ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

Session Security Improvement in Net Applications

S. Seema, P. Daniel Sundarraj

Research Scholar, PG & Research Department of Computer Science and Applications, K.M.G College of Arts and Science, Gudiyattam, Vellore, Tamilnadu, India

Head, PG & Research, Department of Computer Science and Applications, K.M.G College of Arts and Science, Gudiyattam, Vellore Tamilnadu, India

ABSTRACT: Net applications available today make use of some way of **Session Management** to be able to couple state to a particular user. This state varies from the users preferences to user authentication and private information. Sadly, it is possible for an attacker to exploit session management in order to take off another user at a net application. In this paper we describe attacks that enable an attacker to impersonate a victim, and the ways in which they can be prevented. Different attacks abusing session management are known: session hijacking, session fixation, and cross site request forgery, wherein the attacker uses a victim's browser to issue requests as if they came from the victim. For all three attacks, different attack vectors exist, which allow an attacker to create complex attack scenarios which are difficult to prevent. We propose a client-side solution to session fixation and session hijacking attacks, which is based on the principle that SIDs that are managed via JavaScript should not be allowed to interfere with SIDs that are managed over HTTP. We implement our solution as an add-on for the Firefox net browser and find that it protects users against the abovementioned attacks, while having little to no impact on the user experience.

KEYWORDS: Net Session; SID; Session Attack; Client side measure; HTTP Proxy

I. INTRODUCTION

Most net applications handle user authentication via the concept of net sessions. These allow users to use a net application without having to enter their login ID for every action taken. Unfortunately, net sessions have many security weaknesses. Many high-profile net applications are vulnerable to attacks on session management: YouTube and Twitter are two examples of net applications that used to contain such vulnerabilities in the past.

A problem is that users of a net application have to trust the developer of the application to take the necessary security precautions. To enable users to protect themselves against session attacks, regardless of the net applications being secure, a client-side tool offering protection against these attacks is needed.

II. RELATED WORK

[1] Developed a fuzzy based trust management framework for web service. Initially, they developed a data model based on consumer views on QoS attributes that evaluates the reputation of services. [2] For service selection in general service oriented environments. It follows Reputation based and Trusted Third Party approach. It overcomes the limitations of Certified Reputation Model. [3] Designed the social rules on describing the trust relationship between the provider and consumer in the open environments. [4] Treated each user as a potential information provider. This model proposed to each user's trustworthiness by propagating over a network of people connected by ratings. [5] Proposed a multi layer trust computation model based on direct search in which service providers need to compute and control the trust of users. [6] Proposed a trust model based on behaviour which describes the consistency of behaviour and focused on behaviour of entities in different contexts.

SCOPE OF RESEARCH

In this paper, we examine the session's security in net applications.

Our aids are as follows:

• We offer a complete overview of three important session attacks, together with a list of possible attack vectors for each of these attacks.

• We systematically evaluate different solutions that were proposed over the years to improve session security.

• We suggest an original client-side approach to solving two important session attacks, and we apply our policy as an add-on for the Firefox web browser. We also provide an extensive estimate of our add-on.

III. PROPOSED METHODOLOGY

Different security measures may be taken by web application and web framework developers to secure their web applications. In this section, we discuss to what extent these measures provide security against session hijacking and session fixation attacks.

1. *Renewing the session identifier*

The practice of renewing the session identifier can provide excellent protection against login session fixation attacks.

2. *Using Http Only cookies*

When setting a cookie in the user's browser, a web server can use the cookie's Http only flag to indicate that the browser should only allow access to this cookie via HTTP.

3. *Checking request headers*

A HTTP request can contain many other headers. Some of these headers provide information that can be used to identify a user.

4. *Checking the IP address*

The IP address can be used by the web server to ensure it is interacting with the same user as before.

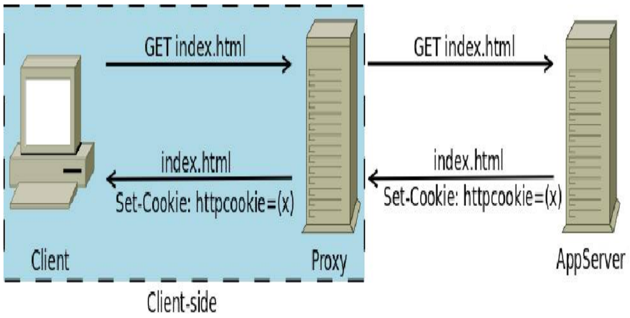
A CLIENT-SIDE SOLUTION TO SESSION FIXATION AND SESSION HIJACKING

The client-side solution is that session IDs will not at all be set over an entrusted channel, only to be requested over a trusted channel shortly. We consider HTTP to be trusted, since this channel is controlled entirely by the web server. As an entrusted channel we consider elements in the web page itself, such as JavaScript and <meta> tags, because they often contain user input.

