CCDCOE Publication, 106-132.

[10]. Gercke, V. (2012). Cybercrime & Cybercriminals: An Overview. Estonia: CCDCOE Publication, 254-258.

[11]. Oltramari, A.; Cranor, L. F.; Walls, R. J. & McDaniel, P. (2016). Building an Ontology of Cyber Security. International Symposium on Information, Computer, and Communications Security, 1(1), 54-61.
[12]. Ommeren, E. V.; Borrett, M. & Kuivenhoven, M. (2014). Staying Ahead in the Cyber Security Game: What Matters Now. California: Sogeti and IBM, 21-69.
[13]. Onyesolu, M. O. & Ezeani, M. I. (2012). ATM Security Using Fingerprint Biometric Identifier: An Investigative Study. International Journal of Advanced Computer Science and Applications, 3 (5), 67-74.
[14]. Pegueros, V. (2012). Security of Mobil Banking and Payments. SAN Institute Info Sec Reading Room, 248-279.