ATLIST Tree for analysis of Vulnerability

Kavita S. Kumavat[1]
[1]PG Student at MET's IOE BKC
University of Pune
Nasik, Maharashtra, India
kavitakumavat26@gmail.com

Ranjana P. Dahake[2]
[2]Assistant Professor at MET's IOE BKC
University of Pune
Nasik, Maharashtra, India
dahakeranjana@gmail.com

Dr. M. U. Kharat[3]
[3]Professor at MET's IOE BKC
University of Pune
Nasik, Maharashtra, India
mukharat@rediffmail.com

***Abstract*—** Multiple internet application and services have become a necessary part of daily life, which enable the communication between different organizations. SOA based architecture provide a middleware in which multiple services can come and serve their application. However, such flexible communication practically always contains vulnerabilities. To remove different kind of vulnerabilities present in SOA based business application vulnerability detection algorithm is most effective by finding explicit link of particular user and also by building sitemap for each user and then comparing it. For analysing vulnerability ATLIST method is most appropriate for managing vulnerability detected in application.

Keywords- *ATLIST method, Business application, Security, SOA-based business process, Vulnerability analysis.*

*****

I. INTRODUCTION

In recent years internet services & web applications become very popular among the users. Because of the popularity of all these services & applications comes with the some problems. So because of these day to day needs such as banking, shopping & networking are done only by using web services. The services which user going to use it its work on two ends front end & back end, on front end because of the user interface user can make use of it & at the back end all the data which user is going to use are going to stored in the data server. The target of the attack is focuses on the data server because all the personal & corporate data which is very important information. For attacking attacker needs the data access so they move their concentration from front end to back end, because all the information is stored at the back end on server. Vulnerability is a weakness which tolerates an attacker to reduce a system security. Vulnerabilities are nothing but the faults occur in developing or scheming of system that causes damages to the privacy, accessibility, reliability of information in the system [1]. Vulnerability analysis supports escaping, realizing, fixing, and observing vulnerabilities present in system or application [4]. Vulnerability analysis provides idea about vulnerability management tasks which involve patterns of vulnerability basically two patterns are there i.e. dynamic and static analysis of source code [8]. A vulnerability pattern contains formal representation of vulnerability's attributes, with which a software tool can recognize the vulnerability. Orchestration is the SOA layer that creates business solution from the vast array of services and information flaws.

Most Internet based application can communicate with each other by transferring data from one application to another by using network based system. A Service Oriented Architecture provides flexibility for communicating multiple services with each other using network by loose coupling. An SOA based Standalone desktop applications vulnerability are detected by using anomalous scanning then create log file containing running application and comparing two systems log file detect malicious application [10]. By blocking that malicious application at run time vulnerability is removed from system. Also for web based application multiple vulnerabilities are detected like cross site scripting attack, Hijack future session attack, Injection attack, Direct DB attack, Vulnerability due to improper input process, Distributed DOS attack and privilege escalation attack [13]. Vulnerability Detection algorithm is useful for finding vulnerability present in application. In this algorithm two important functions are use first is BUILDSITEMAP use for building the sitemap according to the role of particular user. Second is EXTRACTLINK using this function explicit link available according particular role of the user is detected. ATLIST (Attentive Listener) method is useful for analyzing vulnerability present in application. ATLIST method can also monitor vulnerability present in given applications. ATLIST method can categorize detected vulnerabilities by their attack effect, active components, involved standard and triggering property [11].

II. LITERATURE SURVEY

In real world each and every software system or communication having some kind of deficiency due to which the security of that system is get abuses to application. Mostly, vulnerabilities are seen as weakness, flaw or error present in the software system by which attacker can exploit security of the system by behaving as a normal user. Due multiple software systems increases everyday also increases vulnerabilities. For malicious activity over Internet web based attack are primary vector. Number of people increasingly uses Internet for different activity it can also increase malicious activity day by day. Most general way of introducing vulnerability in application is nothing but the attacker is act as normal user or it accesses normal or admin user rights then that time application must be vulnerable. Due to which the confidentiality of the system is break and security of the system is exploit. The ratio of attacker to attack on financial system is more than other. Normally, web based vulnerability exploited due to coding mistake, configuration problem and inaccurate vendor implementation. Most of the industries refer Service Oriented Architecture because it provides a standardized way to serving multiple services collectively without providing detail knowledge about implementation of those services. This feature of SOA reduces project cost, implementation time and promotes software reuse. SOA uses SOAP, WSDL and BPEL.

