IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 12, Issue 3 Ver. V (May-June. 2017), PP 74-78
www.iosrjournals.org

The Usage of Selinux Operating System for the Development of Trusted and Secure Web Application

Sudhansu Bisoyi[1], Chinmaya Ranjan Pattanaik[2]
1(Department of Computer Science & Engineering, Gandhi Engineering College, India)
2(Department of Computer Science & Engineering, Gandhi Institute For Technology, India)

Abstract: *This research paper proposes the Secure Web Application Development and SELinux based distributed Trusted Operating System for maintaining the security aspect in web applications. Unlike the conventional methods of adding more and more security layers at the kernel level, in this proposed design, we are suggesting two subsystems like object manager and security server. This model allows system administrators to skip or disable some unnecessary security checks in SELinux trusted operating systems through which they can effectively balance their performance needs without compromising the security of the system. This paper suggests UML 2.0 based class-based software development and the integration of security engineering into a model-driven software development.*

Keywords: *Security enhanced version of Linux (SELinux), SPF, UML 2.0, object oriented Model Driven Development (OOMDD), Distributed trusted operating System (DTOS)*

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][2].

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 [3].

It is believed that security measures in kernel are much more effective than the application layer. In fact, after lot of research such operating systems have been developed which have much more mechanisms inside the OS kernel providing us very good level of security thus securing our systems [4][5]. In reality, trusted operating systems are better choice for web applications to maintain the security concern.

The Security is not something expected not only by big organizations but also by common consumers so now concerns are being there on this and many vendors are trying hard to fix the issue. This paper suggests UML 2.0 based class-based software development and the integration of security engineering into a model-driven software development [6]. Integration of presented web services has become more and more accepted in the development of new web applications [7][8]. This type of web applications can be classified as the composition of web services around UI flow. In this paper, for the development of user-centric web applications, the author has presented the application of model-driven techniques [9]. This research paper proposes the Secure Web Application Development and SELinuxbased distributed Trusted Operating System for maintaining the security aspect in web applications. This SPF based improved version of secure systems can be used for desired web application. Due to excellent performance of SPF based SELinux for web development; we are suggesting the same for the implementation of web applications[10][11].

II. Secure Web Application Developement And Solution Methodology

First we will be talking about the basic principles of Secure Operating Systems. Trusted OS is interpreted differently and vary from one company to another software company. During system programming, company develops the system software according to the requirement of end users. But there are some important features in all Trusted Operating Systems. They are as follows Least Privilege, Mandatory Access Control (MAC), Discretionary Access Control (DAC) and auditing[12][13].

The essential structural design of distributed trusted operating systems (DTOS) and Flask is revealed in Figure 1. Unlike the conventional methods of adding more and more security layers at the kernel level, we are suggesting two supplementary or extra subsystems in this structural design [14]. The responsibility of object

manager in this model is to call the security server each time whenever a system user tries to access anobject.

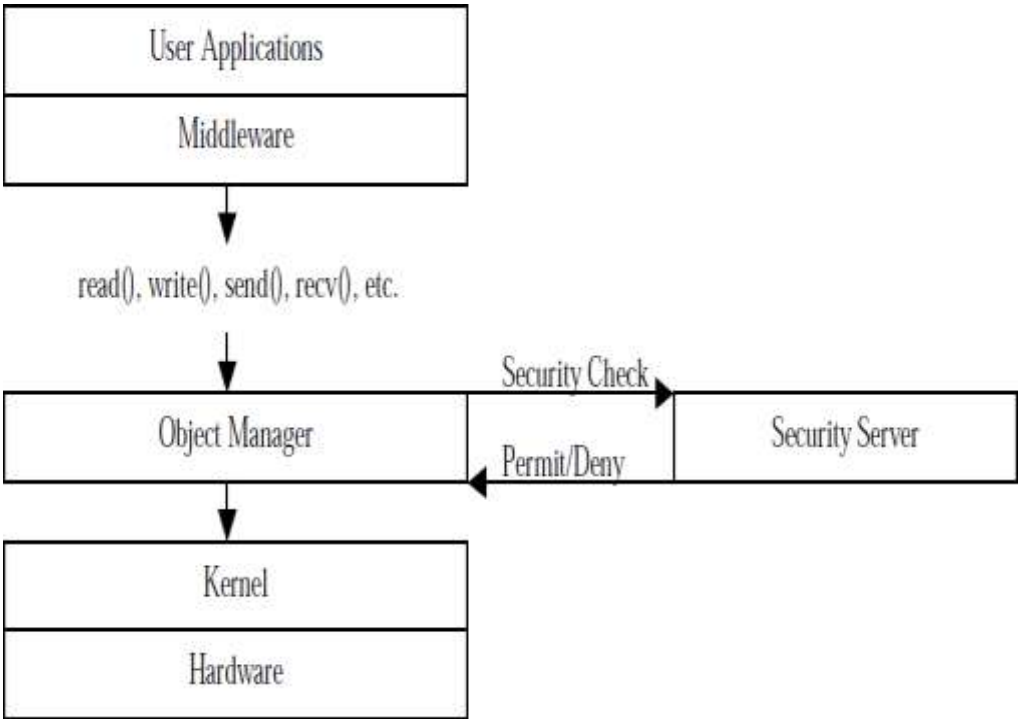

Figure: 1. Flexible security policy architecture for trusted operating systems.

