RA Journal of Applied Research
||Volume||3||Issue||02||Pages-844-850||Feb-2017|| ISSN (e): 2394-6709
www.rajournals.in
Doi :10.18535/rajar/v3i2.02

Improving Network Security using Aspect Oriented Programming

***Mazen Ghareb*[1]**
Computer Science Department, University of Human Development-Iraq
Email: *mazen.ismaeel@uhd.edu.iq*

Abstract :System security is critical in many modern software applications, yet there are many examples of system with little or no effective security in place. Modern computer System provides the most effective tool that the world uses for communication and exchange of information. Security of these applications is essinial ,but many are used to get to data without effective security. It is basic for system architects to ensure that the system is secuere as possible. Software designers need to insert or execute security in their applications. Security can be effectively implemented before the phases of the product improvement process, yet some designers for the most part ignore the security functionalities of their framework. System security issues include ways which program security influences overall system security. This paper concentrates on the ways in which Aspect-Oriented Programming (AOP) can be utilized to incorporate security into such software. The way security is actualized utilizing Aspect Oriented Programming is described, as is an approach to decrease the complexity of the program by permitting programming engineers to include security after the underlying program improvement stage is done.

Keywords: Aspect Oriented Security, Network Security, Application Security, online application security and network security awareness.

1. INTRODUCTION

Computer networks are fundamental for modern day correspondence. This is shown by the way that individuals need to consume information. For instance, a delivery driver may need to accesstheir emailfrom their car, or a manager may need to access their company intranet while on vacation. Web application systems, such as email servers, must be able to exchange emails and files safely, for example, e-business happening over the internet. People are normally delicate to security [1]. They dependably need to keep up their protection and ensure that they are sheltered from security issues like data leakage while purchasing goods on the web. These issues deliver a test to the system architects to ensure that everything is as secure as could be expected throughout the system. Security is a vital part in web application such as e-commerce and mail exchange system . When building a secure web application such e-commerce business site it mustcontribute to the general security of the web [14].There are many web application that have more secure than others in the internet the reasons because the architecture of the program is different so this will affect the security as well. We indicate how the originators of utilizations can manage the security issues. Whatever is left of the paper manages actualizing security amid or change the underlying programming improvement stage. Since we trust that.Since system security is additionally influenced by the application being utilized, it is important to begin at the base of the issue: ensuring that a product application is secured.A program used to get into the system can likewise contribute to the general security of the web. This system could be a web application or whatever other online application such as online banking. Whatever is left of the paper manages to execute security amid or adjust the underlying programming improvement stage. Since we trust that.Since system security is additionally influenced by the application being utilized, it is important to begin at the base of the issue: ensuring that a product application is secured.During programming improvement, developers endeavor to build up a product application that meets the user`s prerequisite and that is secure and solid. A few engineers ordinarily disregard the required security highlights in programming applications since they tend to concentrate more on client necessities. In spite of the fact that Aspect- oriented programming (AOP} makes it conceivable to fix insecurity, modify the underlying advancement stage [1]. It is pretty much as imperative to present security in the prior phases of the software advancement process. As to make a secure mindfulness culture. We consider security to be an essential component of any framework. Despite the fact that it may not generally be unequivocally expressed by a customer as one of the prerequisites for the framework.In this paper, we clarify how AOP can be utilized to build security of the applications and this will impact the security of the network, and give numerous guidance to the engineers and a few techniques for implementing security among developers in system level. The following area of this paper form away from plain sight of AOP. Segment III expounds on how AOP can be utilized to actualize security exclusively at the application level. Area IV displays a strategy that can be utilized to authorize security strategies that help designers at the application level

RA Journal of Applied Research
||Volume||3||Issue||02||Pages-844-850||Feb-2017|| ISSN (e): 2394-6709
www.rajournals.in

and in addition to the system level. The last segment finishes up the paper and gives some future work.

