ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 1, April 2015

SQL VULNERABILITY PREVENTION IN CYBERCRIME USING DYNAMIC EVALUATION OF SHELL AND REMOTE FILE INJECTION ATTACKS

R. Ravi[1], Dr. Beulah Shekhar[2]
Department of Computer Science & Engineering, Francis Xavier Engineering College, Tamil Nadu, India[1]
Department of Criminalogy , Manonmanium Sundaranar University, Tamil Nadu, India[2]

Abstract – *Internet crime is a general term that includes crimes such as phishing, credit card frauds, bank robbery, illegal downloading, industrial espionage, child pornography, kidnapping children via chat rooms, scams, cyber terrorism, creation and/or distribution of viruses, Spam and so on has been done through injection attacks. All such crimes are computer related and facilitated crimes. The different types of Internet crime vary in their design and easy ability to be committed. Internet crimes can be separated into two different categories. Blackmail is an illegal act that has been given as a long-established new weave in the new era. These may threaten to discharge discomforting or other damaging information through the private network or a Internet if the victim does not fulfill with the burden of the criminal. A cybercrime which may go by means of having the victim who move resources to an undetectable bank account using such type of online imbursement program, by making full use of new technology which is used to induce to commit the crime, Blackmail Hackers hack Official government websites, Online credit card scam, Work on cyber crime operations and they make money. Using code injection techniques of various levels shell and remote code injection has been familiarized and to put a stop for these type of attack by preventing its code vulnerability by dynamic evaluation of different algorithms and its comparative analysis was performed.*

Key Words: Internet crime, SQL Injection, Remote File Injection, Shell Injection.

I INTRODUCTION

In modern day, storing all backup of all files whether professional or personal should be created using SQL techniques and also used to display the data of files too. Your personal computer security is in taking the backup of files regularly is the first step towards security measures and its implementation. If your system does not support data protection software to guard online, with all difficulty get an internet security program for your computer for today's situation. Today, almost all new computer systems approach with some sort of security programs installed. Try to create a password that consists of all combination of letters (both upper case and lower case), including numbers and other special characters too. Password should be usually being changed regularly. There should not be any sharing of password with other people for security reasons.

Participating in most social networking [3] sites however, do not expose the personal information to those others and all these web pages have a definite amount of control over safety issues. Bu using these privacy situation this helps us to update and prevent the personal information that are being broadcasted. A simple rule for using the communication tool made not for open any links in emails from people that you may not know. Hackers easily do use certain E-mail as the main target seeking to steal security codes, financial data, personal information and other details. Do not use the link that is sent to you. If you need access to any webpage, then visit the webpage by typing the address in your URL string in menu bar.

As one of the growing problem that exists around the world[9], many countries are opening to execute laws and other authoritarian mechanisms in an attempt to reduce the occurrence of cybercrime. Here the computer is an important sign of mankind when entering the information age and hence the phenomenon of the cyber crime and computer was studied. Laws in many countries on usefulness of the chastisement and also to the prevention of computer crime have need of a robust number and scope of the policy, and also even the proceedings, which were

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 1, April 2015

delayed far behind the reality of command for computer crime in juridical practice.

II RELATED WORK

The objective of this research is the prevention of cyber crime using cyber laws and cyber security methods. The cyber security methods classifies accurately and efficiently detects suspicious URLS, detects malware samples and phishing websites using clustering techniques, the generation of security test to find web application vulnerabilities using balanced approach.

Here about online social networks deals which were done at the hands of spammers from end to end the lens of the tool in Retrospect[5]. As to execute the analysis over a 1.1 million financial records suspended by Twitter for unsettling activities over the duration of seven months will be identified. In this process, a dataset from database of 1.8 billion tweets, 80 million of which fit in to spam accounts that were collected. Dataset in the database will characterize the lifetime and behaviour of spam accounts, in which the campaigns were executed, and the wide-spread mistreatment of legitimate web services such as free web hosting and URL shorteners are used. An emerging marketplace of illegitimate programs operated by spammers includes Twitter financial credit sellers, ad-based URL shorteners, and spam associate programs that are enabling the underground market diversification were identified.

