Proc. of the Intl. Conf. on Advances in Computing and Information Technology-- ACIT 2014

ISBN: 978-981-07-8859-9 doi: 10.3850/ 978-981-07-8859-9_24

Vulnerability of Web-Storage in HTML5 for Web and Mobile Application

Jaeho Choi, Scott Uk-Jin Lee

***Abstract*—HTML5 is not a new version of the existing mark-up language, but a new paradigm for developing web and mobile applications where various new concepts are introduced to improve compatibility and usability. Web-Storage is the one of new features in HTML5 that enables effective client-side storage and retrieval of the frequently used data. However, it has significant security problems which need to solve for providing safe web and mobile environment. Currently, there is lack of analysis and research to detect and prevent attacks on web storage and this can causes critical security threats to web and mobile application developed using HTML5. In this paper, we are going to practically study and analyze possible attacks on web storage as a first step towards ensuring the safety of web and mobile applications based on HTML5.**

Keywords— vulerability, HTML5, web-storage

I. Introduction

HTML5 is not a new version of the existing mark-up language, but a new tool and standard to develop web-sites, web applications, and mobile applications in the era of mobile devices and cloud computing. It is composed of the widely used HTML 4.01, XHTML 1.0 for semantic web, and the improved DOM Level2 HTML. It also enables easy and effective viewing of multimedia contents within a browser [2,3].

As Bob Muglia, the Microsoft President in charge of the company's server and tools business, said, "Silverlight will continue to be a cross-platform solution, working on a variety of operating system/browser platforms, But HTML is the only true cross platform solution for everything, including (Apple's) iOS platform" the biggest strength of HTML5 is the compatibility. In the existing web environment, the use of Flash and Java has reduced the platform compatibility. In order to resolve this problem, HTML5 improves the platform compatibility by analyzing different functions included in various browsers and enables the proper use of them. Thus, it can offer a consistent view to every operating systems and various smart devices.

Jaeho Choi / Dept. of Computer Science and Engineering
Hanyang University
Sa 3-dong, Sangnok-gu Ansan-si, Gyunggi-do, Korea
jaeho34@hanyang.ac.kr

Scott Uk-Jin Lee / Dept. of Computer Science and Engineering
Hanyang University
Sa 3-dong, Sangnok-gu Ansan-si, Gyunggi-do, Korea
scottlee@hanyang.ac.kr

In addition, it ease the supply of multimedia and various other contents. HTML5, which has the strong platform compatibility and efficient multimedia related functions, is expected to be widely applied after its final standardization on 2014.

Besides multimedia related functions, HTML5 provides many new functions to enrich the web and mobile environments. Among many, the Web-Storage enables the off-line use of web and mobile applications [7]. The existing web applications cannot be used off-line as they require a sever connection, but Web-Storage enables the off-line use by saving and backing up the required data in the client-side. The Web-Storage also enables the off-line use of the hybrid applications which is created to use web applications in mobile environment. A representative example for the usage of the Web-Storage is a mobile application, G-mail, developed by Google. In addition, Web-Storage can relieve traffic congestion on Web as the data is distributed from the server to client and web application can request data directly from the client-side. According to these advantages, the utilization of Web-Storage in web and mobile applications is expected to increase rapidly. However, Web-Storage can create serious problems if its vulnerability is not considered and solved.

An OWASP(Open Web Application Security Project) document, that announces the top 10 vulnerability of web application annually, still mentions the vulnerabilities such as XSS(Cross Site Scripting) and Injection. There are still lack of solution to these vulnerabilities in HTML5 as well [4]. The existing vulnerabilities often attacks server-side to extract data, but the use of Web-Storage will extend the scope of such problems from a server-side to a client–side. These problems have to be solved for HTML5 to be used properly. Therefore, in this paper, we study and analyze the possible attacks on Web-Storage as well as appropriate prevention methods as a first step toward ensuring the safety of Web-Storage usages.

II. Types of HTML5 Web-storage

One of essential functions for web and mobile applications is to save data and use saved data when required, but there was not an adequate storage mechanism for client–side prior to HTML5. Although the Cookie is used to store fragments of data in client-side, it is only useful in specific environment due to its characteristics. Basically HTML5 Web-Storage can be considered as an improved version of Cookie where it defines two types of key-value storage, Local and Session Storages [5].

Local Storage is a mechanism that operates very similar to the storage system of a desktop application. Data stored in Local Storage is permanent and can be used at any time by the application that created it . Session Storage is a mechanism

Proc. of the Intl. Conf. on Advances in Computing and Information Technology-- ACIT 2014

ISBN: 978-981-07-8859-9 doi: 10.3850/ 978-981-07-8859-9_24

which allows the access to the data only within a session length of a particular web page. The stored data can only be used by a single window or tab of a web browser and is maintained until the relevant window or tab is closed .

