International Journal of Information and Communication Technology Research

http://www.esjournals.org

Preventing Buffer Overflow DOS Attack in Restful Services

Hussam A. Elkurd , Tawfeeq S.Barhoom
Faculty of Information Technology
Islamic University Gaza
Gaza, Palestine

ABSTRACT

Buffer overflow form one most kind of attacks in restful services, as result from misuse or intentional attack. Therefore client request a resource many times that consume processing time for each request and may cause DOS. In our paper we propose a mechanism to prevent DOS buffer overflow, and limit the authorized token lifetime according to the next expected request time. In addition we apply the study into a real experiment by implementing a web restful API and The results of the test show it is capable of preventing the attack.

Keywords—DOS; Restful; Buffer overflow; JSON; Tokens

I. INTRODUCTION

Representational State Transfer (REST) architectural designed for public web services over the internet. Thus the resources are accessed using Uniform Resource Identifiers (URI's). While the actions are transmitted by Http methods (POST, GET, PUT, and DELETE). The client receive Http status codes as feedback result. Therefore JavaScript Object Notation (JSON) is used for exchanging data, JSON is an open standard format that uses human-readable text to transmit data Objects consisting of attribute–value pairs. In addition stateless is major restful design principle, thus each request are computed from scratch, therefore many REST users are not applying the full standard. Now a days the services are shift to adopt restful architecture instead of Soap-Based, because restful is lightweight and good choice for heterogeneous devices. Despite the benefits restful security are not taken by default and there is no standard can applied. Many types of attacks form security breaches in restful service include data confidentiality and integrity as the attacker try to alter the messages between the client and the provider, or have unauthorized access to resources. In the interim there are many considerable attacks in restful API, as it usually public and have wide range access. The attacks are not totally new. However they are the known attack in the internet such as Man-in-the-Middle (MITM), Replay attack, Spoofing, Message altering, Cross-Site Scripting XSS [8].In the other hand the availability is great concern for restful services, however the services are usually public and wide range of clients are access. Denial of the service (DOS) is one of the most security threats in restful services, either it is created intentionally by attacker or misuse. Exploit the principle of stateless, each request consume time and resource to create a response from scratch. The attack created by repeating specific request many times. Recently the available researches are limited in restful security, their main concern is to adapt current service standards such as WS-Security into REST services with take the differences into account and therefore they are not targeting specific REST security breach. Furthermore the common effective security that all researches agree is security tokens. Tokens are generated from the provider with a time frame to client in order to access resource legally, it is often used in restful services as no username and password resented each time request a resource. The token has a limited time albeit it ended the token are not valid anymore and the client need to request new token from provider. the mechanism to extend the time is the computing activity as each time the resource are requested there is constant increment to the computed time. As the attacker may use this increment to increase attack time by using valid stolen token. In our paper we propose a mechanism to handle client request tokens refresh time and prevent repeating requests for each client, the back draw for the mechanism is that is not identical agree restful principle as all available security solution. But it adopt it, however it save some client request information that needed for security process. The rest of this paper is organized as follows: the next section overview the related work for restful security. Section 3 introduce our propose solution. In section 4, our experiment to apply the solution. Section 5 the result. Section 6 concludes the paper.

II. RELATED WORKS

Our methodology of find related works is looking for topics related to DOS attack in general and Restful security concerns. Irfan siddavatam [1] propose comprehensive set of test mechanism to detect attack on web services. The study prepared for Soap-based web services but some attacks are common with REST and both of them have the same solution idea. The study propose as solution for buffer overflow as define a guard time, thus estimate the difference between the current Soap request and the previous soap request if it is less than the guard time its detected as attack, else the new request is set as previous time. In our solution we add dynamic different guard time for each action, thus computing time is different from one resource to another. In recent study for Zhang Chao-yang [2] provide measurement to prevent DOS attack, part of it depend on hardware like creating trusting computing platform depend on filtering chip that detect attacks earlier.

However hardware solutions have effective impact of security, therefore we can apply the study of Ramin [3] that define frequency characteristics for DoS into high speed intrusion detection systems that support restful web services architecture as Mohsen [4] propose to detect and prevent any DOS attacks and if the hardware does not have a direct restful API it can manipulated as mentioned in recent study for Lei [5]. Furthermore add features to used protocol also is effective way to prevent DoS as the study of D. Cotroneo [6] propose. In our paper the propose solution is algorithmic, However we can provide union solution with hardware for restful in future work. In addition Zhang Chao-yang [2] add certification system defense that define the identity for each client, in restful security we can define the identity of authorized clients as condition to generate new tokens, this limit the probability of attacks outside the defined clients. In recent word of Gabriel [7] the study provide an encryption for message exchange between the provider and client throw restful define new encrypted headers in the http request and response in order to meet restful principles. While the idea is novel but it has a drawback thus always we have to overload request and response with extra headers. Furthermore the study Femke[8] compare the current known security mechanisms used in restfull and find that there is no optimal solution or preferred. Thus increasing the efficiency violate REST principles, in contrast take avoid violating principles solution consume more resources and cause overhead. In our paper we try to adapt the REST design as possible, by save the needed information for security.

III. PROPOSED SOLUTION

Buffer overflow is created by repeating a valid request multi-time that cause consuming resource and processing time and may end of DOS, thus we create a mechanism to manage the time for authorized execution on resources and prevent buffer overflow, the mechanism introduced as the following points:

A. Define the clients Identity

This depends on the nature of service as its public and the clients are not known, or privet. If the clients are known the recommendation is to define the clients identity throw static IP's and discard any request outside the trusted clients group. In the other hand if the service is public, the recommendation is to block any malicious client IP.

