

Analysis of Security Testing with Local Storage Inspector

Bhanu Sharma¹, Dolly Trivedi²

¹Assistant Professor, Computer Science & Engineering Department
Chandigarh Group of Colleges Technical Campus, Jhanjeri Mohali (Punjab) India

²Assistant Professor, Computer Science & Engineering Department
Swami Vivekananda Institute of Technolog, Secunderabad India

Abstract- This paper is based on Website security testing with the help of local storage inspector. Basically most of the developer when develop the website then they do some mistakes during the code because off that the website will not fully secure and it can be hacked easily. During the Software quality assurance people when check these websites then find out these mistakes. Most of the software companies do not prefer to hire testing team as they only work with developers and developers themself do the testing. This paper based on the literature survey of the local storage inspector elements and provide the information to developer and tester.

Keywords- Local Storage, inspect elements, Keys, Value, Roles, Data, Untrusted, browser etc.

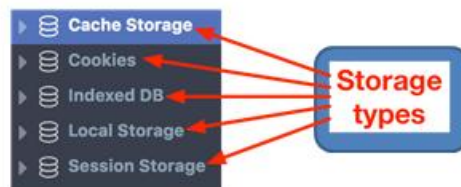
I. INTRODUCTION

Local Storage also known as Web Storage or Offline Storage is a mechanism to store data as key/value pairs tied to a domain and enforced by the same origin policy (SOP). There are two objects, local Storage that is persistent and is intended to survive browser/system reboots and session Storage that is temporary and will only exists until the window or tab is closed.

On average browsers allow to store in this storage around 5MB per domain, that compared to the 4KB of cookies is a big difference, but the key difference from the security perspective is that the data stored in these two objects is kept in the client and never sent to the server, this also improves network performance as data do not need to travel over the wire back and forth [1][3][4][5][6][7].

II. TYPES OF LOCAL STORAGE

There are four types of Local storage [2]



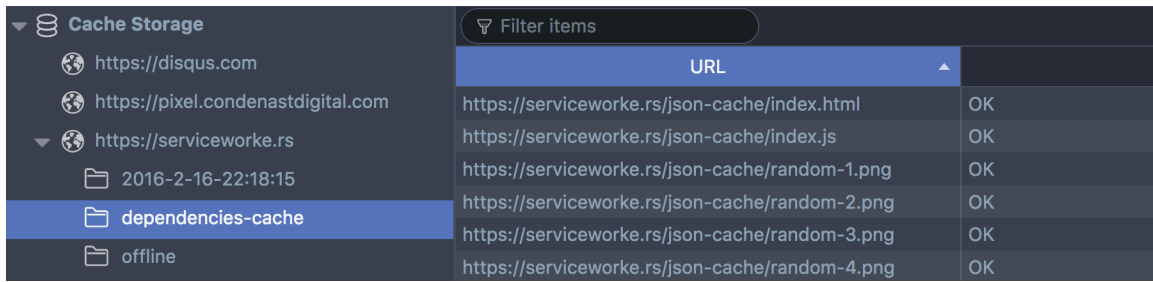
The Storage Inspector enables you to inspect various types of storage that a web page can use. Currently it can be used to inspect the following storage types:

- Cache Storage — any DOM caches created using the Cache API.
- Cookies — All the cookies created by the page or any iframes inside of the page. Cookies created as a part of response of network calls are also listed, but only for calls that happened while the tool is open.
- IndexedDB — All IndexedDB databases created by the page or any iframes inside the page, their Object Stores and the items stored in these Object Stores.
- Local Storage — All local storage items created by the page or any iframes inside the page.
- Session Storage — All session storage items created by the page or any iframes inside the page.

2.1 Cache Storage

Under the Cache Storage type you can see the contents of any DOM caches created using the Cache API. If you select a cache, you'll see a list of the resources it contains. For each resource, you'll see:

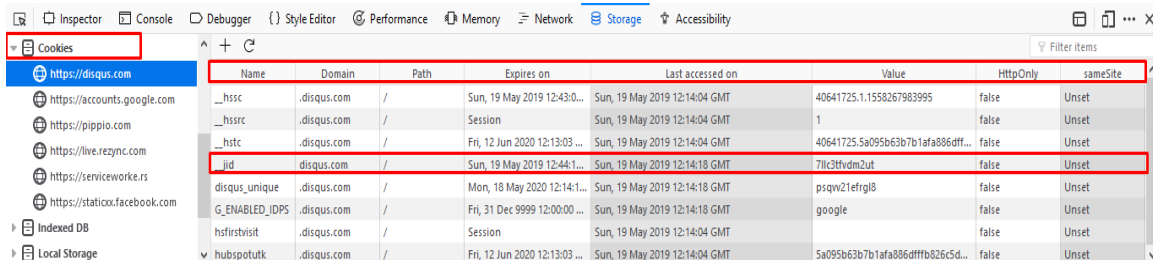
- The URL for the resource
- The status code for the request that was made to fetch it.[2]



