ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Prevention Mechanism for Denial of Service in Web Application Services

Dr. D. Loganathan[1], K. Ramesh[2]

HoD, Computer Science and Engineering, Pondicherry Engineering College, Puducherry, India[1]

Student, Computer Science and Engineering, Pondicherry Engineering College, Puducherry, India[2]

Abstract: Rapid growth of Internet, and Communication technologies, Web services has introduced number of new standards and technologies. Web services are not more resistant to attacks than other open network systems. Web Services are exposed to attacks well known from common Internet Protocols and additionally to new kinds of attacks pointing the web services. Attacker can make severe impact to major business. Attacks can be performed with small effort from the attacker's side. The attacks aim to consume the network resources, thrashing web service applications, and reduce performance. These attacks cannot be detected on the web applications. The current web security standard induces crucial vulnerabilities availability in certain web services implementations. Web Services Security can provide message integrity, confidentiality, user authentication and authorization. The existing prevention mechanism has some unique advantages and disadvantages feature over the other mechanism. This is motivated to propose the Malicious Request Filtering mechanism. It would prevent internet users from taking part in Denial of Service attacks unknowingly and protecting web services from Denial of Services attacks. The new technique will provide strengthen to application level Web Services. The proposed solution is helped to test on Web Services applications and effectively filter out the malicious request from the application users.

Keywords: Web Application, Denial of Service, Attacks, Web Security issues, UDDI, SOAP.

I. INTRODUCTION

Web services are usually including few group of programming, data and human resources. Services are made available from a business's Web server for Web users and other Web connected programs. Sometimes Webs services are called application services. Web services are application components. Components can communicate using open protocols. Services are self-contained, self-describing and discovered using UDDI (Universal Description, Discovery and Integration). HTTP and XML is the basic for Web services.

A. *Web Service Interoperability*

When all major platforms could access the Web by using Web browsers, but different platforms could not interact with other. For these platforms are working together, Web applications were developed. Web applications are simply applications. Applications can work on the web. Applications are developed around the web browser standards. It can be used by any browser on any platform. Web services can use XML to code and to decode data. SOAP (Simple Object Access Protocol) is used to transport data using the open communication protocol.

B. *Web Service Usage*

- *Reuse the Existing Function:* Web service contains a unit of managed code that can be remotely called using HTTP protocol. Services can be activated using HTTP requests then web services allow you to access the functionality of your existing code over the networks. Once it is shown on the network, then other application can use the functionality of the program.
- *Interacting Different Applications:* Web Services can allow different applications to interact with other different application. It provides services among themselves. Other applications can utilize the services of the web services. For example VB or .NET application can talk to java web services and vice versa. So, Web service is used to make the application platform and technology independent.
- *Standardized Protocol:* Web Services can use standard protocol for the communication. All the four layers such as Transport, XML Messaging, Service Description and Service Discovery layers use the well-defined protocol in the web Services. This standardization of protocol gives the business with many advantages such as cost and quality.

C. *Web Service Security*

Web Services Security (WS-Security, WSS) is an addition to SOAP to provide security to Web services. It is part of the web service specifications. It was published by OASIS organization (Organization for the Advancement of Structured Information Standards). Web Services Security (WS Security) is a specification. It defines how security measures are implemented in web services to protect from external attacks. It is a group of protocols that ensure security for SOAP messages by implementing the fundamental of confidentiality, integrity and authentication. Web services are independent of hardware and software implementations. WS-Security protocols need to be easier to accommodate new security mechanisms and provide an alternative mechanism if an

DOI 10.17148/IJARCCE.2015.44109

approach is not suitable. Because SOAP based messages navigate multiple intermediates, security protocols need to be able to identify fake nodes and prevent data interpretation at any nodes. Web service security combines the best approaches to tackle different security problems by allowing the developer to customize a particular security solution for a part of the problem.