In [9] cross site forgery detection attacks discussed in social networks and URL shorteners, spams, phishing, and malware have become regular threats. To address better about this need, a Monarch, which was named a real-time system which crawl the URLs as they were present to web services and determine whether the direct URLs to spam which were made available or not. The feasibility of Monarch and the essential challenges that may arise due to the miscellany of web service spam was evaluated. This Monarch can provide an accurate, real-time shield and also make the underlying individuality of spam which does not simplify details across web services that are focused. In exacting situation, finding that spam across targeting such email qualitatively is different in major ways from spam movement which were targeting Twitter.

Classification [1] in Data Mining (DM) Techniques will be a very useful tool in identifying and detecting e-banking phishing websites towards Mining techniques. In this paper, the author offered a novel approach to conquer the complexity and difficulty in predicting and detecting e-banking phishing webpage. They proposed an effective model and intelligent resilient that is based on using classification and association of Data Mining algorithms. These algorithms were focused and used to characterize the identification of all the factors and some rules in categorize to classify the most phishing webpage and the association that correlate them with each other. They were implemented in six different classification algorithm and techniques which were used to extract the phishing training data sets criterion from the database to classify their legality. These were also compared by their performances, number of rules generated, accuracy and speed in classification algorithms. A phishing case study which were applied to illustrate the webpage phishing process and the rules that are generated from the associative classification model showed the association between some significant characteristics like Domain Identity, URL and Encryption and Security criteria in the final phishing detection rate. These experimental results were demonstrated by the likelihood of using Associative Classification techniques in authentic applications and its improved performance as match up to other traditional classifications procedures.

The increased in the operation volume and communication over the industries in World Wide Web like healthcare, insurance, banking, travel and many others has triggered a different number of exceptional security issues. Mostly web applications that are dealt today are susceptible to attacks ranging from virus attacks, unauthorized access, thefts of data, movement, alteration or deletion of files. The use of perimeter defenses like anti-viruses, firewalls and the likes are inadequate. Because of this, industries are seeking for further wide-ranging security measures that can be integrated in their web applications. Here are people that only intention is to break into computer networks and systems to damage them, whether it is for profit or fun. These could be trainee hackers who are seeing for a shortcut to repute by doing so and conceited about it on the

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 1, April 2015

internet usages. Here by, these could also be a collection of prepared criminals who put their effort silently on the lead. They don't make sound but when their job is concluded, it reflects into a enormous loss for the organization in inquiry – not to state a enormous yield for such criminals.

Existing System

With so many techniques and approaches in testing the security of web applications came into existence, where it can be difficult to understand which system to use and when and how to use those security measures. Experience shows with the intention of there is no right or wrong respond to exactly what technique should be build a testing framework. In all techniques the fact shall remain should almost certainly be used to make sure that all areas that require to experience are completely tested. It is very clear, that there is no particular technique that successfully covers all security testing that can be performed to ensure that all problems have been addressed. Many group adopt one approach, which has in history been in penetration testing. These testing while useful, cannot efficiently tackle many of the concern that necessitate to be hardened and is simply "too little too late" in the software development life cycle (SDLC). There are circumstances and times where only one method is possible, for example, a test on a web submission that has already formed, and where the testing phases does not have contact to the basis code.

