Model Driven Secure Web Application development through trusted operating systems

Dr. Nitish Pathak
[1]*BVICAM, Guru Gobind Singh Indraprastha University (GGSIPU), New Delhi, India*

Abstract- This research paper propose the experimental setup for secure Engineering and use of security performance flexibility model to keep high security in web applications. This model allows system administrators to skip or disable some unnecessary security checks in trusted operating systems through which they can effectively balance their performance needs without compromising the security of the system. For example, system admin can tell that video on demand server is allowed to skip only security checks on reading files, while database server is allowed to skip only security checks on seeking files. Which operation is need to be skipped and which operation is not need to be skipped is very much subjective in nature, this will depend upon the user's requirement and the particular application's requirement. This work proposes object-oriented class-based software development, source code generation in C++ and the integration of security engineering into a model-driven software development.

Keywords- *Object Orientation, Forward engineering,* SELinux, *SPF, requirement elicitation, UML 2.0.*

I. INTRODUCTION

In the last decade, there has been vast growth in the field of networking, sharing of data worldwide. And then comes the most extensively used thing Internet have made cyber security a very crucial aspect of research and development. Its matter of concern for both the common users and researchers connected all over the world. Despite of lot of works undergoing we are still unable to get something that reliable and silver bullet that it may provide us with complete security for our systems. Being so advanced we still lack the basic potential to create such a system that is capable of stopping viruses and accessing our confidential data from our systems [1]. The security methods developed, researched till yet are implemented in the application layer of the computers which is making our systems more prone to data insecurity. These methods includes encryption using a key i.e. cryptography, using firewalls, access control using authentication, and application layer access control. The most two burning domains are Cryptography and authentication techniques in which max research is being done. Although these are something very difficult to crack but no one knows the dynamic minds making some of probability of data insecurity [2]. To some extent using firewalls and application layer access control have helped us but they do have a drawback. These two techniques can help in stopping the attacks using viruses uploaded on internet but fails to protect from internal security issue thus finally making our system vulnerable [3].

In reality, trusted operating systems are better choice for web applications to maintain the security concern, but this security will come at a cost. By using trusted systems, our web application will be more and more secure, but due to more security checks, the performance of the same system will disgrace in all respect[4][5]. The companies which came up with some promising operating systems with security features are Argus-Systems Group, HP, and Sun Microsystems [6]. Open Source OS are also providing well secured kernel having excellent security features and commonly known as secure systems, operating systems. National Security Agency has released the most secured and that too open source operating system called as SELinux i.e. security enhanced version of Linux operating system. Proper definitions of secure system vary from organization to organization [7]. SELinux trusted operating system can be used to implement the security in Web Application. These secure Systems are more complex for computer administrators to handle and manage. Such secure Systems require much more extra effort and time to setup the desired security policy on the part of administrator. The implementation of security policies, as per the requirement of user, is very complex in such systems. In this research paper, we are suggesting security enhanced version of Linux (SELinux) trusted operating systems for maintaining the high security concern in web applications [8]-[9].

