International Journal of Computer Theory and Engineering, Vol. 2, No. 5, October, 2010
1793-8201

Mitigation of Web Based attacks using Mobile Agents in client side

E. Rajesh, R.Raju and R.Ezumalai

Abstract—**Web applications are becoming the dominant way to provide access to on-line services such as e-commerce & e-banking. Attackers have found new type of attacks to exploit vulnerabilities in these web applications. Among these attacks reflected web parameter attacks have received much attention in the recent scientific literature. Efficient mechanism such as Input validation is an addition overburden scenario for executing a secure transaction in a server side. This paper presents a new, highly automated approach that prevents web parameter attacks using mobile agents in the client side. Our Proposed system has to extract keywords from the web application parameter in a client side and use the gene alignment approach to measure the identity between two parameter sequences in order to avoid web attacks. Hirschberg algorithm is an divide and conquer approach for measuring the relevance between two sequences and it has advantageous over other methods in order to reduce the time complexity O((nm)) and space complexity O(min (nm)). This system was able to stop all of the successful attacks and did not generate any false positives.**

Index Terms—**Web attack detection, Intrusion Prevention System, Web Application Parameter, Hirschberg Algorithm, Browser Security, Mobile agents.**

I. Introduction

Many web applications have been purely based on databases which widely incorporated on the Internet and firms use these services to provide a broad range of services to customers. Web based parameter attacks has been severely exploit the valuable web applications. Database are the core of many information system, reason for which database are increasingly coming under large number of attacks. In poorly configured or designed web application, malicious users can modify the parameters, session tokens or values stored in cookies and even HTTP headers.

Developers show keen interest in developing the application with usability rather than incorporating security policy rules. SQL Injection attacks, Cross site scripting attacks and Path disclosure attacks are the most dangerous attacks which will breach the integrity and logic validation mechanism errors. Malicious user can breach the security of the application and perform illegal operation for their own benefit or an attacker who wishes to attack another person using a man-in-the-middle attack.

Cryptography techniques in the transport layer can never protect the parameter manipulation attack in which data is mangled before it hits the wire [2]. Cookies, form fields, URL query strings, HTTP Headers are some of the parameter tampering holders which exploits the web application. Even the Web application attacks can bypass the security mechanism such as Firewall, cryptography and traditional intrusion detection systems. The most worrying aspect of web application attacks is; it is very easy to perform, even if the developers of the application are well known about these types of attacks.

A. Cookie Manipulation

During normal operation cookies are sent back and forth between a server and the user computer. Since cookies may contain sensitive information (user name, a token used for authentication, etc.), their values should not be accessible to other computers. Cookie theft is the act of intercepting cookies by an unauthorized party. Both persistent and non-persistent cookies, secure or insecure can be modified by the client and sent to the server with URL requests. Therefore any malicious user can modify cookie content to the advantage for access the information.

B. HTML Form Field Manipulation

When a user performs any action on HTML page, the action is stored as form field values and sent to the server as an HTTP request. Hypertext mark up language can store form field values as Hidden Fields and it is not rendered to the screen by the browser but are collected and submitted as parameters during form submissions [5]. Web site developers prevent the user to inject buffer overflow to set some minimum value to the form parameters. However the attacker can save the page, remove the length tag and do some malfunctions and reload or reset the page.

C. URL Manipulation

URL manipulation is a most dangerous attack used by hackers to perform some manipulation in the Uniform Resource Locator parameters. URL manipulation can be employed as a convenience by a Web server administrator or for nefarious purposes by a hacker [3].URL manipulation is that redirecting the user request from correct site to illegitimate site. URL manipulation differs from URL

Manuscript received September 8, 2009.

E. Rajesh is with the Department of Computer Science and Engineering, Sri Manakula Vinayagar Engineering College (Affiliated to Pondicherry University, Puducherry), Madagadipet, Puducherry -605 107, India.
Phone: +91 9943998866, +91 9629455044

R. Raju is with the Department of Information Technology, Sri Manakula Vinayagar Engineering College (Affiliated to Pondicherry University, Puducherry), Madagadipet, Puducherry -605 107, India.
Phone: +91 9442029188

R. Ezumalai is with the Tata Consultancy Service, Chennai, India.
Phone: +91 9345486422

poisoning, also known as location poisoning. That is a method of tracking Web user behavior by automatically adding an identification (ID) number to the URL line of the Web browser when a user visits a particular site. This ID number can then be used to determine which pages on the site the user visits thereafter.