Disadvantages of Existing System

- Testing in initial or end of product development
- Does not use all available security features
- Inefficient
- Does not find all vulnerabilities
- Use only known vulnerabilities for testing

III PROPOSED METHODOLOGY

Proposed System

An enclosure of defense which will obviously decrease vulnerabilities in web applications is seen to be in progress lifecycle of the request itself. Developers look at the vulnerabilities and need to be trained that could possibly take place in web applications so that preventive measures can be adopt in the completion stage. The proposed system[7] serves as an uncomplicated guideline for all those concerned in the application's expansion procedure and with the additional important designs and

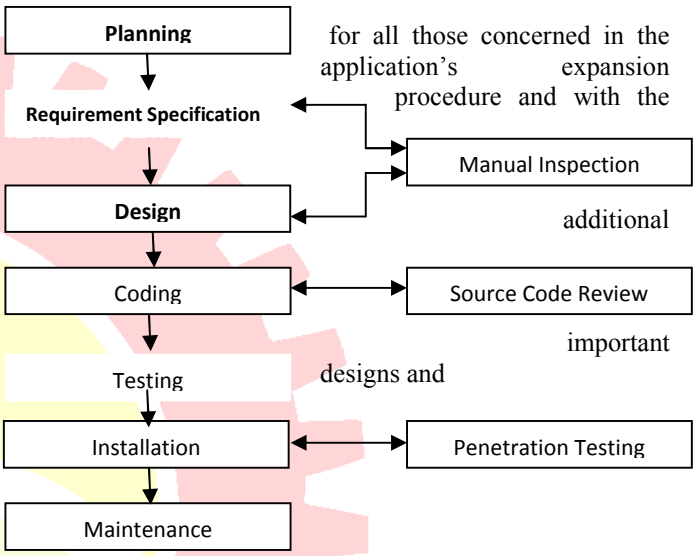


Fig 3.1 Software Development Life Cycle

formulate a set of secure coding policies guidelines as pro-active remediation strategy to reinforce the security of web applications.

Besides the implement SDLC methodology to we can design a new manufacture model web site and testing the conservatory website which recently published hosted. These are balanced move toward that includes a number of techniques, from physical interviews to practical testing. The balanced approach is convinced to envelop testing in all stage of the SDLC. This move toward by leveraging the most appropriate techniques obtainable depending on accessible SDLC phase. A balanced approach may differentiate depending on different factors, such as the ripeness of the trying out with the test process and corporate culture. Fig 6.1 shows the proposed system architecture.

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 1, April 2015

Advantages of Proposed System

- Testing covers all phases of Software Development
- Developers or analyst must be aware of web application vulnerabilities
- Finds all security weakness while development
- Removes all kinds of vulnerabilities by combining different techniques.
- Testing generated by different techniques has high secured function.

V ALGORITHM USED

Finding SQL Injection vulnerability

Mostly many web applications have the authentication to t he users for accessing those services. Our applications also have the authentication to allow only the registered users. For example, Fig. 5.1, shows the code for the authentication,

ResultSet r=st.executeQuery ("select * from registration where email= ' "+request.getParameter("uname")+" ' "+"And Password=' "+getParameter("psw") + " ' ");

Fig 5.1 Dynamic Generation of SQL Query

In the code, the email and password has been checked for the registered table 'Registration'. If the password and email matches from the table, the application allows the user to access the services. The flaw in the application is it generates the SQL Query dynamically. The email and password is checked by the SQL Query 'AND' at run time. The Dynamic Generation of the SQL Query allows the attacker to implement the malicious code like "abc' OR 1=1--". Fig. 5.2 shows how the SQL Injection attack has been done. If the attacker knows only the username, the above malicious query is put to the password field to make the SQL Injection. For example, the username is 'parthi88@hotmail.com' and the password is 'abc' OR 1=1--'. The code executes the password field as true. Because 1=1 is always true. The operation is done by the 'OR' SQL Command. It is the logical operation which makes the SQL command true. Fig. 5.2 shows the algorithm to find SQL Injection Attacks.