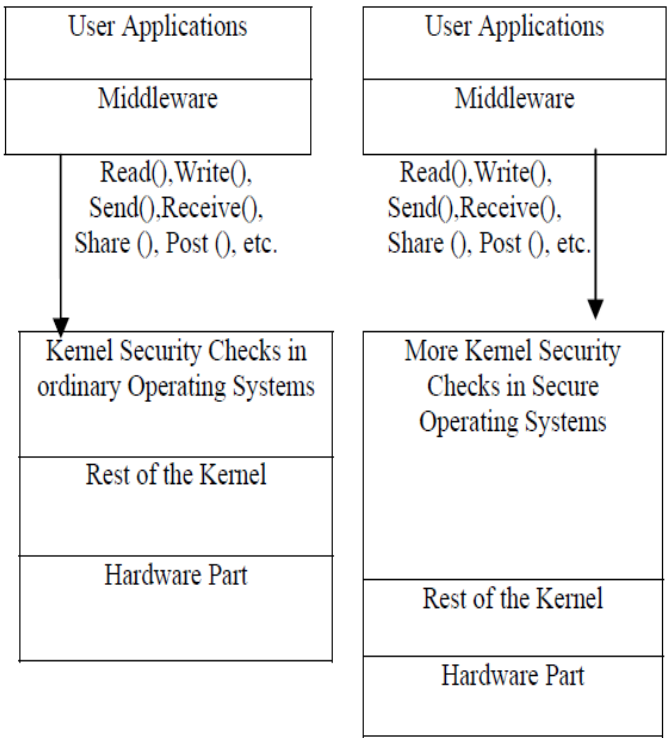
Model Driven Secure Web Application development, we are implementing through Unified Modeling Language. UML is an extensible language for software design such as web applications, database applications, and business modelling etc. UML is used in many customs for expressing the concepts such as software specification, website structure and business modeling [10]. In the proposed design and development, we are using UML2.0 based conceptual modeling for the development of secure application. It makes the programming simpler, more effective and manageable [11]. This research work proposes SPF based secure software analysis, SPF based secure software design and SPF based secure software development. Far above the ground, quality of software design is necessary for the success of software [12][13].

II. TRUSTED OPERATING SYSTEM BASED SECURE WEB APPLICATIONS

The essential structural design of this operating system is shown in Figure 1. The architecture of traditional operating systems is given in Fig. 1(a). System call interface helps the application and middleware interface to communicate with the

Operating System [14]. Fig. 1.(a), illustrating thin or slim security layer of operating systems kernel security checks. Now in order to provide higher security, lots of security checks are there in kernel of Trusted Operating Systems.

Fig. 1(b) demonstrates the additional security checks in the kernel. This will cause trusted operating systems to be slower than standard operating systems. Fig.1 (b) clearly depicts the thicker layer of kernel security checks. What all security measures are being taken in the kernel security check depends all on implementation and modeling. But the disadvantage of having extra security check is that whenever user tries to do any useful work it need to undergo all the checks thus deteriorating the system overall performance[15].



(a)Ordinary operating systems (b) Proposed structure of trusted operating systems

Fig. 1: Structure of trusted operating systems and ordinary operating systems.

III. PROBLEM EXPLANATION AND SOLUTION METHODOLOGY

Before moving into the problem, we are dealing, first we will be talking about the basic principles of Secure Operating Systems. As was mentioned previously, the term Trusted OS is interpreted differently and vary from one company to another software company. Security of TOS also affects the quality of multimedia and video streaming services. But in case of trusted operating systems for every read from disk system have to do repeated checks. Due to repeated security checks, the frame rate and quality of video streaming will be decreased. The result will be worst when the running system is heavily loaded with lot of programs. In this case when system is very loaded with multiple processing and multiple applications are in running mode, definitely the streaming will be very slow and the quality of video may be very poor[16][17].

The architectural consideration behind the SPF configuration is demonstrated in Fig. 2. This research paper proposes a concept of SPF, in order to gain improved performance and speed for particular system workloads for Trusted Operating Systems. This particular proposed model allows system administrators to skip or disable some unnecessary security checks in these operating systems through which they can effectively balance their performance needs without compromising the security of the system [18].

Fig. 2 providing administrator's option of disabling security checks for some useless system processes or undesired process. By skipping these processes or system calls the performance of system will be improved. These SPF based improved version of secure systems can be used for desired web application, for maintaining the desire level of security concern in particular application.