ATLIST method is very effective method for analyzing vulnerability present SOA service orchestrations. ATLIST make SOA based services more flexible, reusable, extensively use standard and more transferable than other. ATLIST method is useful for deriving appropriate type of vulnerability and also help for deriving vulnerability pattern. Initially, Fault/Attack tree and FMEA (Failure Mode and Effect Analysis) methods are use for manually analyzing the vulnerability [2]. These methods leave much room for security expert to use personal experience and subjective skills for discovering old as well as new type of vulnerability [14]. FMEA and attack tree use the bottom up approach by which it can provide only a starting point but complete analysis is not cleared by this.

III. Vulnerability Types

Vulnerabilities are mostly occurs in web based applications which are as follows-

1. SQL injection

In SQL injection attack attacker can insert or inject malicious SQL query to system by which exploit can read sensitive data or change data present in system. As shown in following query attacker can enter into system and access all rights of normal user [3]:

{username'--}

Where, ' means or
-- means always true

By SQL injection by writing above username without password user can enter into system.

2. Direct DB Attack

Direct database attack can attack on data stored in system. For example if the system does not provide delete option for normal user but also by using following query writing in update option that particular record will be null.

'-- delete * from tablename

Where, ' means or
-- means always true

3. Privilege Escalation Attack

Computer exploit is nothing but privilege escalation which allows a user to access privileges extended to another user, potentially creating a vulnerability where a hacker could reconfigure a system and perform illegal operations.

4. Session Hijacking Attack

Session hijacking is also called as cookies hijacking in which exploitation of valid computer session is takes place.

5. Distributed Denial of Service

In client server application, multiple clients can run simultaneously if the server stop its working then also another systems can run as it is.

6. Password Brute force attack

Password brute force attack is nothing but repeatedly try guesses of username and password. This attack may be cause by normal user when he forgot his username or password or also by hacker which try commonly occurring password for specific username. To prevent this attack, the trying chances are fix for some number normal user know that and he will stop trying beyond that but attacker can try and then access is denied for that system.

7. Cross site scripting

Cross site scripting attack is occur when hacker hack on target website from a user's browser, often causing side effect such as stealing of user session or data compromise [6].This attack cause on browser side mostly using JavaScript. In cross site scripting attack script is generated when attack is performed. For example system can decide that username and password must be less than 8 characters.

```
<script>
 function check(){
     var a=document.getElementById("u_n").value;
     var b=document.getElementById("u_p").value;
         if(a.length>8 || b.length>8){
            alert("Script Attack Perform");
            return true;
         }
         return true;
}
 </script>
```

So, when attacker attacks on system then he should not know that limitation of username and password. Whenever attacker enter username and password more than 8 character then the script as alert will display and access denied.

IV. System Overview

Proposed system shown in figure 1. appropriately work for analysis of different types of vulnerability present in the system. SOA based system is refer for vulnerability analysis. System supports mainly two types of application. First is desktop based application in which system blindly scan and also comparison of two log files are generated. In web based application vulnerabilities are tested and prevented. Lastly by using detected vulnerability analysis is takes place and ATLIST tree is generated.

