The Impact of AJAX Vulnerability in Web 2.0 Applications

Abdullah Abdulaziz Al-Tameem
Al-Imam Muhammad Ibn Saud Islamic University, Saudi Arabia.

Abstract: *The web has profound applications in the world and influenced the society than any other technology. The present decade is marked by the increased application of web servers and the use of web 2.0 for many utilities. As the access to the Web 2.0 by the global users is on ever increase, the hijackers eye on the Web 2.0 for attacks. The principal technology behind the Web 2.0 is the AJAX which now becomes the target of many security attacks. The current study has visualized an enhanced architecture to reflect the AJAX vulnerability and the architecture is applied in randomized trails and the results call for increased applications*

1. Introduction

The most striking technology application in this century is the impact of web on the human life. The current period has witnessed the increased use of web to a greater extent and the Web 2.0 has made the cyberspace as the global information space. Web 2.0 is a collection of technologies and services that allow increased user-creator interaction, content syndication, advancements in web-based user interfaces, which ultimately lead to the creation of an entirely new application platform.

According to Pete Lindstrom, Director of Security Strategies with the Hurwitz Group, Web applications are the most vulnerable elements of an organization's IT infrastructure today. An increasing number of organizations (both for-profit and not-for-profit) depend on Internet-based applications that leverage the power of AJAX. As this group of technologies becomes more complex to allow the depth and functionality discussed, and, if organizations do not secure their web applications, then security risks will only increase [1]. The most striking features of web 2.0 are its ability in harnessing collective intelligence and bringing rich users participation. The web 2.0 has rich applications with features such as user interaction, knowledge transfer and sharing and end user as well as source collaboration. The AJAX technology is used in many synchronous environments ranging from learning to many other purposes and their potential as well as implications are addressed in many studies [18-20].

The basic strength of the present web services environment is the Rich Internet Application as because of the deployment of JAVA script and DHTML for gaining interactivity in web pages. The fuse of many interrelated technologies has made the AJAX, (the Asynchronous JavaScript) to provide the avenues for increasing user participation. As the AJAX is the basic component of Web 2.0, it becomes the vulnerable for attacks also. It has been observed in many applications that most of the Ajax toolkits have been found vulnerable that leads to JavaScript hijacking;

JavaScript Hijacking allows an unauthorized attacker to read sensitive data from a vulnerable application using a technique similar to the one commonly used to create mashups [5].

The resources and tools on Web Services Threats and Vulnerabilities focused on the news, white papers and other media in the recent past addressing many issues. The studies and prototypes are initiated with large scale experiments and implemented in practice; still the threats are increasing in many dimensions.

2. Background

In the last few years, the use of AJAX as the major scripting language is accelerating due to several benefits it offers. These includes inclusion of dynamic forms for in built error checking, area measurement in pages, dynamic changes in background, URL access history, availability of data such as framed and non-framed while users request, and many others.

AJAX is evolved by combing many technologies where each contributing component offers the significant milestone thus making web applications to a higher level. The web browsers such Explorer or Firebox utilize the techniques of AJAX to improve the performance and increase applications. Thus, the JavaScript deployment enables the browsers and consequently the end-users with the bundle of technological advantages.

The AJAX technologies particularly the JavaScript is vulnerable and ignoring such vulnerabilities raises security concerns among developers, website owners and the end-users. The proposed solution suggests auditing AJAX and JavaScript based applications with a web vulnerability scanner that not only parses the HTML code of a webpage to identify embedded JavaScript, but also executes the code. Automating the process is also an important key when considering the increasing complexity of such web applications [2].

3. The Web 2.0 Architecture layout

The peddle of the Web 2.0 concept is its ability to make the web pages more dynamic and its technological strength in bringing the 'database' concept. The dynamism of web 2.0 is not just because of user interaction but on the dynamic links and the degree of structure and organization. What is dynamic about the web 2.0 is not just the pages, but the links. A link to a weblog is expected to point to a perennially changing page, with "permalinks" for any individual entry, and notification for each change [4].