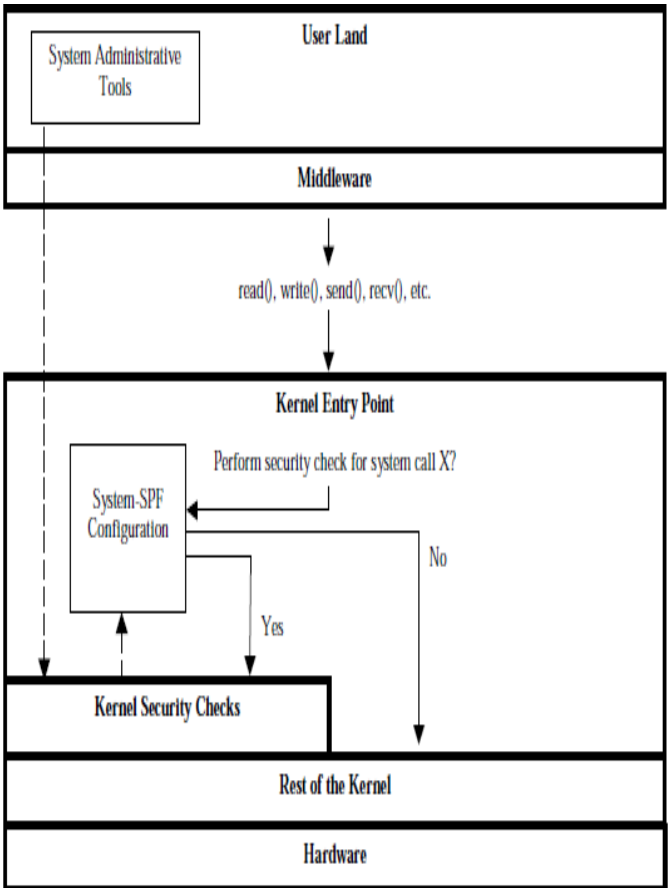


Fig. 2: System-SPF structural design for online store web application.

IV. OBJECT ORIENTED IMPLEMENTATION IN C++ FOR ONLINE STORE BASED WEB APPLICATION

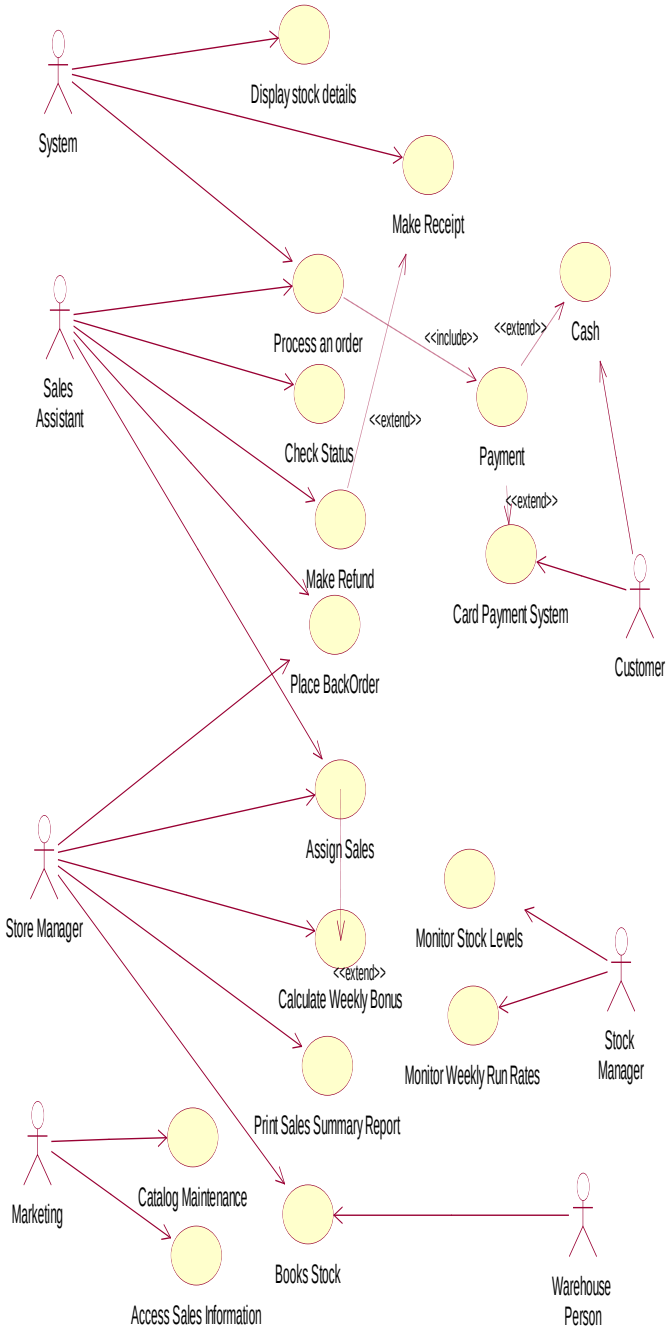


Fig. 3: Use case model for online store based web application.