Begin
Check for SQL Evaluation
Get the Username
Give password as "abc' OR 1=1--"
If (Authentication==Success)
Declare The Application is Vulnerable to SQL Injection
End If
End

Fig 5.2 Algorithm for Finding SQL Injection

Preventing SQL Injection Attack

Preventing addition of SQL injection requires observance of independent data separate from queries and commands. This preferred alternative is to utilize a safe API which avoid the use of the predictor or interpreter wholly or provides a parameterized interface. Beware and careful of APIs, such as stored procedures that are being parameterized but can still bring in injection under the cover.

Finding HTML Script Injection

Review is the operation in our E-banking Websites. Like a Twitter, any users can post a message in the review page to other users. This is the logical flaw in E-banking Application, because the attacker could be a user who can put any malicious post to the review page. If the victim clicks the link the malicious post can send the user to any harmful sites. Figure 5.3 shows the review page. From this, the attacker sends the malicious link with fascinating title like 'Offer' and Hurry. If the victim clicks the link, the malicious page could steal the session ID or do any other harmful operations to the victim. Fig. 6.4 shows one message to the victim from the review page. When the victim clicks the 'More Details....' the victim will be dragged to the malicious website to steal the session id. After the Victim clicks the malicious link, session id is stored to the new malicious site. Fig. 5.3 shows the algorithm to find HTML Script Injection.

Begin
// Find HTML Script Injection
Check for possible HTML Link by user
If any user can post a link

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 1, April 2015

Declare the Application is Vulnerable to HTML Script Injection
End If
End

Fig 5.3 Algorithm to find HTML Script Injection

Preventing HTML Script Injection

To prevent injection attack the only straightforward solution for the user is to disable and stop all script languages in computer programming techniques. Unfortunately, it is highly probable that much functionality of the location regularly visited will be detached. Thus the users are supposed to only follow this option if they necessitate the lowest likely level of request. Alternatively, users must be discriminating as to the sites in which they trust, and the follow up of resources in URL links. Here again, the halting of scripting language techniques will not stop for attackers influencing the manifestation of content provided by confident sites by embedding other HTML label in the URL link which was forwarded.

With scripting which was executed, visual examination of links does not defend users from subsequent malicious links, since the attacker's web location may still use tagged code to change the demonstration of the links in the customer browser. Here unfortunately many incorporated applications enlarge the risk of scripting code being implemented on the users system, which was particularly through the use of the file in which embedded objects such as Flash! As .swf files. To prevent these categories of attacks, users have to uninstall the interpreters or ensure defense scheme that are capable of stopping the implementation of such content which has been pleased. It is visualize that popular anti-virus and any other personal intrusion uncovering systems will eventually be competent for doing such activity like this.

Dynamic Evaluation Injection

When the application calculates the output at the runtime, the attacker can send their own input to make the malicious operations. For example, when the E-Banking Web Application sends the money from one user to another user, the attacker can modify which user can send and which user can receive. Fig. 5.4 shows the Dynamic Evaluation while transferring money.

In Fig. 5.4, the address bar from the browser shows the internal operation and the variables used for operation. This is flaw from bad coding because the sensitive information send from the browser to server, the HTTP GET methods should not be used.

Now the attacker can get the link from the browser History, and modifies the sender or receiver from the browser address bar. If the attacker modifies the sender from 'Parthiban' to 'Senthil' this application still works and sends the money 100 from Senthil to Karthick. Senthil Account Password is not known, but the account name must be known to the sender. Figure 5.4 shows the result of Dynamic Evaluation Injection. Figure 5.4 explains the algorithm to find dynamic evaluation injection.

Begin
Get the input from HTTP method
Check for dynamic operation
Set the input as 'attacker account number'
Send the malicious link by HTTP Method
End

Fig 5.4 Algorithm to find Dynamic Evaluation Injection

Preventing Dynamic Evaluation Injection

Using API properly to protect against all input characters we are approaching this. Parameterized queries (otherwise called as "bound variables", "Compiled queries", "prepared statements") allows for stirring user data out of string to be taken to mean and identify its measure. Additionally criterion on API and similar API's shift away from the notion of authority strings to be interpreted and created.