Received August 20, 2008

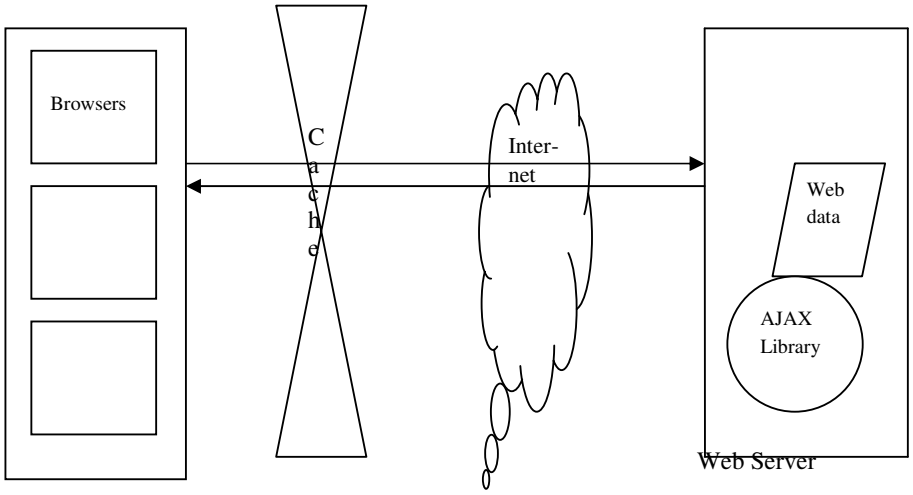

Figure 1. The AJAX Architecture

The web 2.0 relies on AJAX and the Figure 1 illustrates how AJAX works in web services. The AJAX application employs the web page-client framework in the web services. The widely adapted data stream is the JavaScript Object Notation (JSON) data stream, particularly in the AJAX environment. The browsers send the data feeds and updates the JavaScript. The web server replies to the users through the browsers by returning data. This basic procedure brings what the users need in the web server transactions. The web servers operations are improved by applying enhanced solutions, code schemas, and accessibility procedures. However the environment is susceptible as security threats surface.

Used together or separately, these technologies have increased the flexibility of web applications. However, when implemented without security considerations, application inputs can be vulnerable, and old attacks can gain new footing.

4. What makes Web 2.0 vulnerable?

In the last couple of years, there is an ever-increase of cyber microcosm attacks leading to challenge the security landscape. The popularity of the Web 2.0 applications, such as social networking sites, wikis, and blogs, facilitate collaboration and sharing between users and lead to the growing attacks. However, the increased popularity of these applications has driven hackers to target users and businesses using these emerging tools. Using mash-ups, unattended code injection, and other tactics, Web 2.0 hackers provide yet another level of complexity for customers that want to prevent data loss and malicious attacks [19]. Hackers, masquerading as the Internet directory services, went trick-or-treating for personal banking information. Many sites now sites host a large number of malicious binaries.

5. Some of the JavaScript attacks

Hackers are upping the ante with evasion techniques that use poly-morphic JavaScript (Polyscript), which is a uniquely-coded Web page served up for each visit by a user to a malicious Web site. By changing the code every visit, signature-based security scanning technologies have difficulty detecting Web pages as malicious and hackers can extend the length of time their malicious site evades detection [19]. The Web as an attack vector has been steadily increasing for the last five years and now attackers are using compromised sites as their launching platforms than their own created sites [19].

6. AJAX- Vulnerability Tests

The environment of AJAX vulnerability and the access pattern is illustrated in the Figure 2.
The Hijacking site uses the AJAX query using the script tag in the form of a query to the targeted site. The AJAX query function in the GET environment and when the session cookies work in the target site, the scripts embedded in the target site are transferred to the Hijacking site. Similar of hijackings were reported in the recent past [10]. The secured JAVA code can fix such problems.

Many hijackings are reported recently in the web and documented in many news items and extensively addressed in the literature. The solution we could fix is to develop the AJAX client. The client we have developed has in built '*Search Builder*' in the basic site which uses a simple "WYSIWYG" tool to make advanced queries; Our client is an extensive object-oriented user interface programmed in JavaScript. The client follows a *Model-View-Controller* (MVC)