D. Web Services Security Concerns

Web Services operate on the same structure used by normal web applications. The group of request forwarded by an application and displayed in a web browser. Web Services is a SOAP request over HTTP. SOAP data is received by the server, but it is not sent to the client, then it can understand that the threat is primarily aimed at the server side. The following are methods of attack, and how Web Services can be used to fulfil these attacks.

E. Buffer Overflows

Common Impact: DoS (Denial of Service), data corruption, malicious code execution.

An attacker can send malicious XML data. It can cause the XML to call upon itself iteratively therefore constantly increasing in size. It can make a memory overflow and trigger an error messages which disclose information about the web application. A Denial of Service attack can be caused by forcing a server to pass an abnormally long XML file, which use up much more resources then actually XML file. It can crash the application. Attacker can be sending a block of data to web application. Data is stored in a buffer of insufficient size. This block of data can overwrite genuine data and cause a function return which gives control to the malicious code in the hacker's data block.

F. Xml Injections

Common Impact: Command execution, data theft and deletion, schema Poisoning.

SQL Injection is a high-risk offensive. It may be performed using SOAP messages. If the server does not validate data correctly then SOAP message can easily be used to create XML data which inserts a parameter into an SQL query. It has been execute in the server of web service. SQL Injection is only one kind of the threats to server. For example: Schema Poisoning. A schema file can understand by XML parser. It contains important pre-processor instructions. An attacker may damage the XML schema file or replace with a modified file. The modified schema file can allow the parser to process malicious SOAP messages on the server or database.

G. Session Hijacking

Common Impacts: Obtaining of user privileges within application.

Session hijacking means gaining illegal access control of a legitimate user session. Hacking occurs when an attacker steals a valid session ID (valid session cookie), and uses session cookie to gain that particular users privileges in the application. Attacker can intercepting or sniffing the SOAP messages, then attacker can hijack a legal user's session in the same ways as with normal attacks, once a hacker is authenticated as a valid user he may perform many dangerous activities.

H. Denial Of Service (Dos) Attacks And Risk Associated

Denial of Service attacks happen against web sites. Attacker attempts to make the web server unavailable to serve the web pages to legitimate users. Denial of Service (DoS) attacks mean overloading a target's resources, then the system will crash at certain point of stage. Denial of Service attacks against a web application. The application is overloaded by the attack and the application fails to serve the web pages properly. Denial of Service attacks happen on the following services.

- Network bandwidth
- Server memory
- Buffer memory

II. RELATED WORK

Techniques for preventing against Denial of Service attack can be widely divided into two categories.
A.General Techniques
B. Filtering Techniques

A) General Techniques

General techniques are having some common preventive measures for system protection, replication of resources. Each Internet users should follow the common preventive measure to avoid threats, vulnerabilities and attacks.

1) Disabling Unused Services Technique

Felix Lau et al. [1] proposed disabling mechanism. There are applications and open ports in Internet systems. The open port applications are chance to use the vulnerabilities by attackers. So if network services are not needed and unused, then the network services could be disabled to prevent attacks such as UDP echo, character generation services.

2) Install Up Do Date Security Patches Technique

X. Geng et al. [2] proposed patches solution techniques. Today, many of the attacks use vulnerabilities in target computer. So kindly we have to removing known security holes by installing all relevant latest security patches to prevent re-exploitation of vulnerabilities in the target web system.

3) Firewall Technique

R. Oppliger et al. [3, 4] proposed firewall filtering technique. Firewall means controlling the network access to one or more computers. The Internet is large network in which includes your computer. A firewall protects computer by acting as a gate wall. Data can be passed through the gate. Firewall could protect your computer or network from unauthorized users and data from attacks. Firewalls can effectively prevent users from launching simple flooding attacks from machines behind the firewall. Firewalls used simple rules such as to allow or deny

DOI 10.17148/IJARCCE.2015.44109

protocols and IP addresses. But some complex attack for example: if there is an attack on port 80 (web service). Firewalls cannot prevent that kind of attacks because firewall cannot distinguish good and bad traffic from DoS attack traffic.