2. RELATED WORK

As it is stated in the introduction section, there is anclear relationship on how application security can impact system security. The foundation in this area concentrates on AOP and how it can be use at an application level development. Despite the fact that exploration demonstrates that the utilization of article arranged programming (OOP) has improved programming advancement throughout the years, by one means or another designer still find it hard to express an issue completely into a model that is totally secluded and exemplify [2]. AOP is a programming techniques that goes for determining a few difficulties that OOP couldn't address viably. These difficulties incorporate code scatter and code tangling. Code scattering happens when the code required to satisfy one concern is spread over the classes required to satisfy another concern. Code tangling needs to do with utilizing a solitary technique or class to actualize numerous concerns [2]. A concern is requiremed framework and is executed in a code structure [2]. The aspect-oriented paradigm makes the code to be compact and simple to reuse. AOP was developed at the Xerox Palo Alto Research Center by Gregor Kiczales and his partners [4]. AOP was additionally intended to increase modularity of code. Modularity helps with clean out code tangling and diffusing. An application or a project with tangled code and scattered code is hard to alter and troubleshoot,therefore system need to be breaking into little modules of then apply aspect techniques to solve of these issues [2]. AOP additionally addresses this issue by making the utilization of aspect and it keeps software engineers from committing basic errors like invoking a wrong method, whileAn aspect is a programming build that on same place on own record. It recognizes a state of interest and operations to be connected. Purpose of interest is a particular part of the project that a particular activity must be executed and operations are the moves that are made by the perspective code. A perspective is created with different aspects that location security worries for this situation and it is autonomous of any programming language[3]. Figure 1 demonstrates a case of a aspects the login interpreter called in the two different classes.

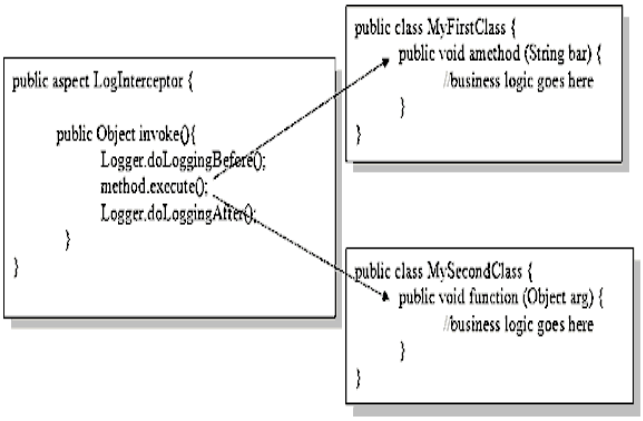

Fig.1 login security functionality into multiple class

it have basically made a proxy for technique calls. The imperative thing to note about this case is that both MyFirstClass and MySecondClass have no information and no reliance on LogInterceptor. In a genuine application that logging code is most likely imitated in many diverse class records, so incorporating it like this will diminish a considerable measure of source code. This has significant consequences since we can now add security Aspects to applications without modifying existing code. Another example is an input validator using AspectJ the following code in programming language AspectJ, which can be effectively coordinated into existing Java applications to moderate the XSS in the application. Figure 2 explainsthesecond example of an XSS attacker detector.

```
public aspect InputValidator {
    private Pattern pattern;
    private final String patternString = "[a-zA-Z0-9@:\\/$.]*";

    pointcut myPointcut(String argument): execution(* NewLeadBean.set*(String))
        && args(argument);

    before (String argument): myPointcut(argument){
        Matcher matcher;
        if (pattern == null)
            pattern = Pattern.compile(patternString);
        if (argument == null)
            return;
        matcher = pattern.matcher(argument);
        if (!matcher.matches())
            throw new IllegalArgumentException("application security attack thwarted!");
    }
}
```

Fig.2 Validating input using AOP

A pointcut in AOP defines where we want our aspects to interact with the rest of the code. What'sattractively better is that I now have a main bit of code that It can be changed effectively. For example, assume it have found that the specific whitelist is helpless against a newfound type of XSS, then it can be changed the rundown in one place and have it consequently proliferate. Besides, on the off chance that It has found different parts of my code are helpless against the same XSS, basically characterize extra pointcuts and I'll rapidly be shielded against theseattacks also. Likewise, on

