

Modeling and Verification of Online Shopping Business Processes by Considering Malicious Behavior Patterns

G. Raja Gopal Reddy¹, Dr. M. Sree Devi ²

¹Student, Dept. of MCA, SVU.College of C M & C's

²Professor, MCA,M.Phil,PhD, Dept. of MCA, SVU.College of C M & C's, Tirupati,A.P.

Abstract- Recently, online shopping integrating third-party payment platforms (TPPs) introduces new security challenges due to complex interactions between Application Programming Interfaces (APIs) of Merchants and TPPs. Malicious clients may exploit security vulnerabilities by calling APIs in an arbitrary order or playing various roles. To deal with the security issue in the early stages of system development, this paper presents a formal method for modeling and verification of online shopping business processes with malicious behavior patterns considered based on Petri nets. We propose a formal model called E-commerce Business Process Net to model a normal online shopping business process that represent intended functions, and malicious behavior patterns representing a potential attack that violates the security goals at the requirement analysis phase. Then, we synthesize the normal business process and malicious behavior patterns by an incremental modeling method. According to the synthetic model, we process design time and, thus, reduces the difficulty and cost of modification for imperfect systems at the release phase. We demonstrate our approach through a case study.

Index Terms- Modeling and Verification of Online Shopping Business Processes by Considering Malicious Behavior Patterns, dynamic programming.

I.INTRODUCTION

1. ORGANISATION PROFILE

SkdotCom Technologies has a rich background in Software Development and continues its entire attention on achieving transcending excellence in the Development and Maintenance of Software Projects and Products in Many Areas. Some of them are ERP, Banking, Manufacturing, Insurance and Laying Emphasis on the Multimedia Projects. These projects are prevalent and have been distributed and implemented for clients' world over. Major software

Development Park at Kodambakkam, Chennai, India.

After stabilizing the skdotCom Products, Mr. P. Siva Kumar M.C.A started giving Counseling Services in the name of skdotCom Agency, privately handled several Projects for leading Companies like Alstom, MRL and Spencer's. DSEL has confronted challenges and rooted itself has a niche player in the Multimedia and the Business Software Segment. The Level of performed has been exemplary leaving it to nothing than the best of benchmark. The indispensable factors, which give DSEL the competitive advantages over others in the market, may be slated as:.

II. PROBLEM DEFINITION

2.1. Existing System:

However, most of the existing work concentrates on the soundness and correctness of workflow, cooperative systems, and composition of web services, but fail to consider malicious behaviors related to the financial security issues. These security issues may result in the financial loss of legitimate users. Many accidents of existing online shopping systems are caused by data errors and state inconsistency as exploited by malicious users. Thus, both data properties and data state non determinacy must be depicted. At the requirement analysis and design levels, one can identify how the software can be attacked by malicious users. According to this idea, misuse or abuse cases and threat modeling are studied.

2.2. Disadvantages:

1. No security provides.
2. TPP verification is not possible.

2.3.Proposed System:

Today's businesses are inherently process-driven, and the security of business processes is increasingly

important. At the source code level, a method for statically checking the security and conformance of the system implementation is proposed. In order to rapidly implement new processes, research on the compliance of cross-organizational processes and their changes is performed. Most of them focus on the security properties like Access Control and Confidential Information in enterprise business processes, and ensure the security of secret and sensible information that cannot be leaked to other parties. Other related studies refer to the process consistence in complex business processes. They are proposed to deal with the inconsistencies among business processes of different departments in a cross-organizational process, or the inconsistencies between real process executions and their designed model, and guarantee the consistence of these business processes when some have changed.

III. FEASABILITY STUDY

FEASIBILITY STUDY:

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ OPERATIONAL FEASIBILITY

3.1. ECONOMICAL FEASIBILITY:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.2. TECHNICAL FEASIBILITY:

This study is carried out to check the technical feasibility, that is, the technical

requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.3. OPERATIONAL FEASIBILITY:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

IV. SYSTEM ANALYSIS

4.1. UML DIAGRAMS:

Unified Modeling Language:

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

➤ User Model View:

- i. This view represents the system from the users perspective.
- ii. The analysis representation describes a usage scenario from the end-users perspective.

➤ Structural model view:

- i. In this model the data and functionality are arrived from inside the system.
- ii. This model view models the static structures.

➤ Behavioral Model View:

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various

structural elements described in the user model and structural model view.

- **Implementation Model View:**
In this the structural and behavioral as parts of the system are represented as they are to be built.
- **Environmental Model View:**
In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the system.
- UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

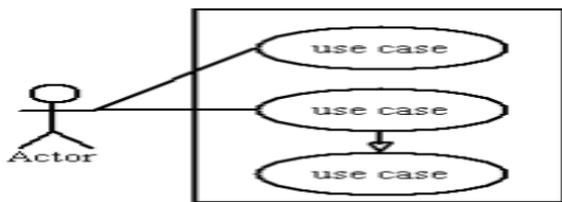
Use case Diagrams represent the functionality of the system from a user's point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer ...etc., or another system like central database.