Fig. 3 shows the functional requirements of system through use case diagram. These functional requirements are implemented through functions, member functions, methods, procedures, sub routines etc. We can see that store manager, sales assistant, customer, warehouse person and stock manager etc. are the actor. Rest of the diagram indicates the use cases (Display Stock Details, Payment Use Case, Order Delivery Use Case and Sales Summary Report Use Case etc.).

4.1. Description of major use cases for above Web Application

TABLE 1: DISPLAY STOCK DETAILS.

Brief Description	The system displays info to Sales assistant
Actors	System
Flow of Events	• Sales assistant enters the product numbers (Pro. No.) and required quantities into system. • System displays the description of the product.
Alternative Flow	If the stocks control system in not functioning, it will not start.
Precondition	Product numbers must be entered
Post condition	Information regarding product is displayed

TABLE 2: ORDER DELIVERY USE CASE.

Brief Description	This use case enables the customer to get the delivery of their order from the stock control system.
Actors	**Customer**
Flow of Events	• Places order • Enter delivery details
Alternative Flow	If the processing doesn't get successful, generate error report to the sales assistant.
Precondition	The order should be successfully processed on the system
Post condition	If the use case was successful the order shall be delivered to the customer. If not, the system state is unchanged.

TABLE 3: PAYMENT USE CASE DESCRIPTION.

Brief Description of Payment	Payment Use Case allows the customer to make payment for his order in the stock control system.
Actors	Customer
Flow of Events	• Places order • Enter delivery details • Makes payment
Alternative Flow	If the processing doesn't get successful, generate error report to the customer and order is unsuccessful.
Precondition	The order should be successfully processed on the system
Post condition	If the Payment Use Case was successful the payment for the order shall be made by the customer. If not, the order is incomplete.

TABLE 4: MAKE REFUNDS USE CASE

Brief Description	The sales assistant makes the refunds to the customer by initiating this use case.
Actors	Sales Assistant, Customer
Flow of Events	• The customer produces a valid receipt. • The refunds are made by the sales assistant. • The use case ends.
Alternative Flow	Invalid Receipt • The customer doesn't produce a valid receipt. • He is asked for a valid receipt. If he is able to produce a valid receipt, the basic flow step 'REFUND' is resumed. Otherwise the use case ends.
Precondition	Customer has made all the payments.
Post condition	Customer has got refunds.

With the help of above use case diagram, we develop the component based classes. In object oriented class diagram, designer will identify the classes. These classes can be identified through software requirement specification (SRS). As normal practices, actors of use case diagram are considered as classes and the use cases are considered as member functions or methods of the classes. The standard class diagram of online store is as follows in Fig. 4.