Received August 20, 2008

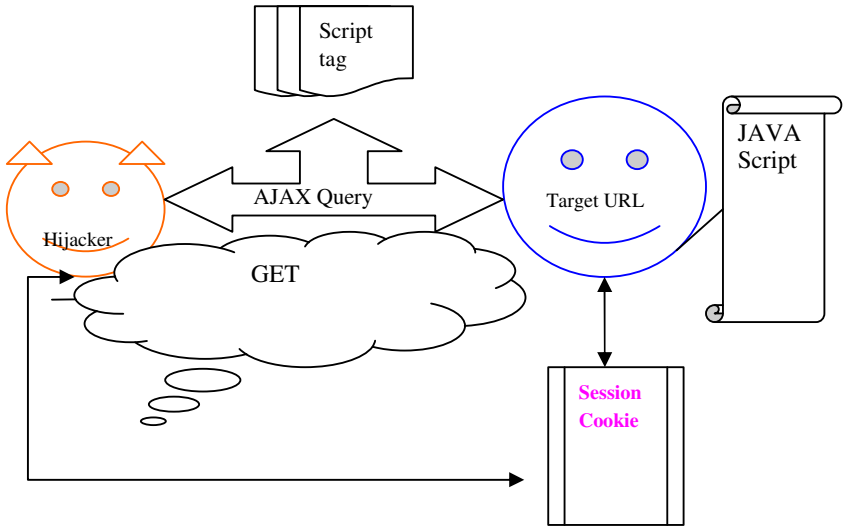

Figure 2. AJAX vulnerability

architecture, with state changes, rendered in the UI and sent to/retrieved from the backing server. Optimization techniques essential to the user experience/lower latency include aggregating multiple JavaScript and CSS files, compressing the result, and browser caching (of code, not user data).

The framework, we have developed has a package of web service codes written using JavaScript APIs. The *servlets* are exposed to the end user through our Ajax client. The *servlets* permit the pattern to use for the actions such as configuration and access control. We have borrowed a local search engine that is developed to manage the activities of indexing and content access. Our aim is to make it scalable to all possible created templates.

7. Random trials

Patches were available for the development version of trial that introduces some of the proposed features for JAVA *Servlet*. Key among these is a suspend/resume API on the request object that is a proposed standardization the code continuation mechanism for asynchronous *servlets*. The patch is against the trial version in '*svn*' head. We have used a series of commands to checkout the code, apply the patches and build the server.

We have performed random trials in a stimulated environment to document the effect of AJAX client. The '*n*' trials after several looping and iterations bring the following benefits. The client has ensured storing of multiple messages and meta-data/index stores. The query models we have generated have employed SQL that were able to counter the various forms of queries during the session cookies. Our code has proved to manage all possible all multiple server configuration and administration. Our client has proved to be scalable and secured.

8. Related Research

A careful analysis of potential attacks against Web services as carried out e.g. by Jensen et al. immediately shows that Web services are very vulnerable especially against DoS attacks [12]. The security issues which are inherent to the Ajax programming model and which especially affect cooperative application have been extensively documented by Michael Sonntag [16].

Many organizations are planning to introduce newer Internet Rich Applications where as IT managers and researchers are more concerned with technology vulnerabilities. It has been documented in many recent research studies that AJAX-JAVA based vulnerability is higher than other web 2.0 attacks and more sophisticated in AJAX applications; and that Web Application Firewalls will become an integral part of a unified approach to security [7]. Current attacks come through many means such as Server-side attacks (Traditional), Browser & Plugin Flaws and Client-side attacks (XSS, CSRF) [11].

As we have only seen the tip of the iceberg with Web 2.0 worms. A fundamental review of web 2.0 security challenges were raised in Lawton [13]. We observe that these new, interactive, multimedia-rich forms of communication provide effective means for extremists to promote their ideas, share resources, and communicate among each other [9].

Many of the current vulnerability countering mechanisms address one or few specific issues. In a simple scheme for eliminating a wide range of script injection vulnerabilities in applications built on top of popular Ajax development frameworks such as the Dojo Toolkit, *prototype.js*, and AJAX.NET, Livshits and Úlfar developed a framework for code injection attacks [7].

Received August 20, 2008

Browser cache and history are intended to be private in the normal stream, yet it's not difficult for malicious Web sites to "sniff" cache entries on visitors' computers and then use that information to more accurately deceive them [15]. This leads to pose a major un-resolving issue to the research community.

9. Conclusions

Web 2.0 applications have moved the Internet forward and help fulfil the promise of more interactive functionality and community building [8].

The open nature of Web 2.0 presents significant challenges to the traditional enterprise approach to controlling intellectual property and proprietary content [6]. However, security is not usually considered. The increase in functionality and interactivity has increased the ways in which an application can be attacked successfully.