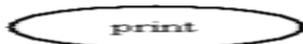
4.2. Use Case Diagram:

Use case diagrams model the functionality of a system using actors and use cases. Use cases are services or functions provided by the system to its users. Basic use case diagram symbols and notations.

System: Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.

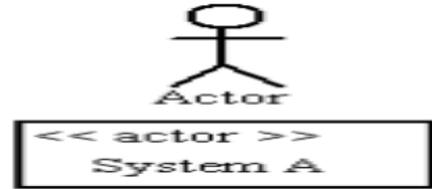


Use Case: Draw use cases using ovals. Label with ovals with verbs that represent the system's functions.



Actors: Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.

V.



SYSTEM DESIGN

INTRODUCTION:

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap and synergy with the disciplines of systems analysis, systems architecture and systems engineering.

5.1. DATABASE DESIGN:

Register Table:

Column Name	Data Type	Size
User id	Number	250
User Name	Text	150
Password	Text	250
Confirm Password	Text	150
Mobile No	Number	250
Email Id	Text	150

Login Table:

Column Name	Data Type	Size
User Id	Number	250
Password	Text	150

Add Product Table:

Column Name	Data Type	Size
Product id	Number	250
Product Name	Text	150
Product Category	Text	250
Product Price	Number	150
Shipping Date	Date Time	250
Delivery Date	Date Time	150

Buying Details Table:

Column Name	Data Type	Size
Buying id	Number	250

Card Number	Number	150
Bank Name	Text	250
Price	Number	150
User Id	Number	250
Name	Text	150
Status	Text	150

Cart Details Table:

Column Name	Data Type	Size
Cart id	Number	250
Product Name	Text	150
Product Category	Text	250
Product Price	Number	150
Product Image	Text	250
Cancel Product	Text	150
Purchase Product	Text	150

interoperate. There’s no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

“NET” is also the collective name given to various software components built upon the .NET platform. These may be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

THE .NET FRAMEWORK:

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).
2. A hierarchical set of class libraries.

The CLR is described as the “execution engine” of .NET. It provides the environment within which programs run. The most important features are:

- ◆ Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
- ◆ Memory management, notably including garbage collection.
- ◆ Checking and enforcing security restrictions on the running code.
- ◆ Loading and executing programs, with version control and other such features.
- ◆ The following features of the .NET framework are also worth description:

MANAGED CODE:

The code that targets .NET, and which contains certain extra Information - “metadata” - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

5.2. COMPLETE CLASS DIAGRAM:

1. Apply design axioms to design classes, their attributes, methods, associations, structures and protocols.

1.1 Refine and complete the static UML class diagram by adding details to that diagram.

- 1.1.1 Refine attributes.
- 1.1.2 Design methods and the protocols by utilizing a UML activity diagram.
- 1.1.3 Refine the associations between classes.
- 1.1.4 Refine the class hierarchy and design with inheritance.

1.2 Iterate and refine.

In designing methods or attributes for the classes, we are confronted with two problems. One is the protocol, or the interface to the class operations and its visibility and

Another is how to implement it.

VI. SOFTWARE ENVIRONMENT

FEATURES OF .NET:

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely

MANAGED DATA:

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you're using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn't get garbage collected but instead is looked after by unmanaged code.

COMMON TYPE SYSTEM:

The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn't attempt to access memory that hasn't been allocated to it.

COMMON LANGUAGE SPECIFICATION:

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

THE CLASS LIBRARY:

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary.

The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O,

threading, and so on, as well as XML and database connectivity.

The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

LANGUAGES SUPPORTED BY .NET:

The multi-language capability of the .NET Framework and Visual Studio .NET enables developers to use their existing programming skills to build all types of applications and XML Web services. The .NET framework supports new versions of Microsoft's old favorites Visual Basic and C++ (as VB.NET and Managed C++), but there are also a number of new additions to the family.

Visual Basic .NET has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, interfaces, and overloading, among others. Visual Basic also now supports structured exception handling, custom attributes and also supports multi-threading.

Visual Basic .NET is also CLS compliant, which means that any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET.

Managed Extensions for C++ and attributed programming are just some of the enhancements made to the C++ language. Managed Extensions simplify the task of migrating existing C++ applications to the new .NET Framework.

C# is Microsoft's new language. It's a C-style language that is essentially "C++ for Rapid Application Development". Unlike other languages, its specification is just the grammar of the language. It has no standard library of its own, and instead has been designed with the intention of using the .NET libraries as its own.

Microsoft Visual J# .NET provides the easiest transition for Java-language developers into the world of XML Web Services and dramatically improves the interoperability of Java-language programs with existing software written in a variety of other programming languages.

Active State has created Visual Perl and Visual Python, which enable .NET-aware applications to be built in either Perl or Python. Both products can be integrated into the Visual Studio .NET environment.

Visual Perl includes support for Active State's Perl Dev Kit.