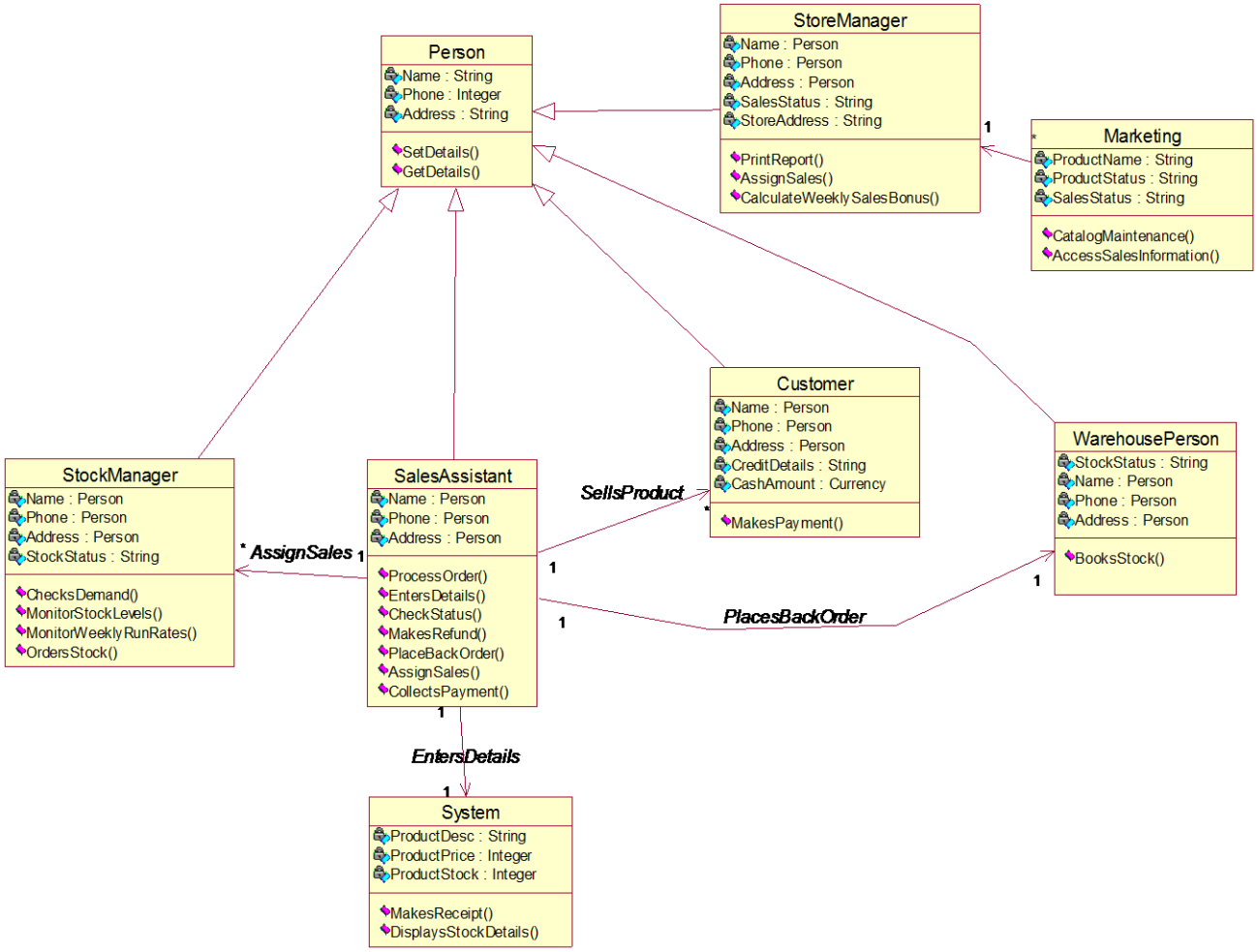


Fig. 4: Class diagram for online store web application.

When we want to model the structure of a system or a web application, we can make use of object oriented class diagram. Classes of applications are more or less like entities in entity relationship diagram. As we know, analysis is close to design phase of software development life cycle and design is close to development. We develop the component based class diagram. In this online store Based Web Application, storing objects may be sales clerk, inventory, Credit card, Cheque, store manager, Payment, Person, marketing, stock manager, person, warehouse person, invoice, system, customer etc. (See Fig. 4.).