RA Journal of Applied Research
||Volume||3||Issue||02||Pages-844-850||Feb-2017|| ISSN (e): 2394-6709
www.rajournals.in
Doi :10.18535/rajar/v3i2.02

another hand finding there are a few diverse white records that I require in my application (e.g. One that acknowledges just alpha, one that acknowledges alphanumeric, and so on.) then I can characterize various consistent expressions inside the same aspect. AOP makes programming simple for engineers since they don't need to dependably do up the same code for a particular issues that has as of now been modified. Clients typically bring increasingly extra attention toward the application among the advancement procedure and accordingly different concerns can now and then be not entirely obvious. One of the primary motivations behind AOP is that of determining the organized change on a system. Organized change includes embedding or leave code at very much defined point [5]. AOP depends on the components of its host language. Which is the reason the client does not need to learn such a variety of new methods. There are various advantages that outcomes from utilizing the AOP system. The AOP approach enhances execution since the techniques or operations are more compact and software engineers invest less energy not rewriting the same code. It is clear that AOP empowers better encapsulation of various strategies and advances future interoperation [6]. AOP has been around for around 15 years now. Aspect has a comprehensive and expressive pointcut specification language that allows specifying particular points in the control flowof the program where advices are to be applied. All of them are important from a security point of view.The usefulness of the AspectJ pointcuts according to the security target. Although AspectJ supports those efficient and useful pointcut designators for security hardening such as setting the security envirement during the initialization of the object and the execution time, calling for methods or constructor and execute the particular code regarding particular conditions.
They are not enough to express all the security harden pr actices. Indeed, the following possible extensions to Aspe ctJ are identified for security hardening:

Dataflow pointcut, Predicted control flow pointcut,loop poin tcut.Wildcard for pattern matching.The
modifiers in type pattern syntax,Pointcuts for getting and set ting local variables. Synchronized block join points.
The detailed discussion about the suggested extensions available in [7], [8], [9], [10], [11].

Another important discovery with aspect in web was a Spring techniques for secuirty which impart all applications on the network and make your network more secure. Spring security started in late 2003 as "The Acegi Security System for Spring", Acegi Security turned into an official Spring Portfolio venture towards the end of 2007 and was rebranded as "Spring Security" [12]. Spring Security (formerly known as Acegi) alongside AOP can improve the execution of the exemplary crosscutting concerns of more grounded implementation of security in an application. Enterprise applications need to address numerous crosscutting functionalities: exchange administration, security, reviewing, administration level understanding, checking, simultaneousness control, enhancing application accessibility, error handling, etcetera. Numerous scheme applications use AOP to actualize these functionalities. Every one of the case in given here depend on true issues and their AOP solutions. Virtually every project that utilizations Spring utilizes AOP [9]. Many applications begin with prewritten Aspect supplied bySpring (basically exchange management and security). But due to the AspectJ punctuation, composing custom aspect is turning into a typical assignment. Subsequent to achieving the cutoff points of Spring AOP, numerous applications move toward AspectJ weaving. The rank of the typical trigger point of this change is crosscutting of space articles or different types of more profound crosscutting functionalities. At that time, it's basic to begin with the AspectJ language structure (which is utilized by Spring's intermediary based AOP) alongside the load time weaver. In any case, applications that don't utilize spring regularly utilize AspectJ weaving from the earliest starting point. Spring security alongside AOP can streamline the implementation of the great crosscutting concerns of securing enterprise application. The security perspective needs to do two things, firstly select join points that need authentication or approval and after that prompt the chose join points to perform validation and approval as appeared in Fig.3.

```
aspect authentication
{
        before: call (public void update* (..))  // this is a pointcut
        {
                int tries = 0 ;  //this advice should be woven to the system
                string userPassword = Password.Get ( tries ) ;
                while (tries < 3 && userPassword != thisUser.password ( ) )
                {
                        // allow 3 tries to get the password right
                        tries = tries + 1 ;
                        userPassword = Password.Get ( tries ) ;
                }
                if (userPassword != thisUser.password ( )) then
                        //if password wrong, assume user has forgotten to logout
                        System.Logout (thisUser.uid) ;
        }
} // authentication
```

Fig.3Authintication in Spring AOP

Moreover Spring Security gives instant answers for a couple of basic situations that permit implementing certain security necessities by including only a couple lines of design. Authorization is a procedure that sets up whether a verified client has adequate benefits to existing assets. For instance, just clients with the administrator privilige may get to certain website pages or invoke certain business strategies in your network. Spring Security gives part based and protest level approval. To oblige potentially complex custom necessities,

RA Journal of Applied Research
||Volume||3||Issue||02||Pages-844-850||Feb-2017|| ISSN (e): 2394-6709
www.rajournals.in

it gives a few segments that can alter.