VII. CONCLUSION

This work's contribution is a systematic approach for modeling and verification of online shopping business processes against some specific malicious behavior patterns. The modeling process is done in a step-by-step manner, and the Composed. EBPN by considering the malicious behavior pattern is built through composing function and malicious behavior models. By analyzing the models through two special procedures, we can verify whether such a process is resistant to some specific malicious behaviors. Through a case study, we illustrate how to model and verify an online shopping system at the design level. The proposed methodology can also be used in other online shopping business processes and malicious behavior patterns that have three parties through defining different business processes and data sets. It is a basic and generic method of modeling and analyzing systems. In fact, there may be another party who participates in an online shopping process sometimes, e.g., banks, but in the trading process, it is common that users transfer a sum of money from bank to TPP such as Paypal or Alipay and then pay for goods with the money in their account in TPP in a relatively long time, and do not use the bank transfer. Therefore, the opportunities for banks to involve in a trading process are limited. In the cases, there are hardly any other parties involved. In addition, the online shopping process with another party (except Shopper, Merchant, and TPP) is more complex, and there must be more security issues that we do not know. Thus, our future work will be devoted to extend our work to multiparty cases.

REFERENCES

[1] C. C. Aggarwal, Y. Li, J. Wang, and J. Wang, "Frequent pattern mining with uncertain data," in Proc. ACM SIGKDD'09, 2009, pp. 29–38.

[2] R. Agrawal and R. Srikant, "Mining sequential patterns," in Proc. IEEE ICDE'95, 1995, pp. 3–14.

[3] J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," in Proc. ACM SIGIR'98, 1998, pp. 37–45.

[4] T. Bernecker, H.-P. Kriegel, M. Renz, F. Verhein, and A. Zuefle, "Probabilistic frequent itemset mining in uncertain databases," in Proc. ACM SIGKDD'09, 2009, pp. 119–128.

[5] D. Blei and J. Lafferty, "Correlated topic models," Adv. Neural Inf. Process. Syst., vol. 18, pp. 147–154, 2006.

[6] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in Proc. ACM ICML'06, 2006, pp. 113–120.

[7] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," J. Mach. Learn. Res., vol. 3, pp. 993–1022, 2003.

[8] J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. S. Ebert, and T. Ertl, "Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition," in Proc. IEEE VAST'12, 2012, pp. 143–152.

[9] K. Chen, L. Luesukprasert, and S. T. Chou, "Hot topic extraction based on timeline analysis and multidimensional sentence modeling," IEEE Trans. Knowl. Data Eng., vol. 19, no. 8, pp. 1016–1025, 2007.

[10] C. K. Chui and B. Kao, "A decremental approach for mining frequent itemsets from uncertain data," in Proc. PAKDD'08, 2008, pp. 64–75.

[11] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. X. Zhou, "LeadLine: Interactive visual analysis of text data through event identification and exploration," in Proc. IEEE VAST'12, 2012, pp. 93–102.

[12] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu, "Parameter free bursty events detection in text streams," in Proc. VLDB'05, 2005, pp. 181–192.

[13] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M. Hsu, "FreeSpan: frequent pattern-projected sequential pattern mining," in Proc. ACM SIGKDD'00, 2000, pp. 355–359.

[14] N. Hariri, B. Mobasher, and R. Burke, “Context-aware music recommendation based on latent topic sequential patterns,” in Proc. ACM RecSys’12, 2012, pp. 131–138.

[15] T. Hofmann, “Probabilistic latent semantic indexing,” in Proc. ACM SIGIR’99, 1999, pp. 50–57.

[16] L. Hong and B. D. Davison, “Empirical study of topic modeling in Twitter,” in Proc. ACM SOMA’10, 2010, pp. 80–88.

[17] Z. Hu, H. Wang, J. Zhu, M. Li, Y. Qiao, and C. Deng, “Discovery of rare sequential topic patterns in document stream,” in Proc. SIAM SDM’14, 2014, pp. 533–541.

[18] A. Krause, J. Leskovec, and C. Guestrin, “Data association for topic intensity tracking,” in Proc. ACM ICML’06, 2006, pp. 497–504.

[19] W. Li and A. McCallum, “Pachinko allocation: DAG-structured mixture models of topic correlations,” in Proc. ACM ICML’06, vol. 148, 2006, pp. 577–584.

[20] Y. Li, J. Bailey, L. Kulik, and J. Pei, “Mining probabilistic frequent spatio-temporal sequential patterns with gap constraints from uncertain databases,” in Proc. IEEE ICDM’13, 2013, pp. 448–457.

[21] N. R. Mabroukeh and C. I. Ezeife, “A taxonomy of sequential pattern mining algorithms,” ACM Comput. Surv., vol. 43, no. 1, pp. 3:1–3:41, 2010.

[22] A. K. McCallum. (2002) MALLET: A machine learning for language toolkit. [Online]. Available:

[23] Q. Mei, C. Liu, H. Su, and C. Zhai, “A probabilistic approach to spatiotemporal theme pattern mining on weblogs,” in Proc. WWW’06, 2006, pp. 533–542.

[24] D. Mimno, W. Li, and A. McCallum, “Mixtures of hierarchical topics with Pachinko allocation,” in Proc. ACM ICML’07, 2007, pp. 633–640.

[25] C. H. Mooney and J. F. Roddick, “Sequential pattern mining approaches and algorithms,” ACM Comput. Surv., vol. 45, no. 2, pp. 19:1–19:39, 2013.