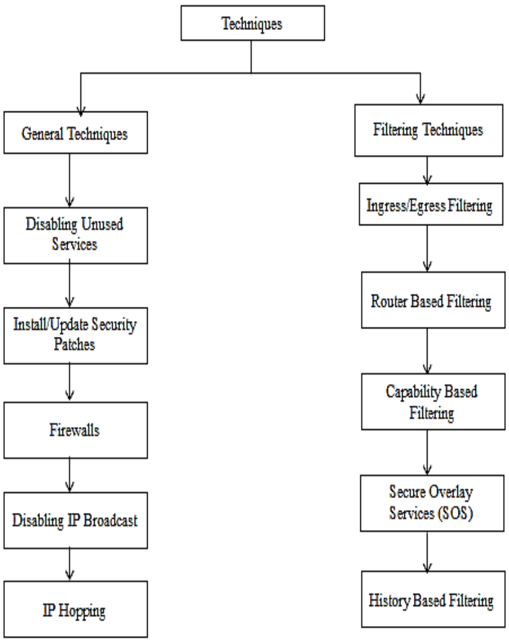

Fig. 1. Filtering techniques

4) Disabling Ip Broadcast Technique

Felix Lau et al. [1] proposed Disabling IP broadcast filtering techniques. The routers are respectful. It will work what the source requested. A Smurf attack is a version of a Denial of Service (DOS) attack in which an attacker sends an ICMP echo request to your network's broadcast address using a spoofed source address. That request causes all the hosts to respond to the broadcast request and it will slow down the network. In order to prevent the cause then we have to disable the network broadcast in your network.

5) Ip Hopping Technique

U.K.Tupakula et al. [5] DDoS (Distributed Denial-of-Service) attacks can be protected by changing location or IP address of the active server proactively within a pool of pre-specified set of IP address ranges [18]. The victim computer’s IP address is invalidated by changing it with a new one. Once the IP addresses change are completed and informed to all internet routers and edge routers. These routers will drop the attacking packets. Attacker can launch the attack at the new changed IP address. Attackers can make the technique useless by adding a domain name service tracing function to DDoS attack tools. This changing address may include the IP addresses of different ports on a server. The pattern of changes of the IP address is known to both the client and the server but secret to others. Attacker cannot eavesdrop with data set without knowing the IP changing pattern. To further improve the security of this technique, the server system is configured to subsequent requests at the changed IP address pattern. If the subsequent requests do not arrive within a threshold time limit, then the server system is to terminate to access the data set by the requesting client.

B) Filtering Techniques

This technique can be included ingress filtering, egress filtering, router based packet filtering, history based filtering etc. These techniques have been used by security expert team.

1) Ingress/Egress Filtering

Ingress filtering techniques proposed by Ferguson et al. [6] is a restrictive mechanism to drop traffic with IP addresses that does not match a domain prefix connected to the ingress router. Egress filtering technique is an outbound filter. It ensures that only assigned or allocated IP address space present in the network. The important requirement for ingress and egress filtering is knowledge of the expected IP addresses at a particular port. Reverse path filtering can help to build that knowledge. Generally, a router always knows which networks are reachable through its interfaces. Router looking up source addresses of the incoming traffic and it is possible to check whether the return path to that incoming address would fall out in the same interface as the packet arrived. If the path present, then these packets are allowed else packet are dropped. This technique cannot work effectively in real networks where asymmetric Internet routes are not unusual.

2) Capability Based Filtering

Baker, F et al. [7, 8] proposed Capability Based mechanism. In this model provides destination machine has control traffic capabilities with itself. The source machine first sends request packets to destination. Router can be added capability mark to request packet while passing through the router. The destination system may or may not be provided permission to the source machine to send the request. If permission is granted then destination returns the capabilities mark to source. The data packets carrying the capability marks are sending to these systems but this technique had high computational complexity and space requirement.