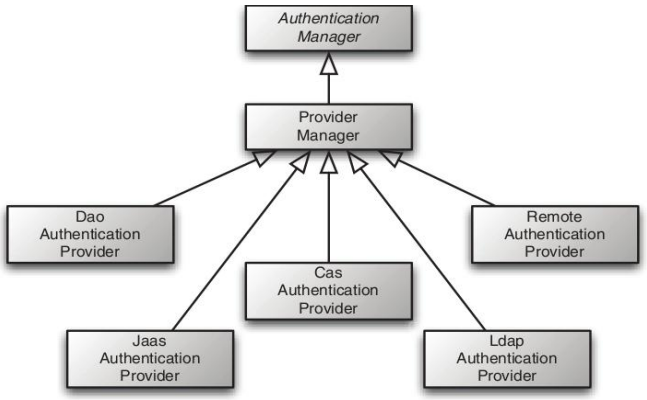

Fig. 4 Spring Security Level

There have been some security executions utilizing AOP as will be talked about inthe next section.

3. AOP SECURITY IMPLEMENTATION

3.1 AOP Application Level Security

Security must be implemented in any standard application, regardless of the language thathave been used. For instance, reuse, practicality and clarity, security related components {pieces of the code) ought to be speperate from other components source code in a project. The AOP system permits software engineers to separate security concerns from the code. This empowers software engineers to simply concentrate on building up the fundamental application and security specialists to determine the security properties that should be available in an application [1]. It has been demonstrated that designers are not exceptionally concerns about creating software applications that should be secured [13]. No change of the application is required for sources to explain security when it accompanies Aspect Oriented security. AspectOriented security is adaptable and extensible [8]. This is because of the capacity of AOP to weave or connect the concernsof making use of the aspetcs. By making the use of AOP there are systems of programming hardening location that have been actualized in applications running on untrusted hosts [5]. A perspective-situated project was utilized to do self-checking. Self-checking is a procedure where a system checks itself to confirm that it has not been modified [8]. As indicated by discoveries from the analysis that was directed by Bostrom [2]: database encryption can be included adjusting the consummation of the framework by utilizing AOP. Most of language have an issue of presenting software products with classified and secure exchange information [12]. The other issue is that the software engineer should consider security together with instance running of the software. The following example shows the application level security using AOP.

Executions of this case: a conventional article situated usage in Java, and an aspect oriented execution in AspectJ. The key distinction between the executions is that in the AOP rendition the security conduct is executed in an aspect, whereas in the non-AOP code it is scattered over the techniques for getting permissioninto the record. In the aspect secure, the main part declare a pointcut named cross (). This pointcut distinguishes certain join pointson the system's execution, particularly the execution of the get permission. As it is shown in Figure below:

```
Class GetPermision {

public void GetPermision {return GC;}

Public Void PerformAction (File Path){

SecuirtyManager.check Permission(new  FilePermission(this.path,FILE_DELETE_ACTION))
}

public Void Write (File path)
{
SecuirtyManager.check Permission(new  FilePermission(this.path,FILE_UpDATE_ACTION))
}}

Class PerformAction
{

public void Getpermision() {return file;}

public void performAction (File path)
{
public void prepare(){

AccessContrller.doPrivileged (new PrvilegedAction())
public object deletefile(){
for(String path:path) {new prepare(path).deletefile(); }}

public void performaction (File path)
{
for (String path:path)
{
new updatefile(path).performAction();
}
}
}
Aspect makesecure{
pointcut cross():excution (void GetPermision.performAction(file))
before () returning:cross(){
SecuirtyManager.check Permission(new  FilePermission(this.path,FILE_UpDATE_ACTION))
}
}
}
```