D. HTTP Header Manipulation

HTTP headers manipulation is the attack used by attackers to manipulate the header information which has been passed from web clients to web servers on HTTP requests, and vice-versa on HTTP responses. Header normally consists of a single line of ASCII text with a name and a value [7]. Modifying the header information in the web browser is not quite simple. An attacker have to write his own program to perform the HTTP request, or he may use one of several freely available proxies that allow easy modification of any data sent from the browser.

The remainder of the paper is organized as follows: Section 2 contains related work; Section 3 describes our proposed approach, Section 4 describes the conclusion and future work.

II. Related Work

Traditional security mechanism such as intrusion detection system, firewall and encryption methods are not sufficient to block web based parameter attacks. Web based attacks are most dangerous issues which exploit the valuable web application. Countering threats to an organization's internal databases from database applications is an important area of research. Input validation mechanism is an extra burden to the server side for complete secure transaction. Many techniques has been discussed to prevent web parameter attacks are followed.

Christopher Kruegel et al [1], propose an intrusion detection system that uses a different anomaly detection technique to find attacks against web applications. The system correlates the server-side programs referenced by client queries with the parameters contained in these queries. The application-specific characteristics of the parameters allow the system to perform focused analysis and produce a reduced number of false positives.

Jae-Chul Park et al [2] propose a mechanism to detect SQL injection attacks using pair wise sequence alignment of amino acid code formulated from web application parameter.

YongJoon et al [3] propose an Anomaly Intrusion Detection model to find out the input validation attacks against web applications. WAIDS derives a new intrusion detection method using generated profile from web request data in normal situation.

Jae-Chul Park et al [4] propose a mechanism to provide a solution against web parameter manipulation attack using genome sequence. In their work, Needleman wunsch algorithm has been used to compare the web parameter with the predefined values to find out the attacks and the results.

Benferhat Salem et al [5] propose a mechanism for detecting Web attacks targeting either server-side or client-side applications. Request general features, Request content features, Response features and Request history features are the four kinds of features provided. Experimental studies discussed in this paper show the efficiency of the selected feature set in detecting Web related attacks

Sanghyun Cho et al [6] propose the Bayesian parameter estimation method and it is very effective in analyzing web logs and detecting anomalous sessions. Snort, a well-known IDS based on misuse detection, caught only slightly more than one third of web attacks. But, Session anomaly detection (SAD), detected nearly all such attacks without having to rely on attack signatures at all. SAD works by first developing normal usage profile and comparing the web logs, as they are generated, against the expected frequencies.

R. Faradzhullaev [7] propose a mechanism to analyze log files and find out the anomalous changes that take place on the web server and identifying attacks. The proposed method allows detecting anomalous queries received from malicious users in log files of the web server.

Författare [8] use the mechanism to alert from several audit sources to improve the accuracy of the intrusion detection system (IDS). A theoretical model has been designed to automatically reason about alerts from different sensors, thereby also giving security operators a better understanding of possible attacks against their systems.

Livshits and Lam [9], propose a mechanism for finding the SQL injection using vulnerability pattern approach. Vulnerability patterns are predefined pattern to find out the attacks. The main issue of this method is that it cannot detect the SQL injection attacks patterns that are not known beforehand.

Marco Cova et al [10], proposed a method to the anomaly-based detection of attacks against web applications. Swaddler analyzes the internal state of a web application and learns the relationships between the application's critical execution points and the application's internal state. The main drawback is the overhead grows linearly as the number of executed basic blocks increases. This is due to instrumentation and detection overhead associated with each basic block in the program.

William G.J. Halfond, Alessandro Orso, Panagiotis Manolios [11], proposed the mechanism to keep track of the positive taints and negative taints. It will compare the SQL statement with this taints and if it finds any suspicious activity, it will generate the alarm. The advantage of this mechanism, that it does not require any modification of the run time system even at application level and imposes a low execution overhead.

Many authors proposed different techniques to prevent web parameter attacks. But all these methods reported to have a lot of pros and cons of its own proposal. The authors classified their mechanism as signature method, anomaly method and auditing method.

In Signature Based Method, a threat is always be stored in database. This Mechanism finds very difficult to detect the new threat. In Anomaly Based Method, it monitors system and its web application behaviors. It set the baseline of network and system. But this mechanism generates more false alarm rates. In Auditing and Logging Method, it provides a limited audit functionality of database management systems (DBMS). Inconsistency across DBMS

types and the performance penalty are drawback of this system.