B. Computing Guard Time

The main idea is to define a computing time for each access resource, therefore the resources in restful are accessed throw URI's and it may include different forms of parameters to access the same resource, thus we will take the parameters into account. However this enable define set of conditions to define the required computing time for resource if it contains parameter. Moreover the parameters should restricted to types and have a guard values (min and max) in order not allowing the attacker to exploit it to change the values always to get more computing time. In addition the algorithm discard any request from the same client to the same resource while the defined time is still available to avoid DOS attack. The request time is checked after that it saved as previous time to compare it with the new request. If the difference between the new request time and the previous time are less than the expected computing time, the request is negated. Figure 1 show the algorithm pseudo code.

Input: Current and Previous Time of REST resource request.
Variable:
Current_Time:- Current REST resource request time.
Previous_Time:- Last Time at which REST resource request made
Computing_Time:- Current Process computing time
Next_Max_CT:- Next expected request maximum computing time
Allowance_Time:- fixed time wasted throw communication and request

Output: set new token, or discard computing
Begin

Step 1: /*Check token authorization & availability*/
If the token is not authorized or available
Then
Return message to confirm token.
End if
Step 2: /*Check validity for parameters*/
If the parameters not meet the guard values
Then
Return message to validate resource request.
End if
Step 3: /* Get resource computing time */
Define the computing time according to request resource
If Difference between Current and Previous Time Less
Than
Discard request
End If
Step 4:
Store current time as previous time
Set the token value as Next_Max_CT + Allowance_Time
End

Figure 1 Algorithm to prevent DOS buffer overflow attack and extend token lifetime

C. Time to extend token lifetime

Tokens are used to have authorized access on resources, its generated from the provider with limit interval, if the limit is ended the token is being not valid anymore and new token should be generated, as many provider provide a

constant increment for tokens when request new resource in order to make increment the lifetime in active computing. To limit the token time, the token will have new defined time according to the requested resource. The token new time can formulated as the following:

$$T = NEt + At$$

Therefore the T is the new token time and the NEt is the next expected maximum computing time and the At is allowance interval including communication time and the time wasted to move from one resource to another.

IV. EXPERIMENTAL SETUP

To apply our preventing mechanism, we create a REST API with PHP script and make computing time for 5 second using sleep function. The code has the same idea of figure1.thus define the computing time for 6 second to complete. We use Firefox firebug add-on to create multiple request and get the response.

V. RESULTS

The test get the expected result, therefore while we are running request on the requested resource over REST API the new requests will be prevented until the defined computing time is completed. In the running request the token refresh time is defined as the next maximum executing time and the allowance time.

VI. CONCLUSION

In this paper we show an overview on Restful and the security threats, however DOS buffer overflow is widely common attack that cause DOS for restful services. In our study we create a mechanism to prevent overload replies by manage the time for authorized execution. We experiment the mechanism throw implement a restful API and apply the algorithm to prevent request the same resource multiple time as we get the expected result. In addition the proposed mechanism are violating REST principles of full stateless requests. In future work we are looking to find an effective way to prevent the attack without violating REST principles.

REFERENCES

[1] Siddavatam, I.; Gadge, J., "Comprehensive test mechanism to detect attack on Web Services," Networks, 2008. ICON 2008. 16th IEEE International Conference on , vol., no., pp.1,6, 12-14 Dec. 2008.

[2] Zhang Chao-yang, "DOS Attack Analysis and Study of New Measures to Prevent," Intelligence Science and Information Engineering (ISIE), 2011 International Conference on , vol., no., pp.426,429, 20-21 Aug. 2011.

[3] Fouladi, R.F.; Seifpoor, T.; Anarim, E., "Frequency characteristics of DoS and DDoS attacks," Signal Processing and Communications Applications Conference (SIU), 2013 21st , vol., no., pp.1,4, 24-26 April 2013.

[4] Rouached, M.; Sallay, H., "RESTful Web Services for High Speed Intrusion Detection Systems," Web Services (ICWS), 2013 IEEE 20th International Conference on , vol., no., pp.621,622, June 28 2013-July 3 2013.

[5] Lei Gao; Chunhong Zhang; Li Sun, "RESTful Web of Things API in Sharing Sensor Data," Internet Technology and Applications (iTAP), 2011 International Conference on , vol., no., pp.1,4, 16-18 Aug. 2011.

[6] Cotroneo, D.; Peluso, L.; Romano, S.P.; Ventre, G., "An active security protocol against DoS attacks," Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on , vol., no., pp.496,501, 2002.

[7] Serme, G.; de Oliveira, A.S.; Massiera, J.; Roudier, Y., "Enabling Message Security for RESTful Services," Web Services (ICWS), 2012 IEEE 19th International Conference on , vol., no., pp.114,121, 24-29 June 2012.

[8] De Backere, F.; Hanssens, B.; Heynssens, R.; Houthooft, R.; Zuliani, A.; Verstichel, S.; Dhoedt, B.; De Turck, F., "Design of a security mechanism for RESTful Web Service communication through mobile clients," Network Operations and Management Symposium (NOMS), 2014 IEEE , vol., no., pp.1,6, 5-9 May 2014.