Fig.5 Apllication level AOP Security

Apart from real execution code in that technique all other are the cross cutting concerns (optional concerns) which cause the scrambling and tangling of the code. E.g. authorization checking, logging, checking for client is the cross cutting concerns. The aspects oriented programming leave these sorts of concerns by characterizing the cross cutting concerns as aspect,It assumes an essential part ofthe separate security code from application code in the execution. An aspect can be represent as a blend of four vital parts: the aspect itself, a join point, a point cut, and advice. These ideas are critical to making a usable model with a detachment of concerns from configuration models in aspect-situated programming. Despite the fact that definitions may change, anaspectis largely considered as a component of a framework, which is scattered in numerous joint points all through the framework. Aspectsare generally used to communicate to crosscutting concerns that are isolated from the center business logic of a framework. For instance, see a File access application where a client can execute the operations like erase, update or add. The operation relies on the consents of the client given by the

RA Journal of Applied Research
||Volume||3||Issue||02||Pages-844-850||Feb-2017|| ISSN (e): 2394-6709
www.rajournals.in
Doi :10.18535/rajar/v3i2.02

security framework, i.e. a few clients might be approved to erase or upgrade a specific record while different clients may not be approved. Core business logic for this application framework would be the strategies that include erase or upgrade of record picked by the client, though concerns isolated from business rationale would incorporate all security concerns, for example, validation and access control. In this way security concerns can be demonstrated with both validation and access control as independent parts of the framework, since they are not specifically included with center business logic. As it is shown in the figure5 below

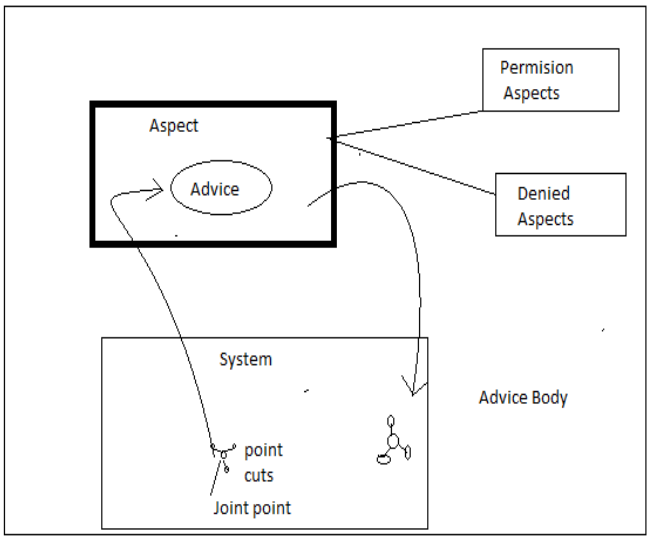

Fig.5 Security with AOP application level

3.2 Network Security Level

Computer networks serve to be a stage where applications cometogether, communicate and blend. These applications go about as a center layer between system clients and computers. Individuals use programming applications, i.e. web programs to get into the system. It is in this manner critical to view system security from the application level as well. The system is populated by applications that are created by various programmers and designers. Security at the system levelcan likewise be executed in two ways. The strategy of using a security checker and that of including another security module then will be connected at the system level on the grounds that the system involves applications that are running.Security tasks, for example, confirmation of client and approval of a client to view application assets are typically taken care of by the application server. These tasks can be designated to Spring security stream which is improve application server from taking care of these activities. Spring security fundamentally handles these assignments by executing standard Java. servlet. Filter. For instance if you want to define Spring security into your application, you have to pronounce the accompanying channel in your web.xml. As it appears in figure 6.

```
<filter>
  <filter-name>springSecurityFilterChain</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-cla
</filter>

<filter-mapping>
  <filter-name>springSecurityFilterChain</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

Fig. 6 Spring security Filter chain

Presently, this filter (springSecurityFilterChain) only delegates the solicitation to Spring security system where the security assignments characterized will be taken care of by security channels characterized in application connection. So how can this happen?

Inside the doFilter strategy for DelegatingFilterProxy (implementation of Java. servlet.Filter), the spring application connection will be checked for a bean named 'springSecurityFilterChain'. This "springSecurityFilterChain" bean is really aalias characterized for the spring channel chain. So when the check is done on application setting, it gives back the filterChainProxy bean. This channel chain is unique in relation to that of Java. servlet. FilterChain which is utilized by Java channels characterized as a part of web.xml to call up the following feasible channel if there exists one or go on the solicitation to a servlet/jsp. The bean filterChainProxy comprises of a requested rundown of secure channels that are characterized in the spring application setting. So here are the following arrangement of inquiries:

1. Who instates/characterizes this filterChainProxy?

2. What security channels are characterized in Spring application setting?

3. How are these security channels not quite the same as the ordinary channels characterized in web.xml?

Presently going to the principal question, filterChainProxy is instated when the ‹http› component from security namespace is characterized in the application connection. Here is the essential structure of ‹http› component: as it is shown in figure 7 below:

RA Journal of Applied Research
||Volume||3||Issue||02||Pages-844-850||Feb-2017|| ISSN (e): 2394-6709
www.rajournals.in