In this paper, a new attempt has been proposed and worked out effectively against web parameter based attacks. New technique such as Mobile agent has been incorporated in this proposal in order to reduce the work burden of the server side. Mobile agent gathers the Hirschberg algorithm to the client side for further processing and if it find any suspicious activity, it block the further transaction and shows warning to the user. If its suspects a malicious attacks, it blocks the further transaction and it avoids the network usage, traffic and bandwidth. Hirschberg algorithm is a gene alignment approach for measuring the identity between two different gene sequences. This paper incorporates this algorithm for exact match with incoming keywords with predefined keywords and also reduced the time and space complexity. Even in signature based method, all types of attacks are stored in database and it compares the incoming attacks with this predefined attacks. There are two issues is present in this signature based method. The first issue is, this mechanism detect the attacks only which is present in the database. The second issue is, this mechanism takes more time and space complexity. This paper discussed an approach could analyze and compare each and every keyword with predefined keyword for effective way of preventing attacks and also to reduce the time and space complexity. In addition to that, it reduces the work burden of the server and also it avoid unnecessary transaction between client and server if its anomaly transaction.

III. Our Approach

Our approach is entirely based on Signature based method, which has been used to address security problems related to web based attacks. Mobile agent plays a vital role to establish the secure transaction and also reduce the work burden of the server side. Mobile agent gathers the overall source code of the algorithm and it travels to the client side and it check for the secure transaction. If it finds any suspicious activity, it blocks the transaction as well as it reduces the network throughput and transmission. There are four modules incorporated which is used to detect the security issues. Agent module has gathered the core algorithm and it brings to client side for further processing. Decision module has gathering the web application parameter from the web application and it decides whether it can send the statement to network for execution. Examiner module uses Hirschberg algorithm to find out the attacks. Signatures comprise the predefined keywords and identifier and send it to Examiner module for comparisons. In analysis module, if it finds any suspicious activity, it acts as an active agent to stop the transaction and audit the attacks.

A. Agent Module

Mobile Agent is a software agent, with the feature of autonomy, social ability, learning, and most importantly, mobility. Mobile agent is a process that can transport its state from one environment to another, with its data intact, and be capable of performing appropriately in the new environment. Mobile agents decide when and where to move the location. This mobile agent gathers the information from the server to client and it will send to Decision Module.

B. Decision Module

In Decision module, it's the in-between module which can get a parameter from the web application and sends it to examine module. Decision module has to decide that this statement can send it to network

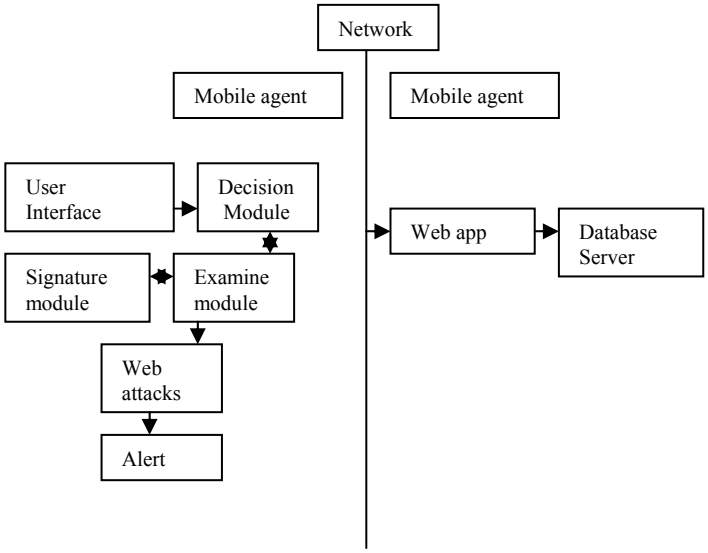

Figure 1: Prevention of Web attacks using Hirschberg Algorithm

C. Hirschberg Algorithm

Hirschberg Algorithm is a dynamic programming algorithm that finds the least cost sequence alignment between two strings. Hirschberg's algorithm is a divide and conquer version of the Needleman-Wunsch algorithm. Hirschberg's algorithm is commonly used in computational biology to find maximal global alignments of DNA and protein sequences and this proposal incorporated this algorithm here to find out the similarities. Hirschberg's algorithm is a generally applicable algorithm for finding an optimal sequence alignment. One application of the algorithm is finding sequence alignments of DNA or protein sequences. This algorithm mainly used for reducing the time and space complexity of the system.

If x and y are strings, where |x| = n and |y| = m, Hirschberg's algorithm is a clever modification of the Needleman-Wunsch Algorithm which takes O(nm) time, but needs only O(min{m,n}) space for comparing these two x and y strings. This is the formula to plot the values in the table.