Particular security server confirms the security pattern and informs the respective object manager if permission for requested operation is granted or denied. In this design approach we have to notice that the security server is not at all the component of the kernel. It is a different and separate part that can be called as per requirement, by the kernel. These different modules of the security server also can be altered or changed [15][16]. This is the main reason that's why DTOS as well as Flask both are built and implemented upon kernels. Almost security policies can be implemented if we consider security server as a separate part or separate module. These separate modules can be easily modified as per web applications requirement. This is not hard and fast that security need always will be unchanging or statically placed within the kernel. These security needs and implementation will vary from one real life application to another[17][18].

The implementations of web application for maintain the security is very much subjective in nature. The security requirement for the same will depend upon the user's needs. The object manager all the time calls the security server for checking the granted permissions. If security server grants permission for particular operations, then ok, operation or specific system call will be passed to kernel layer for execution. If permission is not granted, requested operations will notexecute.

The security features and security policies can be altered, as per need, dynamically as the system is in execution phase. Security layer will execute the security checks according to new altered security policies. In order to boost performance in DTOS as well as in Flask, security policy caching was recommended as a means to strengthen performance. The mainly referenced security checks are stored in a software-implemented cache, located in the module of object manager. This can surely increase the performance of systems. By caching recent security policies in the object manager, few parts of the security check can be ignored. In thiscase, if security policies changed by authentic user, new security policies will be implemented with immediateeffects.

III. Object Oriented Model Driven Development For Web Application

The Unified Modeling Language is an extensible language for software design such as web applications and business modeling etc. UML is used in many customs for expressing the concepts such as software specification, website structure and business modeling. In the proposed design and development, we are using UML2.0 based model driven development (MDD) and conceptual modeling for the development of secure application. It makes the programming simpler, more effective and manageable. Far above the ground, quality of software design is necessary for the success ofsoftware.

First of all, identify the pattern for the system calls, through requirement elicitation techniques. For any web applications, we can 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. 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. These object oriented languages are close to real world mapping. We develop the component based class diagram. In Store Stock Control Based Web Application, storing objects may be sales clerk, store manager, Person, marketing, warehouse person, etc. (See Fig. 2.). The standard class diagram of Store Stock Control is as follows in Fig.2.