```
<sec:http auto-config="true">
 <sec:intercept-url pattern="/**" access="ROLE_USER" />
</sec:http>

<sec:authentication-manager id="authenticationManager">
  <sec:authentication-provider>
    <sec:user-service>
      <sec:user name="admin" password="password" authorities="ROLE_USER, ROLE_ADMIN" />
      <sec:user name="user" password="password" authorities="ROLE_USER" />
    </sec:user-service>
  </sec:authentication-provider>
</sec:authentication-manager>
```

Fig. 7 HTTP configuration with Spring security

Presently the HttpSecurityBeanDefinitionParser from Spring system understands this ‹http› component to enlist filterChainProxy in application connection. The http component with auto-config set to genuine is really a short hand documentation for the accompanying:

```
<sec:http>
  <sec:form-login />
  <sec:http-basic />
  <sec:logout />
</sec:http>
```

Fig.8 HttpSecurityParser

So as a matter of course when we include ‹http› component, the above three channels will be included. What's more, as we have set auto-config to genuine, BasicAuthenticationFilter, LogoutFilter and Username,PasswordAuthenticationFilter additionally gets added to the channel chain. Presently in the event that you take a part at the source code of any of these channels, these are likewise standard javax.servlet.Filter usage. It will be the characterizing of these channels in application connection as opposed to in web.xml, the application server exchanges the control ofSpring to manage security related assignments. What's more, the Spring's filterChainProxy will deal with fastening security channels that are to be connected on the solicitation. This answers the third question [14] [15].

4. EXPERIMENTAL ASPECTJ AND SPRING AOP INTEGRATION

Spring's proxy based AOP structure is appropriate for taking care of numerous bland middleware and application in particular issues. In any case, there are times when an all the more effective AOP arrangement is required: for instance, if the software engineer needs to add extra fields to a class, or prompt fine-grained objects that are not made by the Spring container, in that circumstance aspect is best option. Likewise, springsupply an effective mix with AspectJ. The most critical part of the Spring/AspectJ combination permits the spring to design AspectJ perspectives utilizing Dependency Injection. This conveys comparative advantages to aspects as to items. There is no requirement for aspects to use inspecial parts of the system systems; they can be designed in the same, steady, approach utilized for the whole application. Aspects can rely on upon application objects. For instance, a security aspect can rely on upon a security chief. It's conceivable to get a reference to an aspect through the significant spring setting. This can take into consideration dynamic reconfiguration of the aspect. AspectJ can uncover JavaBean properties for Setter Injection, and even actualize spring lifecycle interfaces, for example, BeanFactoryAware. In most cases, AspectJ aspectis singletons, with one case for every class loader. This single example is in charge of exhorting numerous article instances. A Spring section can't instantiate a perspective, as aspect doesn't have callable constructors. However, it can acquire a reference to an aspect utilizing the static aspectOf () technique that AspectJ characterizes for all aspects, and it can infuse conditions into that aspect. Consider a security perspective, which relies on upon a security supervisor. This aspect applies to all adjustments in the estimation of the equalization case variable in the Account class. It couldn't do if there should arise an occurrence of utilizing Spring AOP. The AspectJ code for the aspect [17].

5. CONCLUSION