2.2 Cookies

When you select an origin inside the Cookies storage type from the storage tree, all the cookies present for that origin will be listed in the table widget. The cookies table has the following columns:

- Name — The name of the cookie
- Path — The path property of the cookie
- Domain — The domain of the cookie
- Expires on — The time when the cookie will expire. If the cookie is a session cookie, the value of this column will be "Session"
- Last accessed on — When the cookie was last read
- Created on — When the cookie was created
- Value — The value of the cookie
- HostOnly — Is this cookie a domain cookie: that is, the domain value starts with a "."
- Secure — Is this cookie a secure cookie
- HttpOnly — Is this cookie HTTP only
- sameSite — Is this cookie a same-site cookie? Same-site cookies allow servers to mitigate the risk of CSRF and information leakage attacks by asserting that a particular cookie should only be sent with requests initiated from the same registrable domain.[2]



2.3 Indexed DB

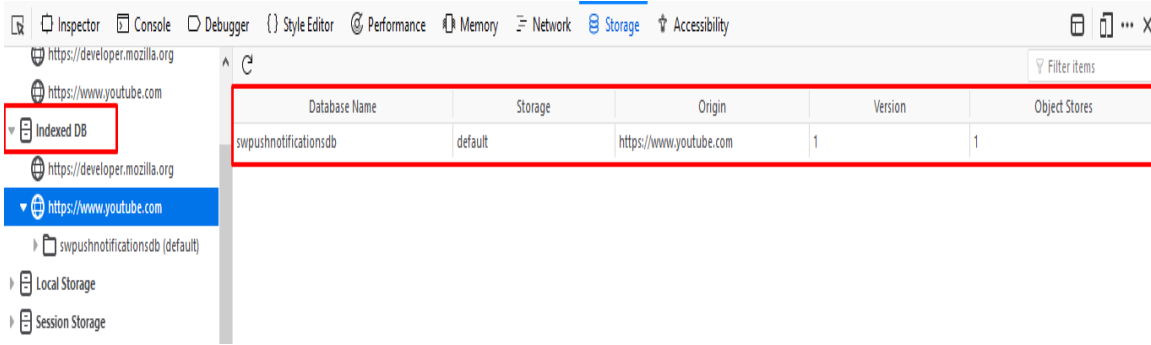
When you select an origin inside the Indexed DB storage type in the storage tree, the table lists the details of all the databases present for that origin. Databases have the following details:

- Database Name — The name of the database
- Storage — The storage type specified for the database
- Origin — Its origin
- Version — The database version
- Object Stores — Number of object stores in the database

When an IndexedDB database is selected in the storage tree, details about all the object stores are listed in the table. Any object store has the following details:

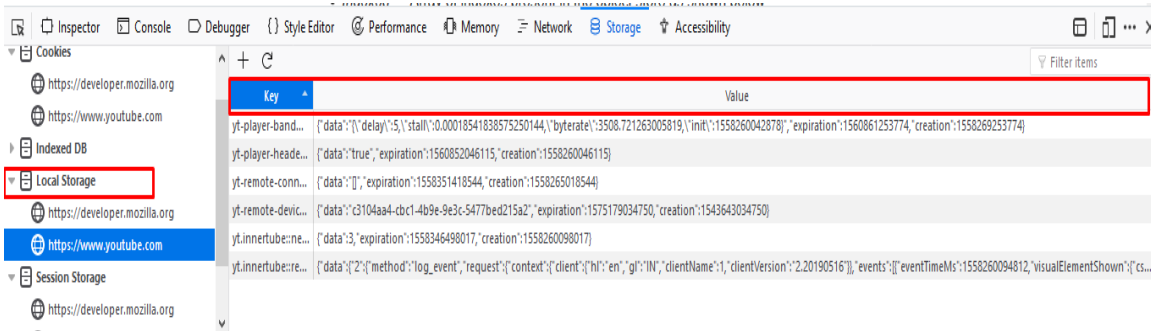
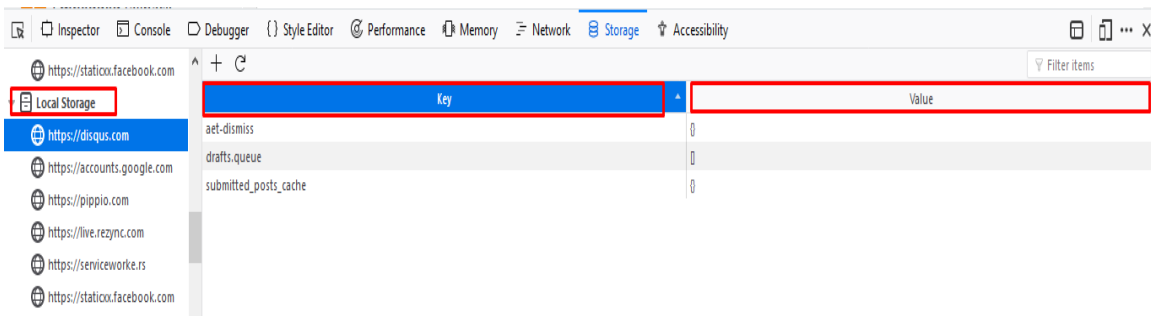
- Object Store Name — The name of the object store
- Key — The keyPath property of the object store.
- Auto Increment — Whether auto increment is enabled
- Indexes — Array of indexes present in the object store