III. PROPOSED SYSTEM

The main objective of proposed work is to design a new technique for web service to avoid the vulnerabilities. Behavior based is a new technique. This technique can capture behavior of web service user and compare the behavior of normal user. The behavior is various from the normal user of web service components then the particular user can be blocked based on some parameter such as user attitudes.

- *Service Control*

Service control provides web service clients with easy access to a web service. The service control is provided as one of the system controls. When using a service control, you can invoke a web service operation by simply calling a method of the service control. The service control manages the SOAP message exchange with the web

DOI 10.17148/IJARCCE.2015.44109

service and returns the results of the web service operation.

- *Service Call Analysis*

The user can request a web service from their form control. The Service call analysis can call particular service request from registered service in web service directories. The directories contain lots of service name through some binding service mechanism. The binding mechanism means service level agreement between server and client. Finally the response can hand over to requested user control.

- *Web Security Standard*

W3 consortiums, OWAP (Open Web Application Project) consortiums, WASC (Web Application Security Consortiums) are organization provide the security standard for web application project. The developers can develop the project based on the recommended standard. Unfortunately developers fail to follow the standard, and then it can lead to make some security problems.

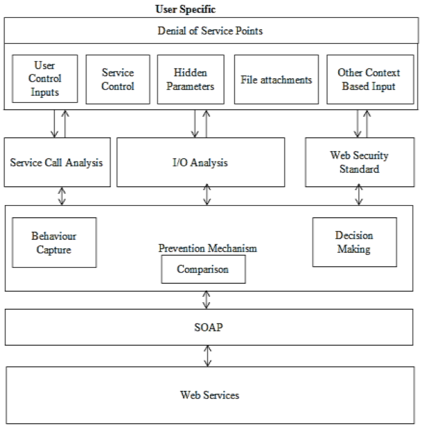

Fig. 2. Overall architecture of proposed system

IV. MODULE DESCRIPTION

Behavior based filter mechanism contains three modules:
MODULE 1: Behavior Capture Module
MODULE 2: Behavior Analysis Module
MODULE 3: Decision Making Module

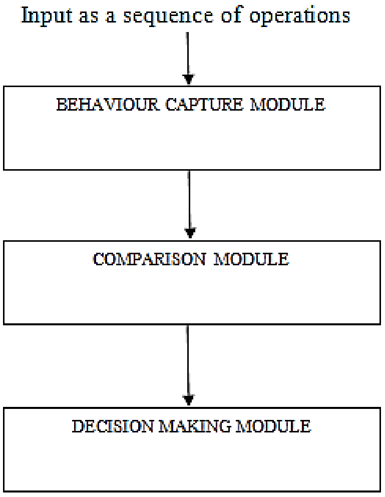

Fig. 3. Behavior based filter mechanism

MODULE 1: Behavior Capture
This module can capture the user's behaviour from the available web service application over the internet communication environment. The behaviour can capture/calculated by the user access the web service applications. Capture data can be given to next module of prevention mechanism.

MODULE 2: Behavior Analysis
Comparison module can get the input from the behaviour capture module. The input can be compared with current user presently using the web service application over the network system, and internet technologies. The current user is not obeying with compared behaviour data then the particular user activities are temporary or permanently blocked. That particular user activity is monitored and notification can be sent to administrator. The administrator can view the vulnerable generated user in their profile. The module can produced data to next module of prevention mechanism such as decision making module.

MODULE 3: Decision Making
This module can get the input data form the comparison module. Decision making module can make exact decision to block or allow the user to continuously using web service application present the network environment. Decision can be made based on the result came from the comparison module. User can be temporary blocked or permanent blocked depend on the decision of decision making module.