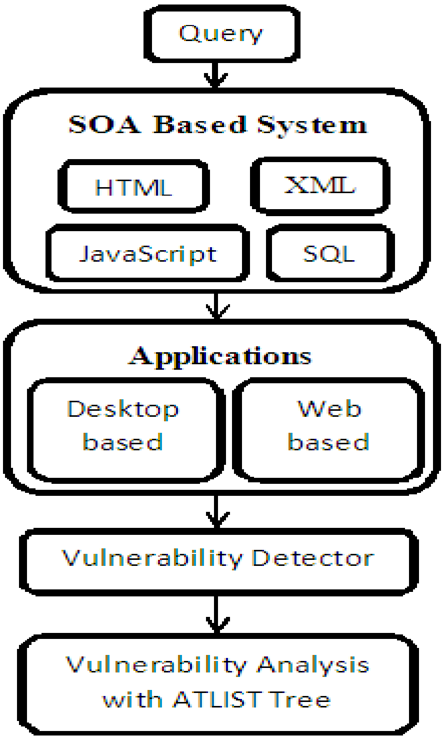


Figure 1: System Overview Diagram

In SOA based system different standards are use like HTML, XML, JavaScript and SQL which is explain briefly below.

A. SOA based Architecture

Normally SOA is an architecture style for building multiple web based application. It help for designing system and also for developing multiple services within that system. Service Oriented architecture is a software design or architecture design pattern which provides functionality of multiple services of different application [1]. In SOA based system services interact with each other also exchange data between each other [12]. Main layer in service oriented architecture is orchestration layer which contain the business logic. Hence for communication multiple services this layer provides one type of interface which is logic foe business. For building this architecture multiple standards are used. XML is use for configuration. By using TCP connect server and client and HTTP request and HTTP response is use for communication between client and server. All database of system is created using SQL. For developing SOA based architecture WSDL, SOAP, BPEL and UDDI are fundamental pieces use which is as follows-

1. WSDL(Web Services Description Language)

WSDL is a language useful for description of multiple types of services are mostly web based which and also communication between different web services. It mostly uses XML for writing. Namespace is associated to document location by using WSDL.

2. SOAP (Simple Object Access Protocol)

For accessing different web services in SOA based architecture use SOAP. It is platform independent, language independent, simple and extensible protocol. It also uses XML for exchanging structured information. Main features of it is its neutrality which means it can be use over any transport protocol like TCP, HTTP, SMTP etc. Also it is independent of any programming mode and extensibility is also providing by SOAP.

3. BPEL (Business Process Execution Languages)

Standardization of business processes flow and description of the process logic for the involved Web services will be invoked by using BPEL.

4. UDDI (Universal Description, Discovery and Integration)

UDDI is platform independent standard for XML based service registry. It provides for discovery of services and also retrieves that services description which is in WSDL.

V. ALGORITHMIC STRATEGY

For analyzing vulnerabilities present in SOA based system mainly following algorithms are use. First algorithm specifies steps for detecting vulnerability present in the system. CHECKACCESS function is use for finding reachable nodes for the user. And finally for analyzing detected vulnerability ATLIST tree algorithm is use which is as bellow-

A. Vulnerability Detection Algorithm

For detecting Vulnerability present in SOA based system following steps are useful. In this algorithm consider user a having high privilege than user b. Input to algorithm is specification of user a, user b and SQL query. Then by using CHECKACCESS function check the access of each user and find reachable node for each user. Compare privileges of both users by checking reachable nodes. If both users have same access for reachable nodes mean both users having same role hence vulnerability is present in the system.

SELECT (S, W)
DETECT VULS (Spec$_a$, Spec$_b$, SQ)

```
Vuls $\leftarrow$ 0
$N_a \leftarrow$ CHECKACCESS ($Spec_a$, SQ)
$N_b \leftarrow$ CHECKACCESS ($Spec_b$, SQ)
Privileged $\leftarrow N_a \setminus N_b$
For each n in Privileged
If ACCESS ($N_a$) = ACCESS ($N_b$)
And $R_a = R_b$
Then Vuls $\leftarrow$ Vuls $\cup$ {n}
return Vuls
ATLIST (Vuls)
return Vulnerability Analysis
```

B. CHECKACCESS Function

CHECKACCESS function is useful for checking access of user with specification of that user and for finding reachable nodes for that. Most reachable nodes are stored into work list with its adjacent edge. If any node is in working list then add that edge into edge set which is initially empty. By checking all file inclusion set F_i , page redirection set R_i and match links available for that user find the final most reachable nodes. The working of CHECKACCESS function work is specify by using below steps-