When an object store is selected in the storage tree, all the items in that object store are listed in the table. All items have a key and a value associated with them and If the database cannot be deleted (most commonly because there are still active connections to the database), a warning message will be displayed in the Storage Inspector. [2]

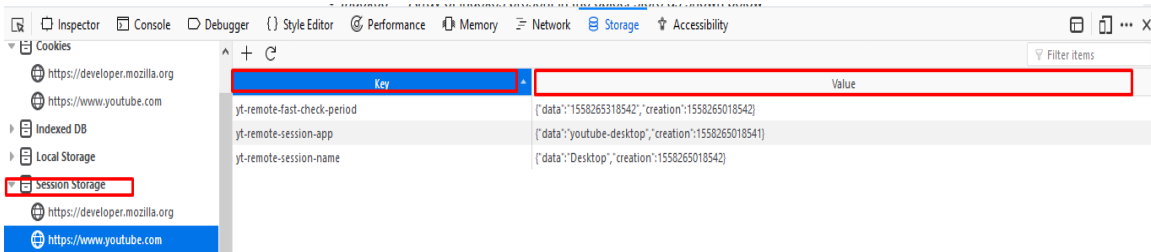


2.4 Local storage / Session storage

When an origin corresponding to local storage or session storage is selected, the table will list the name and value of all the items corresponding to local storage or session storage.



You can edit local and session storage items by double-clicking inside cells in the Table Widget and editing the values they contain. [2]



III. HOW TO TEST

3.1 Black Box testing

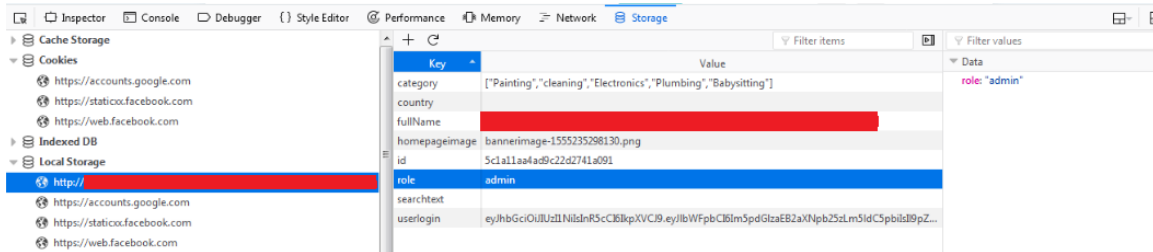
Black box testing for issues within the Local Storage code is not usually performed since access to the source code is always available as it needs to be sent to the client to be executed.

3.2 Gray Box testing

First of all, we need to check whether the Local Storage is used.

For example

- Access to every element in localStorage with JavaScript.
- Using Firefox with the Firebug add on you can easily inspect the localStorage/sessionStorage object in the DOM tab.
- How improper use of user input and lack of validation can lead to XSS attacks.



If tester change the role related to the website then find out the particular bug or mistakes so when developer develop the code then few things need to be take care so that no one can easily break the security of the website.[8][9][10][11][12]

IV. REFERENCES

- [1] [https://www.owasp.org/index.php/Test_Local_Storage_\(OTG-CLIENT-012\)](https://www.owasp.org/index.php/Test_Local_Storage_(OTG-CLIENT-012))
- [2] https://developer.mozilla.org/en-US/docs/Tools/Storage_Inspector.
- [3] https://www.owasp.org/index.php/HTML5_Security_Cheat_Sheet.
- [4] <https://mage2.pro/t/topic/1659>.
- [5] <https://disqus.com/features/engage/>.
- [6] <https://serviceworke.rs/strategy-cache-only.html>.
- [7] <https://developers.google.com/chrome-developer-tools>
- [8] https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project.
- [9] <http://www.w3.org/TR/webstorage/>
- [10] <https://mathiasbynens.be/notes/localstorage-pattern>.
- [11] <https://disqus.com/features/engage/>.
- [12] <https://serviceworke.rs/strategy-cache-only.html>