Enforcing separation of languages via a Stream of static type has been followed Input justification, such as White listing only accepting known good values

Finding Remote File Injection Vulnerability

When the server sends the file to the client through HTTP Protocol, Remote File Injection is Possible. In the E-banking example, the branch addresses are stored in the file. When the client tries to access the branch address, the html file contains the address which has to be

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 1, April 2015

sent to the client. In Fig. 6.6, the browser address bar contains the file name 'branchoffice.html' which is going to execute when the user clicks the 'Other Branches link'. Thus the attacker uses this link and uses his own file which is going to execute in the server. Fig. 5.5 shows the result of File Injection Vulnerability and explains the algorithm to find the remote file using injection method.

Begin
// Find File Injection method
Get the HTTP method
Check for External file
If the application has External file
Set the malicious file instead of external file at HTTP method
End If
End

Fig 5.5 Algorithm to find Remote File Injection

Preventing Remote File Injection

Remote file inclusion (RFI) attacks should not be possible - yet they are all often too. The RFI is a cousin to the nefarious XSS cross-site scripting attack. Both are forms of code injection, although the RFI is less sophisticated. As such, it can also more easily prevented. Without a solid RFI defense, though, even this simple attack vector can be used to commit a wide range of malice, everything from defacing or deleting the content of your website to lifting sensitive data directly off your web hosting server.

Finding Shell Injection Vulnerability

Shell Injection is also called as Command Injection. It is executed by the malicious OS commands. If the applications use the OS commands for the execution of its purpose, the attacker can get the advantage of the OS from the malicious commands. The following algorithm shows how to find the vulnerability for Shell Injection. Fig. 5.6 explains the algorithm to find shell injection attacks.

Begin
Check the whole application for execution of OS command
If possible
Set the input as 'OS command line'
Endif
End

Fig 5.6 Algorithm to find Shell Injection

Preventing Shell Injection Attack

Despite the innumerable ways that was described above to challenge shell injection, it can be disallowed a few uncomplicated steps. Top among these is to carefully clean all user key in data or information. If the user given arguments are avoided by passing to the OS programs, it should be seriously consider. Alternately, be confident to strip out potentially harmful characters such as separators, semicolons or other which can be used to run as extra commands. In UNIX, this includes and ampersand (&), pipes (|) which has been utilized. The finest way to accomplish this is by means of a white list is, For the filename sample given above, which preserve a list of acceptable files and make sure that the input contest an entry in this catalog exactly. The whole thing whatever needs to be discarded on the other hand as an unsafe operation.

Alternately, OS commands can be wrapped in more protected language, such as Java. If the Java exec command is run, it passes each command given as a separate parameter, by eliminating some of the most general injection vectors. Fundamentally, the shell insertion vectors are firm like other injection attacks to thwart since justifiable input may be very parallel to invader input. The key technique is cautiously sanitizing information, white-listing statistics, and prevents shell instruction from being passed during execution from user input wherever it is possible.

VI EXPERIMENTAL SETUP

Fig. 6.1 to Fig. 6.4 explains the various attacks and the number of vulnerabilities from Automated Generation of