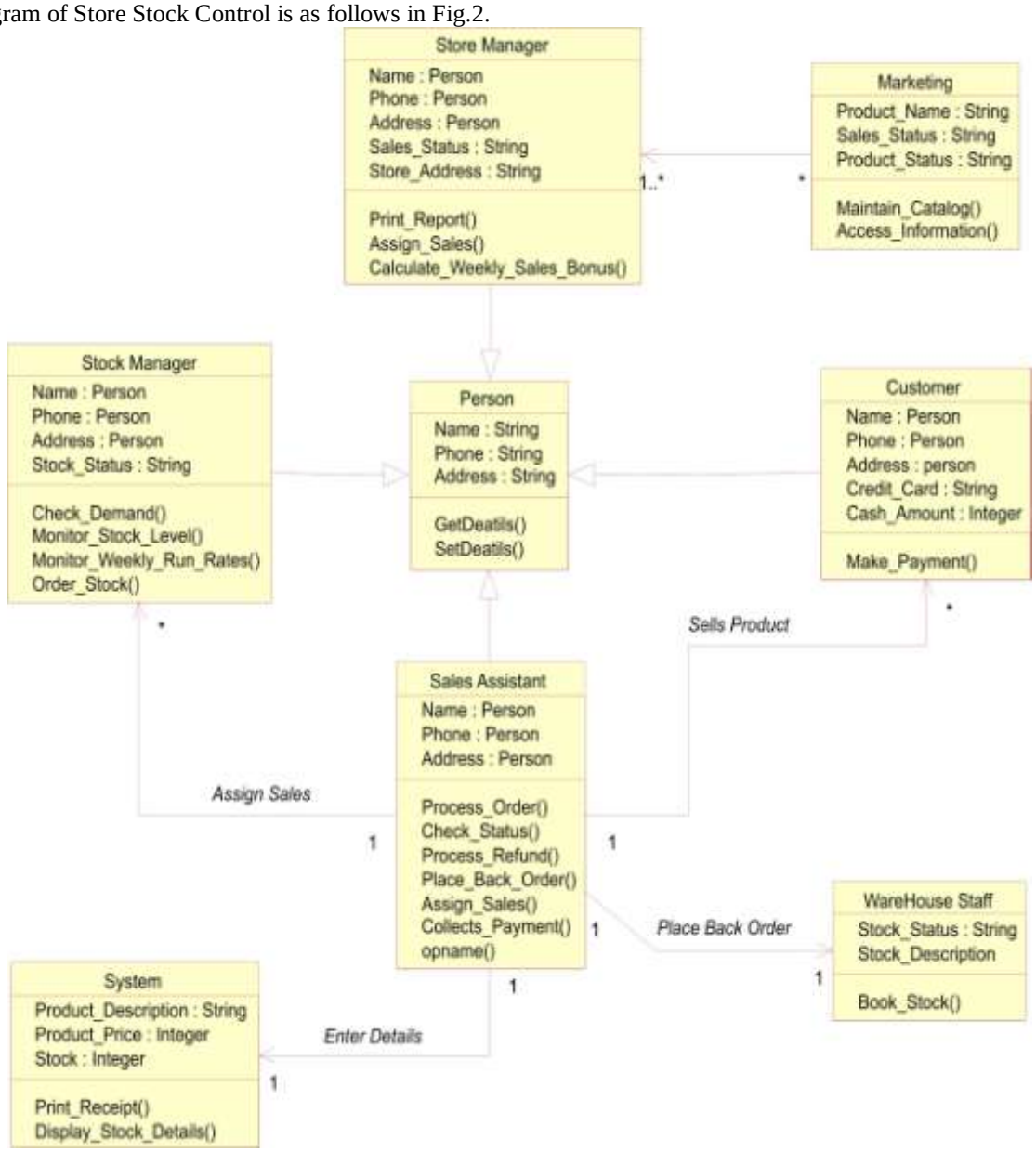

Figure: 2. Class diagram for online store stock control web application.

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, Count of executable statements, member functions per class, data structure metrics, and information flow etc. With the help of above mentioned approach, software project planning will become easier to developers.

IV. Results And Discussion

Tables 1 to 3 show the performance results that are appropriate to SPF. These tables showing the results with SPF, without SPF and showing the performance compression.

Due to good performance of SPF, we are suggesting the same for the implementation of web applications. The selection of SPF model for applications will depend upon the web application requirement. For example, Web

server deals with public information. Since majority of data and information is freely available to all users on web server, task of checking it during read from disk is something useless because this data is already readable by each and every user usinginternet.

Table 1: Security checks executed in SELinux trusted operating system.

File System Tests	SELinux without SPF	SELinux with System-SPF model
Random Disk Reads (K) per second	94167	93135
Random Disk Writes (K) per second	79188	79508
Sequential Disk Reads (K) per second	335527	325591
Sequential Disk Writes (K) per second	149616	153174
Disk Copies (K) per second	102252	102744

Table 2: Security checks skipped in SELinux trusted operating system.

File System Tests	SELinux without SPF	SELinux with System-SPF model
Random Disk Reads (K) per second	94167	99762
Random Disk Writes (K) per second	79188	84768
Sequential Disk Reads (K) per second	335527	363571
Sequential Disk Writes (K) per second	149616	159727
Disk Copies (K) per second	102252	110315

Table 3: 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%

So in this example, admin of the system can disable all the read checks in web server because they are actually useless which finally increases throughput of the web server. The real task of security comes when it comes to writing access. This means, in above application, web server integrity is the main issue rather than its confidentiality.

This paper suggests UML 2.0 based class-based software development and the integration of security engineering into a model-driven software development. Integration of presented web services has become more and more accepted in the development of new web applications. This type of web applications can be classified as the composition of web services around UI flow. In this paper, for the development of user-centric web applications, the author has presented the application of model-driven techniques. Since the rise in the pervasive access to internet, and in the usage of mobile code and networked appliances, there has been a sudden increase in the importance of the protection of software.

V. Conclusion

This paper suggests UML 2.0 based class-based software development and the integration of security engineering into a model-driven software development. This research paper proposes the Secure Web Application Development and SELinux based distributed Trusted Operating System for maintaining the security aspect in web applications. This SPF based improved version of secure systems can be used for desired web application. Due to excellent performance of SPF based SELinux for web development; we are suggesting the same for the implementation of web applications.

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]. 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.

[4]. Thiago, J.; Bittar, R.; Fortes, P. M.; and Luanna, L.L. (2009). Web communication and interaction modelling using model-driven development. SIGDOC'09, 5–7, 2009, Bloomington, Indiana, USA.

[5]. Fernandes, J.M.; and Machado, R.J. (2001). From use cases to objects: an industrial information systems case study analysis. 7th International Conference on Object-Oriented Information Systems (OOIS '01), 319–328. Springer, Berlin Heidelberg, NewYork.

[6]. 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.

[7]. 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).

[8]. 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, InternationalConference.

[9]. Runeson, P.; and Martin, H. (2008). Guidelines for conducting and reporting case study research in software engineering. Empirical Software Eng (2009), 131–164, open access at Springerlink.com.

[10]. 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.

[11]. 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.

[12]. 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.

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

[14]. 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

[15]. 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.

[16]. 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, HongKong.

[17]. 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, HongKong.

[18]. Marcin, H.; Marcin, M.; Ngoc, T. N.; and Andrzej, B. (2015). The automatic summarization of text documents in the cognitive integrated management information system. IEEE Conference on Computer Science and Information Systems (FedCSIS),1387-1396.