TABLE 1: Hirschberg Algorithm

F(i,j)	A	C	C	T	G
A	F(i-1,j-1)	F(i,j-1)+p) p–gap penalty			
T	F(i-1,j)+p p–gap penalty				

$$F(i,j)= \text{Max}\{F(i-1,j-1)+t(xi,yi), F(i,j-1)+px, F(i-1,j)+py)\}$$

$$F(i,j)=1 \quad \text{if}(xi=yi)$$

F(I,j)=0 if(xi≠yi)

t(xi,yi) - score for aligning the characters at positions i and j,

p is the penalty for a gap.

F(i,j) is a type of running best score as the algorithm moves through every position in the matrix. But in our approach, gap penalty has ignored. If Xi=Yi, then plot "1" else "0" till all the character has been visited.

For example, this is sample URL parameter attacks,

http://www.bank.com/svepge.asp?tr=100&status=read

Modifying the status variable to delete the page:

http://www.bank.com/svepge.asp?tr=147&status=del

The following table clearly explains the approach system to prevent such attacks. Hirschberg algorithm uses divide and conquer approach in order to reduce time and space complexity. X is the predefined statement which has been stored in signature module and Y is the statement which has been get it from Decision module and it compares using this algorithm. It divides the problem in to two sub problems, from tr to status and bank to svepge.asp. First it compares the first sub problems and it find out the attacks and it wont go for other sub problems and it sends the notification to Decision module to stop the transaction and audit the attacks.

TABLE 2: HIRSCHBERG ALGORITHM TO FIND OUT THE ATTACKS

Y	Bank	Svepge. asp	tr	&	Status= read
bank	1	0	0	0	0
Svepge.asp	0	1	0	0	0
		Divide	and	conquer	
tr	0	0	1	0	0
&	0	0	0	1	0
Status=del	0	0	0	0	0

IV. RESULTS

This system can tested with three types of web application such as portal, classified and Events. There are many number of attacks are present in each section such as buffer overflow, cross site scripting attacks and SQL Injection attacks and find out the proposed system ability to stop the attacks. The following table clearly explains the advantages of the proposed system over existing methods are shown. Consider the inputs from the web application are suspicious and behavior of the system show in this table.

TABLE 3: COMPARISON OF PROPOSED METHOD WITH EXISTING METHOD

Techniques	Network Usage	Server Side Burden	Client Server Interaction
Signature Based Method.	Yes	Yes	Yes
Anomaly Based Method	Yes	Yes	Yes
Database Auditing Method	Yes	Yes	\Yes
Proposed System	No	No	No

This system not only blocks the web application and it stops some of the network security issues. Consider this limited number of attacks and tested with this proposed system to find out the attacks and got a encourage results.

TABLE 4: EFFICIENCY OF THE PROPOSED SYSTEM

Subject	Total no of attacks	Successful Prevention of attacks
Portal	3000	0
Classifieds	5088	0
Events	6004	0

This system act as an Intrusion prevention system to detect and prevent the web application attacks. But the drawback of existing Intrusion prevention system can generate the more false alarms, but it may work efficiently. This system can able to stop the attacks as well as it could not generate the false alarms and it work effectively against the web parameter attacks. Consider this limited number of access and tested with this proposed system to find out the alarm rates.

TABLE 5: EFFICIENCY ALARMS OF THE PROPOSED SYSTEM

Subject	False Positives	
	Legitimate access	False Positives
Portal	1,359	0
Classifieds	424	0
Events	658	0

This system can work efficiently against web application attacks and it also reduces the time and space complexity of the system. Again the system can tested with number of input in worst case and Best case scenario access time. In worst case scenario, it takes time complexity as O(nm), but in best case scenario, it takes time complexity as O(min(nm)) where n is the predefined keywords and m is the incoming keywords. But it reduces the space complexity as O(min(nm)).

TABLE 6: EFFICIENCY OF TIME AND SPACE COMPLEXITY OF THE PROPOSED SYSTEM

Subject	#inputs	Worst case Avg access Time(ms)	Best case Avg access Time(ms)	Space complexity
Portal	1,359	122	61	O(min(nm))
Classifieds	424	56	28	O(min(nm))
Events	900	63	31	O(min(nm))

V. CONCLUSION

This paper proposed a system to protect the web application using mobile agents from web parameter based attacks. Our approach using mobile agent as a core to protect the web application as well as reduce the burden of the web database transaction. Our mechanism also provides advantages over the other existing techniques whose

application requires customized and complex runtime environments. Hirschberg algorithm plays another part and it uses divide and conquer approach to detect the web based attacks in order to reduce the time and space complexity. Traditional security mechanism such as IDS and firewall have not been sufficient to provide the security of web application, however, this mechanism is able to block abnormal approach to web application and to detect previously unknown attacks as well as variations of known attacks.