The class wise equivalent C++ code for this case study is as follows-

Sample Code for understanding:

```
#include "Customer.h"
//##ModelId=4F7A84E80006
Customer::MakesPayment()
{
}
```

```
Marketing:
#include "Marketing.h"
//##ModelId=4F7A801C0072
Marketing::CatalogMaintenance()
{
}
//##ModelId=4F7A802F036A
Marketing::AccessSalesInformation()
{
}
```

```
Person:
#include "Person.h"
//##ModelId=4F7A86790240
Person::SetDetails()
{
}
//##ModelId=4F7A86840363
Person::GetDetails()
{
```

```
}
SalesAssistant:
#include "SalesAssistant.h"
//##ModelId=4F7A821600FF
SalesAssistant::ProcessOrder()
{
}
//##ModelId=4F7A821E0271
SalesAssistant::EntersDetails()
{
}
//##ModelId=4F7A82340127
SalesAssistant::CheckStatus()
{
}
//##ModelId=4F7A8238031B
SalesAssistant::MakesRefund()
{
}
//##ModelId=4F7A823D03A7
SalesAssistant::PlaceBackOrder()
{
}
//##ModelId=4F7A8267014F
SalesAssistant::AssignSales()
{
}
//##ModelId=4F7A851701B4
SalesAssistant::CollectsPayment()
{
}
StockManager:
#include "StockManager.h"

//##ModelId=4F7A81730087
StockManager::ChecksDemand()
{
}
//##ModelId=4F7A81780131
StockManager::MonitorStockLevels()
{
}
//##ModelId=4F7A81920127
StockManager::MonitorWeeklyRunRates()
{
}
//##ModelId=4F7A81B302E9
StockManager::OrdersStock()
{
}
StoreManager:
#include "StoreManager.h"
//##ModelId=4F7A8109020D
StoreManager::PrintReport()
{
}
#endif /*
```

With the help of above software development process, developers can identify software Metrics like no. of data members, no. of data members per super class, no. of data members per sub class, member functions, the length of the program, Volume, vocabulary of a program, average number of live variables, Count of executable statements, member functions per class, data structure metrics, and information flow etc. These software Metrics can be identified through halstead software science measures and data structure metrics.

V. RESULTS AND DISCUSSION

Tables 5 show the performance results that are appropriate to SPF. This table showing the results with SPF, without SPF and showing the performance compression for SELinux.

Table 5: Comparison of performance improvement after security checks skipped in SELinux trusted operating system.

File System Tests	SELinux without SPF Performance Degradation	SELinux with System-SPF model Improvement Over SELinux No SPF
Random Disk Reads (K) per second	-6%	+5%
Random Disk Writes (K) per second	-6%	+6%
Sequential Disk Reads (K) per second	-9%	+7%
Sequential Disk Writes (K) per second	-5%	+6%
Disk Copies (K) per second	-7%	+7%

Additional advantage of choosing SELinux is being open source thus allowing modification and change as per your requirement. Security performance Flexibility (SPF) model of SELinux Operating system can be implemented for security. Just because of privacy and confidentiality in Trusted Operating Systems, the source code of any software company, business and armed forces will not be available for normal user. So obtaining such source code in specific language is not as easy as we think. The privacy and security implementation for any system will vary from one Development Company to another.

VI. CONCLUSION

This research paper presents a SPF based approach for web applications and the integration of security engineering into a model-driven software development.This research work showcase the effectiveness of UML 2.0 based object oriented modeling with primary focus of security through the system level SPF in web applications.This model allowed system administrators to skip or disable some unnecessary security checks in trusted operating systems through which they can effectively balance their performance needs without compromising the security of the system.

VII. REFERENCES

[1] Pedro, V.; and Vicente, P. (2014). A survey of requirements specification in model-driven development of web applications. ACM Transactions on the Web, 5(2).

[2] Davis, J. P. (2009). Propositional logic constraint patterns and their use in UML-based conceptual modelling and analysis. IEEE Transactions on Knowledge and Data Engineering, 19(3).

[3] Barbara, P.; and Myra, S. (2012). Privacy-preserving query log mining for business confidentiality protection. ACM Transactions on the Web, 4(3).

[4] Nitish Pathak, Girish Sharma and B. M. Singh "Forward Engineering Based Implementation of TOS in Social Networking" published in International Journal of Computer Applications, Volume 102 - Number 11, Sep-2014, pp: 33-38, ISSN: 0975 - 8887.Foundation of Computer Science, New York, USA.

[5] Selby, R.W.; and Basili, V.R. (1987). Clean room software development: an empirical evaluation. IEEE Trans. Software Eng., 13(9), 1027-1037.

[6] Betty, H.C.C.; and Enoch, Y. W. (2002). Formalizing and integrating the dynamic model for object-oriented modelling. IEEE Transactions on Software Engineering, 28(8).