CHECKACCESS (Spec$_a$, N$_a$)

```
$E_a \leftarrow 0$
Visited $\leftarrow$ 0
WkLst $\leftarrow$ GETENTRIES ($Spec_a$)
while WkLst
do $< n_i , q_i > \leftarrow$ GETWORKNODE (WkLst, $Spec_a$)
$L_i \leftarrow$ EXTRACTLINK ($N_a$, $Spec_a$)
$N_j \leftarrow L_i \cup R_i \cup F_i$
for each $n_i$ in $N_i$
do $E_a \leftarrow E_a \cup \{(n_i , n_j)\}$
Visited $\leftarrow$ Visited $\cup \{n_i\}$
N $\leftarrow$ ACTIVE ($N_j$) \ (Visited $\cup$ WkLst)
WkLst $\leftarrow$ APPEND (WkLst, N)
return GETNODES ($E_a$)
```

For clarifying terms present in the above algorithm following table 1 is used which contain specification for each term present in the table and also elaborate it.

Table I
Algorithm Specification

Variables	Specification
S	Software based application
W	Web based application
R_a , R_b	Role for user a and user b Assume that role of user a having more privilege than role of user b.
$Spec_a$, $Spec_b$	It is the specification for user a and user b
Vuls	Vulnerability present in the application Initially equal to 0.
SQ	SQL Query
N_a ,N_b	Sets of reachable node for user a and user b
N	No. of privileged node i.e. active node
CHECKACCESS	Function for checking access role for particular user.
ATLIST	Function for analysing vulnerability present in the system and generating Vulnerability analysis tree.
Ea	Explicitly edge set
WkLst	List of working nodes
n_i	Working node form WkLst
q_i	Associate state from $Spec_a$
R_i	Page redirection set
F_i	File inclusion set
L_i	Match link that are present in N_a
(n_i ,n_j)	Outgoing edge

C. ATLIST Method

ATLIST Method is useful for analyzing detected vulnerability. In analysis of vulnerability, when application is get stop then first find out active component at that moment after that detect involved standard like TCP, HTTP, SQL and JavaScript and then find triggering properties.

For vulnerability analysis using ATLIST tree following steps are used-

1. Scan (req);
2. My request Processor (req)
3. Generate (token);
4. DController (token);
5. Categories (token);
6. Generate Hack logs;
7. Read records;
8. Generate (Tree);
9. Print Tree;

Figure 2. Shows ATLIST Tree structure which is detect, fix and monitor the vulnerability present in the web based system. According to algorithm request given from any client is scan first then by using Request Processor that request is tokenize. After that tokens get categories by using Data Controller and hack logs are generated. Using record present in hack log table ATLIST tree is generated. Whenever the vulnerability is detected in the application it get stop. At that moment stopped application use which active component is detected it is either web engine or database engine. After that search is minimizes for finding the involve standards normally SOA based system uses four types of standard like TCP, HTTP, JavaScript or SQL. Lastly search triggering properties which is change by specific vulnerability.