REFERENCES

[1] Christopher Kruegel ,Giovanni Vigna, "Anomaly detection of web-based attacks", Conference on Computer and Communications Security, Pages: 251 – 261, ACM, Year of Publication: 2003
[2] Bong-Nam Noh, Jae-Chul Park, " SQL Injection Attack Detection: Profiling of Web Application Parameter Using the Sequence Pairwise lignmen", Information Security Applications, pages 74-82, ACM, 2007.
[3] YongJoon, Park, JaeChul, " Web Application Intrusion Detection System for Input Validation Attack", Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference, Volume:2, On page(s): 498-504, 11-13 Nov. 2008.
[4] Jae-Chul Park, Bong-Nam Noh, "Detection of Parameter Manipulation Using Global Sequence Alignment", Next Generation Web Services Practices, 2006. NWeSP 2006. International Conference, page(s): 83-88, 25-28, Sept. 2006.
[5] Benferhat Salem,Tabia Karim, "Classification features for detecting Server-side and Client side Webattacks" ,Proceedings of The Ifip Tc 11 23rd International Information Security Conference, Volume 278/2008, July 17, 2008.
[6] Sanghyun Cho, Sungdeok Cha, "SAD: web session anomaly detection based on parameter estimation", Elsevier, 20 February 2004.
[7] R. Faradzhullaev, "Analysis of web server log files and attack detection", Automatic Control and Computer Sciences,, pages- 50-54, Springerlink, Friday, April 18, 2000.
[8] Författare, " A Multi-Sensor Model to Improve Automated Attack Detection", CHALMERS, 2008.
[9] V.B. Livshits and M.S. Lam, "Finding Security vulnerability in java applications with static analysis", In proceedings of the 14th Usenix Security Symposium, Aug 2005.
[10] Marco Cova, Davide Balzarotti, Viktoria Felmetsger, and Giovanni vigna, " Swaddler: An approach for the anamoly based character distribution models in the detection of SQL Injection attacks", Recent Advances in Intrusion Detection System, Pages 63-86, Springerlink, 2007.
[11] William G.J. Halfond, Alessandro Orso,Panagiotis Manolios, "WASP: Protecting Web Applications Using Positive Tainting and Syntax-Aware Evaluation",IEEE Transaction of Software Engineering Vol 34, No1, January/February 2008.
[12] Z. Su and G. Wassermann, "The Essence of Command Injection Attacks in Web Applications",33rd ACM SIGPLAN SIGACT Symposium on Principles ofProgramming Languages, Charleston, South Carolina, USA, 2006, pp. 372-382.
[13] Xin Jin, Sylvia Losborn, "Architecture for data collection in database intrusion detection system", Secure data management, Springerlink, 2007.
[14] N. Jovanovic, C. Kruegel, and E. Kirda, " Pixy: A Static Analysis tool for detecting web application vulnerability", In 2006 IEEE Symposium on Security and Privacy, May 2006.
[15] A. Nguyen-tuong , S. Guarnieri, D. Greene, J.Shirley, and D. Evans, "Automatically hardening web applications using Precise Tainting", In Twentieth IFIP Intl, Information security conference(SEC 2005),
[16] David Geer, "Malicious Bots Threaten Network Security", IEEE, Oct 8, 2008.
[17] Christina Yip Chung,"DEMIDS: A Misuse Detection System for Database Systems", Integrity and internal control information systems, Pages: 159 - 178, ACM, 2008

Mr.E.Rajesh pursuing M.TECH in Computer Science and Engineering in Sri Manakula Vinayagar Engineering College, (Affiliated to Pondicherry University, Puducherry) and received the B.TECH Degree in 2006 in Sri Manakula Vinayagar Engineering College, Pondicherry University, Puducherry,. India. His current research involves in Network Security, Mobile Communication and Managing Distributed Network.

Mr.R.Raju pursuing PhD and received the M.Tech Degree in Computer Science and Engineering from Pondicherry University Puducherry. His current research involves in Service Oriented Architecture and Software Engineering. Presently working with Sri Manakula Vinayagar Engineering College (Affiliated to Pondicherry University, Puducherry) as Assistant Professor in the Department of Information Technology.

Mr.R.Ezumalai received M.Tech Degree in Computer Science and Engineering in the Department of Computer Science from Pondicherry University, Puducherry. He has obtained his B.E.(CSE.,) in 2005, from Dr Pauls Engineering College, Anna University. His current research involves in Information Security,Network Distributed System and theory of Computation. Presently working with Tata Consultancy Services (TCS) as Software Engineer inChennai.