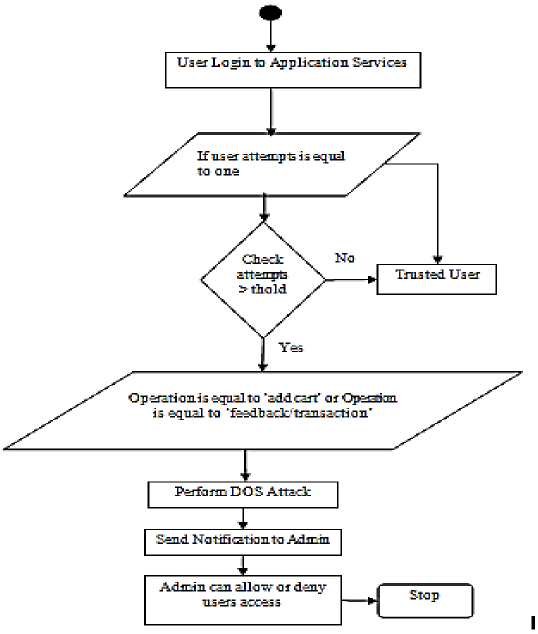
User Login to Application Services
If user attempts is equal to one
Check attempts > thold
No
Trusted User
Yes
Operation is equal to 'add cart' or Operation is equal to 'feedback/transaction'
Perform DOS Attack
Send Notification to Admin
Admin can allow or deny users access
Stop

Fig. 4. Flow chart of behavior analysis algorithm

Algorithm For Behavior Analysis
Let Z be the total number of operations
U be the user attempt
V is the threshold for operation. Let us take the Maximum is 10. (1 ≤ V ≥ 10)
W be the end operation

```
Z = UVW
|V| should be equal to at least 1.
If U==1 && W==0
No change in the system and the user is said to trusted user.
Else
While |V|>1
If Operation (U) == 'add cart' || Operation (U) == 'feedback'
Check V>10
Do U (DOS_ATTACK)
Set U='MALICIOUS USER'
Send notification to admin and user.
Operation (admin) == 'allow user' || 'deny user'
End while
End if
```

V. CONCLUSION

Every upcoming technology, Web Application Services plays a major role in the information exchange. Web services are faced by several security issues. Especially application layer attack has become a major threat to the internet world. The proposed work can provide sufficient security solutions to prevention of clients from unintentionally take part in the vulnerabilities. The proposed work can be implementing in Window Operation System. It has sent the alert notification to user and administrator account. The future works will examine with cloud Web Application Services environment and to prevent the threats present in the cloud service technologies as well as examine with other vulnerabilities present in the today web services world.

REFERENCES

[1] Felix Lau, Rubin H. Stuart, Smith H. Michael, and et al.,"Distributed Denial of Service Attacks," in Proceedings of 2000 IEEE International Conference on Systems, Man, and Cybernetics, Nashville, TN, Vol.3, pp.2275-2280, 2000.

[2] X. Geng, A.B. Whinston, Defeating Distributed Denial of Service attacks, IEEE IT Professional 2 (4) (2000) 36–42.

[3] R.Oppliger,"Internet Security: firewall and beyond," Communications of the ACM, Volume 40, Issue 5, pp. 92-102, 1997.

[4] McAfee, "Personal Firewall". Available at: http://www.mcafee.com/ myapps/ firewall/ov_firewall.asp.

[5] U.K.Tupakula, V.Varadharajan "A Practical Method to Counteract Denial of Service Attacks", Proceedings of the Twenty-Sixth Australasian Conference on Computer Science, ACSC2003, Springer Verlag, Australia. (Feb 2003).

[6] P. Ferguson, and D. Senie, "Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing," RFC 2267, the Internet Engineering Task Force (IETF), 1998.

[7] T. Anderson, T. Roscoe, D. Wetherall, "Preventing Internet Denial-of-Service with Capabilities," In ACM SIGCOMM Computer Communication Review, Volume 34, issue 1, January 2004, pp. 39-44

[8] A.Mohamed Ibrahim, L. George, K. Govind, S. Selvakumar, Threshold based kernel level HTTP filter (TBHF) for DDoS mitigation, 2012.

[9] http://www.applicure.com/solutions/prevent-denial-of-service-attacks [Denial-of-Service].

[10] www.infosecisland.com/blogview/10442-DDoS-Attacks-Possible-via-URL-Shortener.html.