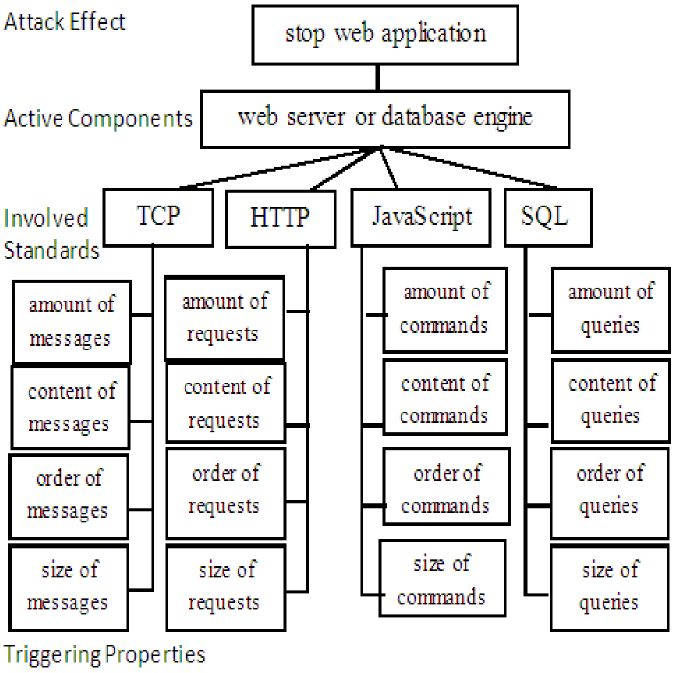


Figure 2: ATLIST Tree

VI. IMPLEMENTATION METHOD

A. Desktop based application

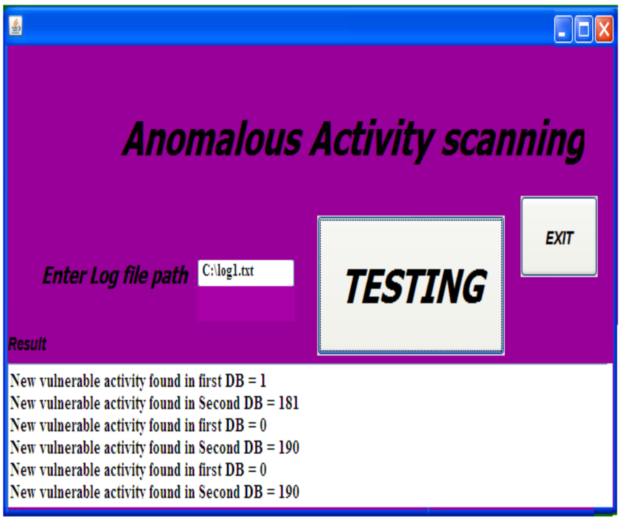
Desktop based application is simple application in which two types of scanning approaches includes are as follows-

1. Anomalous scanning

Anomalous scanning can scan system blindly and find all active services. If system finds any service having vulnerability or having any error then at run time that service will be blocked. When scanning is complete it generate log file which contain record of all running services at that moment.

2. Compare Scanning

In Compare scanning technique, two different log files are comparing. By entering path of another log file with system log file testing is done. When system stop working then that time by using compare scanning stop system log file with running system log file the vulnerable services can be find easily. Following figure 3. shows comparison of two log files.

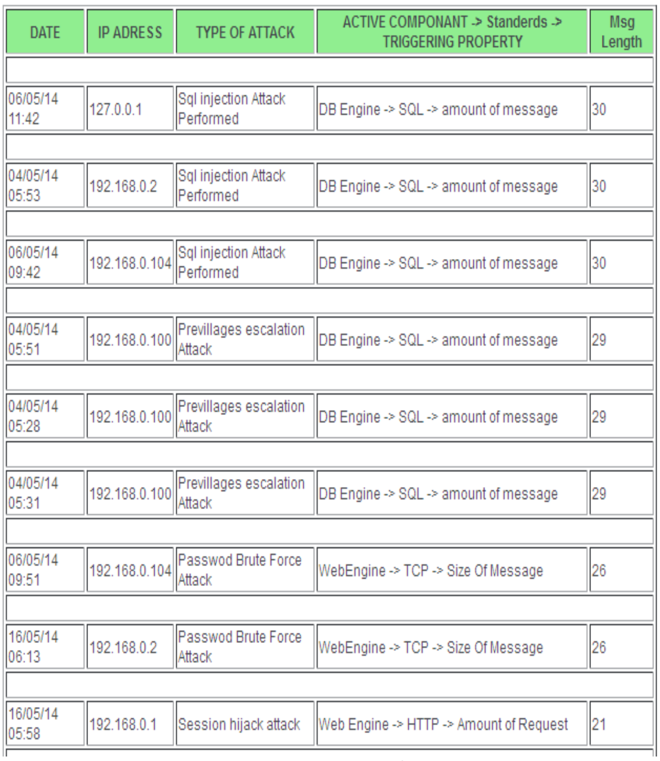


```
New vulnerable activity found in first DB = 1
New vulnerable activity found in Second DB = 181
New vulnerable activity found in first DB = 0
New vulnerable activity found in Second DB = 190
New vulnerable activity found in first DB = 0
New vulnerable activity found in Second DB = 190
```