[7] Nitish Pathak and Neelam Sharma "SPF Based SELinux Operating System for Multimedia Applications." Published in International Journal of Reviews in Computing, ISSN: 2076-3328, pp.97-101, Vol.8, December-2011.

[8] Simona, B.; Jos,´E. M.; and Dorina, C. P. (2012). Dependability modelling and analysis of software systems specified with UML. ACM Computing Surveys, 45(1).

[9] Keng, S.; and Lihyunn, L. (2004). Are use case and class diagrams complementary in requirements analysis? An experimental study on use case and class diagrams in UML. Requirements Eng (2004), 229–237, Springer-Verlag London Limited.

[10] Pathak, N.; Sharma, G.; and Singh, B. M. (2015). Trusted operating system based model-driven development of secure web applications. Paper accepted for CSI - 2015; CSI - 50th Golden Jubilee Annual Convention, International Conference.

[11] Pathak, N.; Sharma, G.; and Singh, B. M. (2015). Towards designing of SPF based secure web application using UML 2.0. International Journal of Systems Assurance Engineering and Management, Springer.

[12] Georgia, M. K.; Dimitrios, A. K.; Christos, A. P.; Nikolaos, D. T.; and Iakovos, S. V. (2008). Model-driven development of composite web applications. iiWAS2008, November 24–26, Linz, Austria.

[13] Pathak, N.; Sharma, G.; and Singh, B. M. (2015). Experimental designing of SPF based secure web application using forward engineering. IEEE and IETE Sponsored 9th International Conference, BVICAM, New Delhi.

[14] Peter, D.; Timothy, W.; and Prashant, S. (2012). Modellus: automated modelling of complex internet data center applications. ACM Transactions on the Web, 6(2).

[15] Pathak, N.; Sharma, G.; and Singh, B. M. (2017). UML 2.0 Based Framework for the Development of Secure Web Application.BVICAM's International Journal of Information Technology (BIJIT), DOI: 10.1007/s141870-017-0001-3, February, 2017, Springer

[16] Kim, H.; Zhang, Y.; Oussena, S.; and Clark, T. (2009), A case study on model driven data integration for data centric software development. ACM, 2009.

[17] Pathak, N.; Sharma, G.; and Singh, B. M. (2015). Experimental analysis of SPF based secure web application. International Journal of Modern Education and Computer Science (IJMECS), 7(2), 48-55, Hong Kong.

[18] Pathak, N.; Sharma, G.; and Singh, B. M. (2017). An Empirical Perspective of Roundtrip Engineering for the development of Secure Web Application using UML 2.0. International Journal of Intelligent Systems and Applications, May, Vol.9, No.6 02, 43-54, Hong Kong.

Dr. Nitish Pathak is an Assistant Professor in Bharati Vidyapeeth's ICAM, GGSIPU, Delhi. He completed his Ph.D. in Computer Science and Engineering from Uttarakhand Technical University, Dehradun. He received his M.Tech. (CSE), M.Phil. (CS) and did his MCA with Honors from Dr. A.P.J. Abdul Kalam Technical University, Lucknow formerly UPTU. He has More than thirteen Years of extensive Corporate as well as Professional Teaching Experience at Graduate and Post Graduate level. He got Directorate Award (Two times on teacher's Day) for best teaching performance since 2007 to 2009, at ABES Engineering College, Ghaziabad. Got Best Research Paper Publication Award-2017, paper entitled "Towards Designing of SPF based secure web application using UML 2.0" published in International Journal of Systems Assurance Engineering and Management, Springer. Got INDIACom-2016 outstanding contribution award and CSI outstanding contribution award-2015 for conducting International Conference. These Conferences proceedings published by IEEE and Springer respectively. He has supervised a number of projects and dissertations both at the undergraduate and post graduate level. He has been pro-actively involved with professional associations and is Life member of CSI and ISTE. He has published more than 25 research papers in various reputed International Journals and Conference Proceedings (e.g. IEEE, Springer and ACM etc.). He is the reviewer of various International journals and jointly edited five conference proceedings. His major interests include: Trusted Operating Systems; Software Engineering, OOSE, Technology and Innovation Management etc.