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 1, April 2015

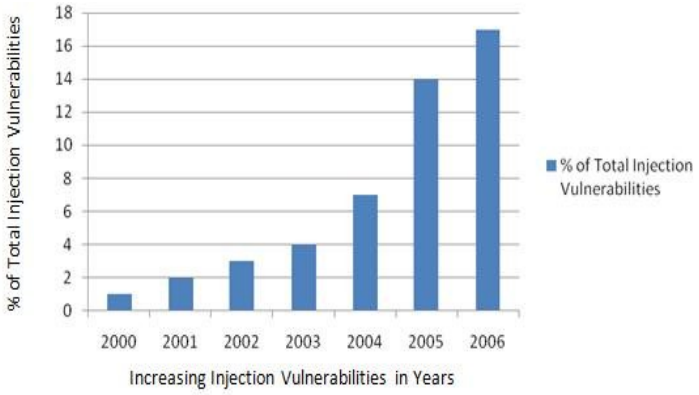


Fig 6.1 SQL Injection per years

Security Testing by Thread model It defines the number of attacks by OWASP Top 10 Security weakness in the year of 2010. Fig 6.1 SQL Injection as percent of total vulnerabilities. Fig. 6.1 explains the increasing injection attacks from the year of 2000 to 2006 by Web Application Scanners: Definitions and Functions. It defines how the injection attacks increases from year to year

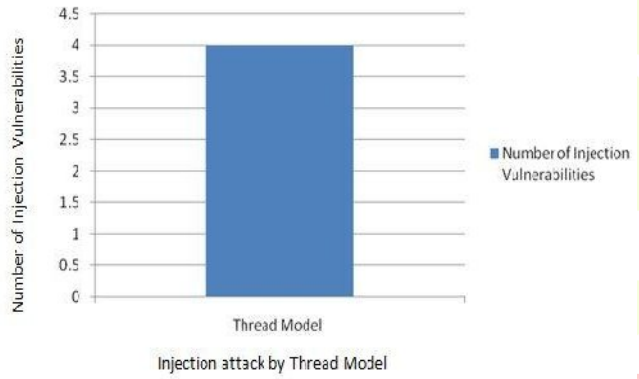
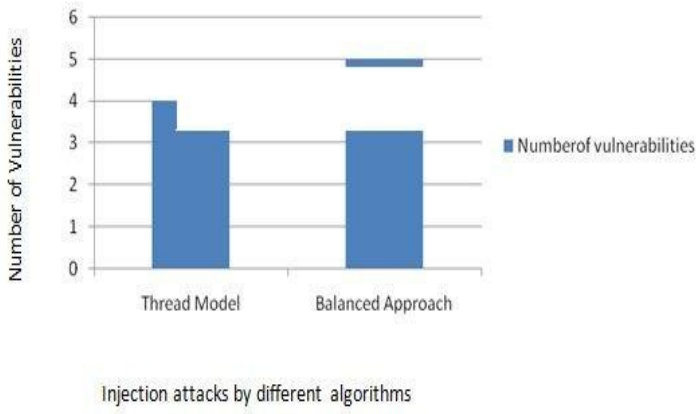


Fig 6.2 Injection Vulnerabilities

Fig. 6.2 shows the number of vulnerabilities by Thread Model. The number of injection found by Thread Model is 4. The Thread Model Testing method does not have the functionality to find the Shell Injection. We can use Thread



Model to find SQL Injection, HTML Script Injection, Dynamic Evaluation Injection and File Injection.

Fig 6.3 Vulnerabilities in Injection using Balanced Approach

Fig.6.3 shows the comparison between the Thread Model and Balanced Approach. The Balanced Approach Testing Technique has the capability to find the Shell Injection. So we can find 5 kinds of Injection vulnerabilities by using Balanced Approach.