In addition to these two type of storage, HTML5 web-storage consists of a light-weight database based on SQLite and an IndexedDB for client-side data storage. Different from the Local and Session Storages, the light-weight database are made for bulk storage of data in client-side. It is similar to the database of server-side but light weighted for client-side storage. The IndexedDB on the other hand can be considered as an expansion to the local storage in its capacity.

In HTML5, these client-side storage mechanisms are newly offered and will be widely used in web and mobile application in the future.

III. Current problems of the Web-storage

Although the web storage provides effectiveness in establishing a web and mobile environment, it also generates problems. In order to control the web storage, JavaScript with very simple API(Application Programming Interface)s are used as shown in the Table 1. With such a simple JavaScript API for controlling web storage, attackers can easily manipulate and exploit data by inserting JavaScript codes . In addition, the users of web applications with implemented web storage do not know what kind of data is stored and have control over it [6]. Hence, important data or information can be stored regardless of user intensions and can be stolen by attackers. This clearly indicates that HTML 5 web storage has serious security vulnerability.

HTML5 web storage, being a key-value type, stores a key as a string variable and supports various format, such as JavaScript objects and JSON, as the corresponding value. Hence, except preventing the access from the other domain, the web storage allows the storage of any data in the client-side without a limit. For example, it can be used to store writings in a bulletin board, various important information, mails, and security certifications. With a simple JavaScript codes as shown below, every key to the data stored in the web storage can be retrieved.

TABLE 1. API to control Web-Storage

Function	Description
key(index)	Inquiry key of relevant index
setItem(key, value)	Set a item with key, value
getItem(key)	Get a value of the item from key
removeItem(key)	Remove an item of relevant key
clear()	Remove all items

```
for (i=0; i<localStorage.length; i++) {
    key = localStorage.key(i);
    pairs += "key:"+key+" value:“
                +localStorage.getItem(key);
}
console.log(pairs);
```

The above JavaScript code utilizes the key() function from the APIs shown in the Table 1. If one can access the relevant domain, the keys stored in a web storage can easily be obtained with the above code and the corresponding values can also be acquired without any difficulties. Currently, the web storage of HTML5 lacks an adequate tool support to detect and prevent such unauthorized access from outside. Attackers can freely access and exploit information or data from web storage by staying between a server with vulnerabilities and users and inserting a simple JavaScript codes. Furthermore, attackers can forge a specific website and insert malicious codes to the clients' web browsers or operating systems on download to steal data and information from web storage. The main problem is that users often do not realize the exposure of their information or data and, as a result, use the attacked web storage continuously, This is a serious problem and can cause a serious damages.

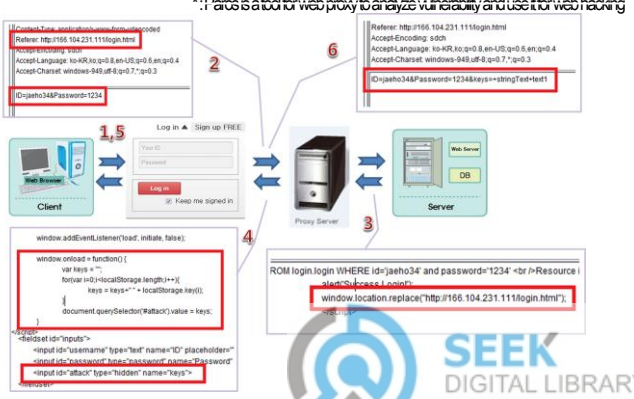
IV. Experiments

In order to grasp the mentioned problems more clearly, a simple example with Cross-Site Scripting (XSS) is conducted in a web environment. Details of the experiment environment are illustrated in the Table 2. whereas the whole structure of the experiment is shown in Figure 1.

Table 2. Details for the experiment with XSS to analyze vulnerability of local storage

Category	Description
Server	Apache server (Ubuntu 12), PHP
Protocol	http (Hypertext Transfer Protocol)
Client	Personal Computer (Chrome browser)
Tools for analysis	Paros*
Proxy server	Stay between server and client

*:Paros is a tool for web proxy to analyze vulnerability and use it for web hacking



Proc. of the Intl. Conf. on Advances in Computing and Information Technology-- ACIT 2014

ISBN: 978-981-07-8859-9 doi: 10.3850/ 978-981-07-8859-9_24

Figure 1. Obtain keys from local storage by inserting scripts

1. A user connects to a website for log in
2. A user inserts ID and Password then clicks a button
3. An attacker staying between a server and a client modifies a packet sent to the server making user log in to fail
4. When the server responds to the log in request from the client, the attacker inserts malicious scripts to steal data or information from the user
5. The user doubts the log in failure, but does not consider it seriously and tries again
6. At this time, the attacker obtains all the keys stored in a local storage to retrieve data or information they want.