The impact that security of programming applications have on the network security makes it to be basic for the product engineers and system designers to ensure that the applications are as steady and secure as could be expected under the circumstances. Poor security in a framework can make vulnerabilities else in the framework. Interlopers may effortlessly hack into the framework by abusing vulnerabilities. This recommends software engineers must ensure that security is actualized in the software applications that they create. Software engineers ought to consequently include security highlights into a project amid the improvement procedure. AOP minimizes the exertion and it spares the client or the engineer a lot of time. Having observe at the security of the software programs, it is clear that AOP makes a gigantic impact in the measured quality procedure of programming. AOP produces the idea that an engineer can simply connect to the security aspect even thenthe those programming advancement cycles has finished. The reason for the paper depends on a method for making software engineers more cognizant about combinesecured in their projects. As it was proposed in Section IV, software engineers should be reminded by method for having an exceptional class that will check the security usage in the project that is being produced. On other hand security has been implemented. An aggregate mistake ought to

RA Journal of Applied Research
||Volume||3||Issue||02||Pages-844-850||Feb-2017|| ISSN (e): 2394-6709
www.rajournals.in
Doi :10.18535/rajar/v3i2.02

happen,Without AOP the functionalities of security would be scattered over the application, with the same code copied in various modules. Truth be told, AOP connected to security comprehends a large portion of the normal useful issues concerning security. We have perceived how it is conceivable to actualize these functionalities without having tangled or scattered code, executing functionalities with aspects and advices. With AOP AspectJ and Spring AOP, arranging it legitimately for more grounded requirement security of the application subsequently we can have cleaner code, considerably more compact and simpler to keep up, troubleshoot and receive.

REFERENCES

[1] Viega, J., Bloch, J. & Chandra, P., 2001.Applying aspect-oriented programming to security. Cutter IT Journal, 14 (2), pp. 31–39.

[2] Gradecki, J.D. &Lesiecki, N., 2003. Mastering AspectJ: aspect-oriented programming in Java, John Wiley \& Sons.

[3] Shah, V. & Hill, F., 2003.An aspect-oriented security framework.In DARPA Information Survivability Conference and Exposition, 2003.Proceedings. pp. 143–145.

[4] Odersky, M., 2004. ECOOP 2004-Object-Oriented Programming: 18th European Conference, Oslo, Norway, June 14-18, 2004, Proceedings, Springer Science \& Business Media.

[5] Sirbi, K. &Kulkarni, P.J., 2010.Stronger enforcement of security using aop and spring aop.arXiv preprint arXiv:1006.4550.

[6] Jones, M. &Hamlen, K.W., 2010.Disambiguating aspect-oriented security policies. In Proceedings of the 9th International Conference on Aspect-Oriented Software Development. pp. 193–204.

[7] Laddad, R., 2003. AspectJ in action: practical aspect-oriented programming, Dreamtech Press.

[8] Kiczales, G. et al., 2001.An overview of AspectJ.In European Conference on Object-Oriented Programming. pp. 327–354.

[9] Alhadidi, D., Belblidia, N. &Debbabi, M., 2006.AspectJ assessment from a security perspective.Concordia Institute for Information Systems Engineering Concordia University, Montreal, Quebec, Canada.

[10] Spinczyk, O., Gal, A. &Schröder-Preikschat, W., 2002.AspectC++: an aspect-oriented extension to the C++ programming language. In Proceedings of the Fortieth International Conference on Tools Pacific: Objects for internet, mobile and embedded applications. pp. 53–60.

[11] Colyer, A. et al., 2004. Eclipse aspectj: aspect-oriented programming with aspectj and the eclipse aspectj development tools, Addison-Wesley Professional.

[12] Xia, L. & HUANG, H., 2008.Research and implementation of permission management based on Acegi security framework [J].Railway Computer Application, 6, p.002.

[13] Courbis, C. & Finkelstein, A., 2005.Towards aspect weaving applications.In Proceedings.27th International Conference on Software Engineering, 2005. ICSE 2005. pp. 69–77.

[14] Mularien, P., 2010. Spring Security 3, Packt Publishing Ltd.

[15] Viega, J. &Voas, J., 2000.Can aspect-oriented programming lead to more reliable software?IEEE software, 17(6), p.19.

[16] Mularien, P., 2010. Spring Security 3, Packt Publishing Ltd.

[17] S\lowikowski, P. & Zielinski, K., 2003. Comparison study of aspect-oriented and container managed security. In Proceedings of the Workshop on Analysis of Aspect Oriented Software, Germany.