Figure 3: Compare Scanning which generate comparison of two log file

B. *Web based application*

Web based application mostly based on two status enable and disable. When status is disabling then all vulnerability can enter into system but when status is enable then vulnerabilities are detected and that client access is denied also record is stored into Hack logs [5]. ATLIST tree is generated. When that vulnerability is deleted from block id table then that client is active otherwise access is denied for that client. Web based application for SOA based business architecture can tested multiple vulnerabilities. Using ATLIST method vulnerability analysis is takes place. It contains details of generated vulnerability like time and date, IP address, types of attack, component which is active, standard which involve and properties. ATLIST tree for vulnerability analysis is as shown in following figure 4 -

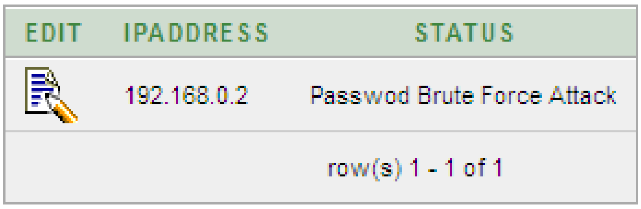


DATE	IP ADRESS	TYPE OF ATTACK	ACTIVE COMPONANT -> Standerds -> TRIGGERING PROPERTY	Msg Length
06/05/14 11:42	127.0.0.1	Sql injection Attack Performed	DB Engine -> SQL -> amount of message	30
04/05/14 05:53	192.168.0.2	Sql injection Attack Performed	DB Engine -> SQL -> amount of message	30
06/05/14 09:42	192.168.0.104	Sql injection Attack Performed	DB Engine -> SQL -> amount of message	30
04/05/14 05:51	192.168.0.100	Previllages escalation Attack	DB Engine -> SQL -> amount of message	29
04/05/14 05:28	192.168.0.100	Previllages escalation Attack	DB Engine -> SQL -> amount of message	29
04/05/14 05:31	192.168.0.100	Previllages escalation Attack	DB Engine -> SQL -> amount of message	29
06/05/14 09:51	192.168.0.104	Passwod Brute Force Attack	WebEngine -> TCP -> Size Of Message	26
16/05/14 06:13	192.168.0.2	Passwod Brute Force Attack	WebEngine -> TCP -> Size Of Message	26
16/05/14 05:58	192.168.0.1	Session hijack attack	Web Engine -> HTTP -> Amount of Request	21

Figure. 4: ATLIST Tree representation of Observed Vulnerability

Also hack log of detected vulnerabilities are generated. It contain types of vulnerability client IP address and also time and date of generated vulnerability.

When status of system is enabling and any attack is perform then vulnerability generated and it record stored into hack log as well as block ip list. That time access is denied of that particular user. If that user is valid it can get access by deleting that entry from block id list which is on database and then that system work otherwise for attacker access is denied.



EDIT	IPADDRESS	STATUS
🖉	192.168.0.2	Passwod Brute Force Attack
	row(s) 1 - 1 of 1	

Figure 5: BLOCK IP representation of Observed Vulnerability

Block list is as shown in figure 5. by clicking on EDIT option detail of that vulnerability is deleted. After deletion of vulnerability entry from this normal user can get access again into that system.

C. *Software Requirement Specification*

One of the computer programming languages is JAVA which objects oriented, concurrent and class based. Java is portable, architecture neutral and also provides high performance. Important feature of Java is exception handling which help for programmer to handle input/output errors [7] [9]. Eclipse is also very important for implementation. Main feature of Eclipse is that when user change design code get change accordingly also provide better syntax checking mechanism. Netbeans provides integrated framework for desktop based application. It is open source. Modularity is main function of Netbeans.