As a result of above a simple experiment, the attacker can gain information or data they want easily by using keys stolen if the user activates a function for log-in automatically, and this data is stored in local storage. Like this situation it is possible to move to domain for the local storage through inserting tiny scripts and get keys and internal values. Besides this experiment with XSS, it is possible to attack with SQL injection to detour log-in and with malicious codes to get data in the same environment. We can see to take a look at process of a simple experiment with tiny tool, Paros, about a fact that the user's local storage is exposed easily. Image that if critical information, public authentication, mails and so on, is stored in local storage, the information may be exposed more and more, but the user does not know about it and feels useful of this function because they do not need to insert for certification to access their information all the time. It is revealed that the vulnerability of web storage surely exists through the experiment and there is no solution to prevent with it.

There are almost of all cases about vulnerability of web to attack to the server, so it tries to strengthen security of the server. In this situation, the attacker can attack to the client because of new features in HTML5. Thus, it may increase to approach to client's side for attack more than the server intensified for security. However, security equipment in client's side has only simple defensive measures inside of browsers or firewall inside of operating system. Unfortunately, that is all.

Thus, to solve vulnerability of security with web storage, we need programming to supplement problems in the server before everything else. When the server responds to the client's request, we have to make programs so scripts cannot be inserted by the attacker. We also recommend using HTTPS protocol to encrypt all of packets requesting of the client and responding of the server. Those equipment can somewhat supplement problems, but it is not still sufficient to defend against tools evolved progressively and the knowledge standard of the attacker. So we need to establish methodology and strong equipment to protect client's side from attack.

v. Conclusion and Future works

HTML5 is innovative technology whoever can approach to it by improving accessibility of web. Especially, a feature of web storage may be a large part of developing web and mobile applications. But it is a new technique, so problems for security still exists.

As a result of the experiment in this paper, we get a conclusion that web storage is not only exposed to insert simple JavaScript codes, but also other attacks. First of all, the main problem is that the user using relevant application is not aware of a fact which their information is exposed. The user who does not know about it can store their critical information or data in web storage. It is not sufficient that study, research and development to solve this problems is not going on widely until now because of that web storage is a new feature. Therefore, we need some equipment to be able to protect user's information, turn the web storage on or not, detect access not to permit. We also need to establish methodology for web storage that is used properly and safely and guideline for developer to reduce problems for security.

Moreover, the advent of hybrid application to improve compatibility and accessibility among platforms using web-kit, HTML5, CSS3, JavaScript in mobile environment may accelerate the amount of used with web storage. But, as penetration rate of HTTPS in mobile environment is lower than web and PC environment, the web storage is easy to be attacked [1]. Nevertheless, mobile applications need this feature because it happens to exchange among windows or screens frequently unlike web applications in UX(User eXperience) and UI(User Interface). If mobile applications are developed by HTML5, they need to use web storage to store or load data when the screen is exchanged. It is useful at this time, so we have to research and develop solutions to figure out access without permission and control data between mobile operating system and browsers.

Future works of this paper, we take a plan to make solutions with two considerations according to above mention. First, for the user, we are going to make a solution to protect their information and data and detect about access. Second, for the developer, we are going to make methodology or guideline to supplement vulnerability. Finally, we would like to make the world of web and mobile environment more safely, more reliably.

References

Proc. of the Intl. Conf. on Advances in Computing and Information Technology-- ACIT 2014

ISBN: 978-981-07-8859-9 doi: 10.3850/ 978-981-07-8859-9_24

[1] C. Amrutkar, P. Traynor, and P. C. van Oorschot. Measuring ssl indicators on mobile browsers: extended life, or end of the road? In *Information Security*, pages 86-103. Springer, 2012.

[2] R. Berjon, T. Leithead, E. D. Navara, E. O'Connor, and S. Pfeiffer. Html5-a vocabulary and associated apis for html and xhtml. *World Wide Web Consortium*, 11, 2012.

[3] D. Bogaard, D. Johnson, R. Parody, et al. Browser web storage vulnerability investigation: Html5 localstorage object. The 2012 International Conference on Security and Management, 2012.

[4] K. Choi. *Web Hacking and Defending.* BurningClass. Freelec, 2008.

[5] J. Gauchat. *HTML5 for Masterminds, 2nd Edition*. HTML5 for Masterminds, 2nd Edition Series. BookBaby, 2012.

[6] M. Schmidt. Html5 web security v1. 2011.

[7] W. West and S. M. Pulimood. Analysis of privacy and security in html5 web storage. *Journal of Computing Sciences in Colleges*, 27(3):80-87, 2012.

About Author (s):

I am a Master's Student in the Department of Computer Science and Engineering at Hanyang University, Republic of Korea. I did my BSc at the same university. My area of interests and research are Web and Cloud computing and Software Engineering.

I am an assistant professor in the Department of Computer Science and Engineering at Hanyang University, Republic of Korea. I did my BE and PhD at the University of Auckland, New Zealand. My research interests are in the area of web, semistructured data, formal methods, and software engineering.