The enhanced option for effective and efficient security auditing is a vulnerability scanner which automates the crawling of websites to identify weaknesses. To do so the engine with the capabilities of parsing and executing the JavaScript, such crawling is inaccurate is required.

The analyst warns that Web 2.0 applications could herald widespread identity theft and transaction fraud, give malware a new infection super highway and erode social networks. It doesn't cease to amaze how many organizations do not give a serious view to security concerns of Web 2.0.

It is imperative that the use of technology is done judiciously and learn how to manage risk with all website applications. Until security is part of the complete software development lifecycle, Web 2.0 applications will remain insecure and can increase the potential for harm [8].

References

1. Acunetix, Are AJAX Applications Vulnerable to Hack Attacks?. In: http://www.acunetix.com/websitesecurity/ajax_applications.pdf
2. Are AJAX Applications Vulnerable to Hack Attacks?: The Importance of Securing AJAX Web Applications, In: http://whitepapers.zdnet.com/paper.aspx?docid=346549
3. Benjamin Livshits, Úlfar Erlingsson, Using web application construction frameworks to protect against code injection attacks, In: Proceedings of the 2007 workshop on Programming languages and analysis for security. Conference on Programming Language Design and Implementation archive 2007, p. 95-104.
4. Blogging and the Wisdom of Crowds, In: http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=3.
5. Brian Chess, Yekaterina Tsipenyuk O'Neil, Jacob West, JavaScript Hijacking, March 12, 2007. In: http://www.net-security.org/dl/articles/JavaScript_Hijacking.pdf
6. Gartner, Gartner warns Web 2.0 will force business to re-examine approach to IT security,2007, In: http://www.gartner.com/it/page.jsp?id=511944
7. Gerald Arcuri, Web Application Firewalls: Application Protection and Much More. Business Market Strategies, Volume 1, Number 13.
8. HP, Securing Web 2.0: are your web applications vulnerable?, October 2007, In: http://i.i.com.com/cnwk.1d/html/itp/HP-4AA1-5390ENW_Securingweb2-0.pdf
9. Hsinchun Chen, Sven Thoms, T. J. Fu, Cyber Extremism in Web 2.0: An Exploratory Study of International Jihadist Groups. In: IEEE International Conference on Intelligence and Security Informatics, 2008.
10. JavaScript Hijacking Vulnerability Detected, In: http://www.fortify.com/advisory.jsp
11. IBM Software group. Understanding the Top Web 2.0 Attack Vectors. 5.5.2008. In: http://www.us.ibm.com
12. Jensen, M., Gruschka, N., Herkenhöner, R., Luttenberger, N.: SOA and Web Services: New Technologies, New Standards - New Attacks. In: The 5th IEEE European Conference on Web Services (ECOWS 2007), Halle (Saale), Germany, November 2007, p. 26-28.
13. Lawton, G., Web 2.0 Creates Security Challenges, IEEE Computer, Vol. 40, Issue 10, Oct. 2007, p. 13 – 16.
14. M.D. Ciocco, N. Toporski, and M. Dorris, "Developing a Synchronous Web Seminar Application for Online Learning", SIGUCCS conference, ACM Press, Monterey, CA,USA, November 6-9, 2005, p. 36-39.
15. Markus Jakobsson, Sid Stamm. Web Camouflage: Protecting Your Clients from Browser-Sniffing Attacks, IEEE Security & Privacy, November/December 2007. Vol. 5, No. 6, p. 16-24.
16. Michael Sonntag, Ajax Security in Groupware, Proceedings of the 32nd

Received August 20, 2008

EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO-SEAA'06, 2006.

17. N.S Chen., Kinshuk, H.C Ko, and T. Lin, ”Synchronous Learning Model over the Internet”, Proceedings of the 4th IEEE International Conference on Advanced learning Technologies 2004, IEEE, 30 Aug.-1 Sept. 2004, p.505-509.
18. S.A. Plefsis, “Synchronous E-Learning”, In: http://www.plefsis.com/en/e-learningsync.htm.
19. Websense Security Labs. Research highlights Q3-Q4, 2007.
20. Yen-Ting Lin, Yi-Chiun Chi, Lien-Chien Chang, Shu-Chen Cheng, Yueh-Min Huang, A Web 2.0 Synchronous Learning Environment using AJAX, Ninth IEEE International Symposium on Multimedia 2007 - Ninth IEEE International Symposium on Multimedia, 2007, p. 453-458.

Received August 20, 2008