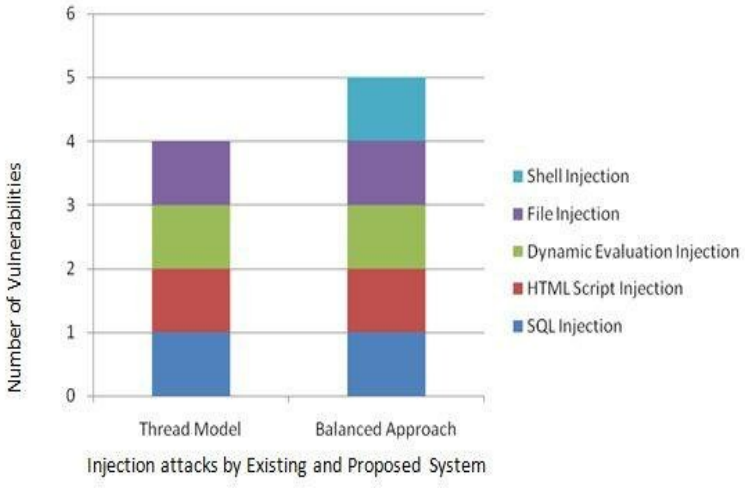


Fig 6.4 Comparison of Existing and proposed System

ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 1, April 2015

Fig. 6.4 defines the different injection vulnerabilities from existing system to proposed system. It represents the various injection attacks such as SQL Injection, HTML Script Injection, Dynamic Evaluation Injection, File Injection and Shell Injection. Fig.6.3 shows the comparison between the Thread Model and Balanced Approach. The Balanced Approach Testing Technique has the capability to find the Shell Injection. So we can find 5 kinds of Injection vulnerabilities by using Balanced Approach.

CONCLUSION

Here a balanced approach is used for finding the injection vulnerabilities in web applications. This approach helps to find more vulnerabilities than any other approaches used for such as Shell Injection. This helps to have more security for our web applications. In other approaches they used their testing approach only at the beginning or at the end of software development. But Software Development Life Cycle (SDLC) phases are used in all stages. Here the test is done manually so it consumes some more time for developer and tester. In future to make this testing approach more efficient, this balanced approach will be made automatic.

REFERENCES

1. Aburrous M., Hossain M.A., Dahal K. and Thabtah K. (2010), 'Predicting phishing websites using classification mining techniques with experimental case studies' in Proc. 7th Int. Conf. Inf. Technol., pp. 176–181
2. Avinash Kumar Singh and Sangita Roy (2012) 'A Network Based Vulnerability Scanner for Detecting SQL Attacks in Web Application' 1st Int'l Conf. On Recent Advances in Information Technology |RAIT-2012|
3. Bailey M., Oberheide J., Andersen J., Mao Z.M., Jahanian F., and Nazario J. (2007), 'Automated classification and analysis of internet malware' in Recent Advances in Intrusion Detection, Vol. 4637, pp. 178–197.
4. Jason Bau, Elie Bursztein, Divij Gupta, John Mitchell (2010) 'State of the Art: Automated Black-Box Web Application Vulnerability Testing' Stanford University 2010 IEEE Symposium on Security and Privacy.
5. Dolatabadi, H. , Shirazi, M.N. ; Hejazi, M.(2011) New mechanism to confront injection attacks.
6. Guo Yucheng, Wu Peng, Lin Juwei, Guo Qingping (2011) A Way to Detect Computer Trojan Based on DLL Preemptive Injection Distributed Computing and Applications to Business, Engineering and Science (DCABES), Tenth International Symposium.
7. El-Bahlul Fgee., Ezzadean H.Elturki and A.Elhounie (2012) 'My Security for Dynamic Websites in Educational Institution' Sixth International Conference on Next Generation Mobile Applications, Services and Technologies
8. Herley C. and Florencio D. (2008), 'A profitless endeavor: Phishing as tragedy of the commons' in Proc. New Security. Paradigms Workshop.
9. Hossain Shahriar and Mohammad Zulkernine (2010) 'Client-Side Detection of Cross-Site Request Forgery Attacks' IEEE 21st International Symposium on Software Reliability Engineering
10. Sadeghian, A., Zamani, M., Manaf, A.A., (2013) 'A Taxonomy of SQL Injection Detection and Prevention Techniques,' pp.53, 56.
11. Naaliel Mendes, Afonso Araujo Neto, Joao Duraes, Marco Vieira and Henrique Madeira (2008) 'Assessing and Comparing Security of Web Servers' 14th IEEE Pacific Rim International Symposium on Dependable Computing
OWASP Testing Project (2013)
12. https://www.owasp.org/index.php/ Testing _Guide_ Frontispiece