VII. Conclusion

Vulnerability analysis is an effective technique by which we can easily detect vulnerability present in SOA based business processes by using ATLIST method and vulnerability detection algorithm. ATLIST is useful to analyse business processes and services for checking vulnerability types. Once a vulnerability type is known it is easy to formulate a vulnerability pattern, tools can automatically locate and observe sometimes even repair or avoid vulnerability. The proposed algorithm in paper is effective for Vulnerability Detection in SOA based business application.

References

[1] C. Landwehr, B. Randell, and A. Avizienis, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans, vol. 1, no. 1, pp.11-33, Jan –March 2004.*

[2] B. Schneier, "Attack trees," Dr. Dobb's Journal, *vol. 24, pp. 21–29, 1999.*

[3] C. Anley, Adv. SQL injection in SQL server appl. http://www.ngssoftware.com/papers/ advanced_sql_injection.pdf. 2002.

[4] D. Balzarotti, M. Cova, V. V. Felmetsger, and G. Vigna. "Multi-Module Vulnerability Analysis of Web-based Application". *In preceding of ACM conference I on Computer and communications security, pages 25-35, 2007.*

[5] Fangqi Sun, Liang Xu and ZhendongSu"Static Detection of Access Control Vulnerabilities in Web Applications" *University of Caliornia, Davis.*

[6] Gary Wassermann, Zhendong Su" Static Detection of Cross-Site Scripting Vulnerabilities" University of California.

[7] *Gosling* James, Joy Bill, Steele. Guy L., Jr: Bracha, Gilad(2005) "The Java Language Specification " 3rd edition ISDN 0-321- 24678-0.

[8] I. V. Krsul,"Software vulnerability analysis," Ph.D. Work *Purdue University, May 1998.*

[9] Java 6 Programming Black Book, New Ed, Kogent Solution Inc.

[10] Rafael Accorsi and Lutz Lowis, "Vulnerability Analysis in SOA-based Business Processes" Member,*IEEE Transactions Services computing-VOL.-60, NO.-2., Aug.-2011.*

[11] Lutz Lowis and Rafael Accorsi, Department of Telematics,"On a Classification Approach for SOA Vulnerabilities" Albert-Ludwig University of Freiburg, Freiburg, Germany.

[12] Meixing Le, Angelos Stavrou, Member, IEEE, and Brent ByungHoon Kang, Member, IEEE "DoubleGuard: Detecting Intrusions in Multitier Web Applications", *IEEE Transc on Dependable and secure computing,VOL.9,NO.4,JLY/AUG 2012.*

[13] W. E. Vesely, N. H. Roberts , D. F. Haasl, and F. F. Goldberg, "Fault tree handbook," *1981, NUREG-0492.*

[14] Willy Jimenez, Amel Mammar, AnaCavalli "Software vulnerability, prevention and detection Methods:A Review "Telecom SudParis.9, Evry, *France.*

Kavita S. Kumavat She is post graduate student of computer engineering at MET Bhujbal Knowledge City, Nasik under University of Pune. Her areas of interest include Computer Networks Security.

R. P. Dahake She is currently working as Assistant Professor in Department of Computer Engineering, MET's IOE Bhujbal Knowledge City, Nasik, Maharashtra, India. She has completed her Post Graduation in Computer Engineering from Govt. College of Engineering Aurangabad Maharashtra.She has presented papers at National and International conferences and also published papers in National and International Journals on various aspects of Computer Engineering and Networks. Her areas of interest include Computer Networks Security and Embedded Systems.

M. U. Kharat, BE, MS, Ph.D. was educated at SGBA University. Presently he is working at MET's IOE, Nasik, Maharashtra, India, as Professor & Head Computer Engineering Department. He has presented papers at National and International conferences and also published papers in National and International Journals on various aspects of Computer Engineering and Networks. He has worked in various capacities in academic institutions at the level of Professor, Head of Computer Engineering Department, and Principal. His areas of interest include Digital Signal Processing, Computer Networks and the Internet.