The solution has the form of a proxy that is located at the client-side. As a basic policy, we choose to only allow cookies in outgoing HTTP requests if they were previously set via an HTTP response from the server. When a new cookie is sent to the client via HTTP, the proxy remembers this cookie. When an outgoing request is sent to the server, the proxy checks all outgoing cookies. If one of these was not set via HTTP, it is removed from the request. This prevents all cookies set via JavaScript or <meta> tags from being used over HTTP.

IV. SIMULATION RESULTS

A browser extension is already behind the regular SSL endpoint. An HTTP proxy, on the other hand, should provide its own SSL endpoint if it wants to intercept secured traffic. This means that it should handle encryption, handshakes and certificates – which are all very difficult to get right – separately from the browser. Mozilla Firefox allows extending the browser using a combination of JavaScript and XML. These can access the browser's XPCOM components, which offer access to various browser features.



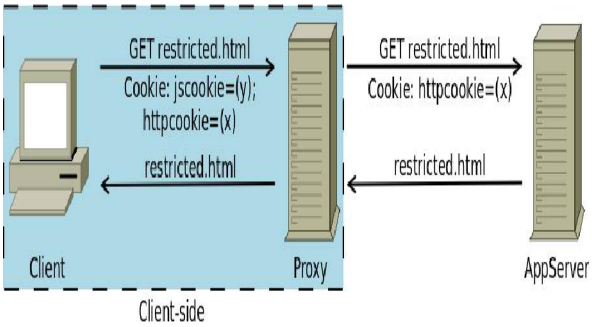


Fig.1. An incoming response containing a new SID. Fig.2. An outgoing request containing a cookie set via HTTP.

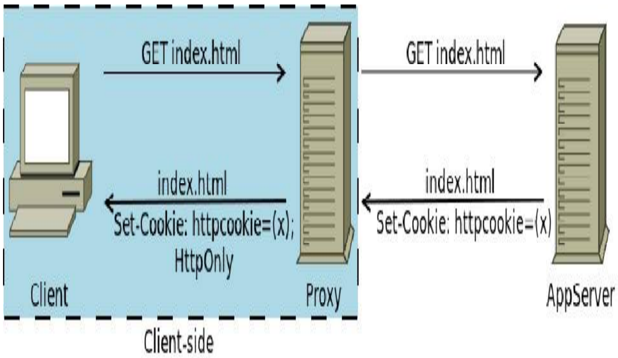


Fig.3. The extended client-side solution, providing both session fixations and session hijacking protection

V. Conclusion and Future Work

The standard of the solution is that a clear division should be made between HTTP cookies and cookies set via other channels. In this sense, our approach is very similar to that of Session Shield and Http Only cookies. However, our approach extends this behaviour to make it robust against session fixation attacks. The solution was implemented as an add-on for the Firefox web browser, which allowed it to be used by testers for a longer period. We tested the impact of enforcing a policy that prevents un trusted cookies from being sent over HTTP, and found that – however a considerable amount of cookies is blocked – enforcing such a policy has no negative effect on the user experience. Furthermore, the implementation of this policy has no noticeable impact on web browsing performance.

References

[1]. Paul Arana. Benefits and Vulnerabilities of Wi-Fi Protected Access 2(WPA2), 2006. Available from: http://cs.gmu.edu/~yhwang1/INFS612/ Sample_Projects/Fall_06_GPN_6_Final_Report.pdf.

[2] Adam Barth, Adrienne Porter Felt, and Prateek Saxena. Protecting browsersfrom extension vulnerabilities. *Proceedings of the 17th Network and DistributedSystem Security Symposium (NDSS), San Diego, CA*, 2010. Availablefrom: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1. 1.154.5579&amp;rep=rep1&amp;type=pdf.

[3] Adam Barth, Collin Jackson, and John C. Mitchell. Robust defenses for crosssiterequest forgery. *Proceedings of the 15th ACM conference on Computerand communications security - CCS '08*, page 75, 2008. Available from: http://portal.acm.org/citation.cfm?doid=1455770.1455782.

[4] Constantin Bejenaru. Disable session IDs passed via URL. Available from:http://www.frozenminds.com/disable-sessionid.html.

Prithvi Bisht and V.N. Venkatakrishnan. XSS-GUARD: precise dynamicprevention of cross-site scripting attacks. *Detection of Intrusions and Malware,and Vulnerability Assessment,* pages 23–43, 2008. Available from: http://www.springerlink.com/index/8561322615176852.pdf.

BIOGRAPHY

Seema is a Research student in the Department of Computer Science and Applications, K.M.G College of Arts and Science, Gudiyattam, Tamilnadu, India. She received Master of Computer Application (MCA) degree in 2012 from Priyadharshini Engineering College, Vaniyambadi, India,. Her research interests are Network Security.