Vulnerability Analysis for Credentials Management in Web Browsers

Mahmood Ahmad[1], Muhammad Idris[1], Zeeshan Pervez[2], Sungyoung Lee[1]
[1] Department of Computer Engineering, Kyung Hee University
Republic of Korea
[2] School of Computing, University of the West of Scotland, Paisley, PA1 2BE, UK

[1]{rayemahmood,idris,sylee}@oslab.khu.ac.kr,
[2]zeeshan.pervez@uws.ac.uk

Abstract. Provision of automation with ease in human life has been a prime focus of computing discipline. In this paper we have focused on cloud computing in general and interaction of home users with browser in particular. Majority of cloud service providers offer free storage space along with synchronization services giving comfort for keeping data consistent on various devices. These storage reservoirs are protected with valid user credentials and can be accessed with almost any web browser or vendor specific desktop application. With ever increased number and usage in browser based applications (emails, social websites, banking); the utility to remember password is becoming inevitable for home user causing a malicious user to hack this information with even greater motivation. We have discussed storage mechanism for these credentials on browsers which have not been changed quite a long especially after the inception of cloud computing. Compromise on these credentials is not an old evidence; however, this overlooked issue has a higher risk of security especially after a user has placed personal contents on the cloud storage.

Keywords: Password Security, Credentials, Cloud Computing

1 Introduction

The first impression of cloud computing is all about elastic storage and extraordinarily fast computational hardware resources which are available and scalable as per user demand round the clock. Provision of this ready-made infrastructure and a hub of services like IaaS, PaaS, SaaS [4],[7] by cloud service providers (CSPs) have alleviated burden for managing privately owned equipment and associated human resources at small and medium as well as enterprise level organizations(SMEs). Interaction of CSP is not limited with SMEs alone, home user is another beneficiary of this cloud. Google Drive, Dropbox, SkyDrive, Apple iCloud are amongst free storage providers [13] where a user enjoys free storage and uploads contents not only for secure storage but for synchronization as well. In contrast to SMEs, the security concerns of a home user from cloud storage are not that much rigid. Free services like storage, emails, automatic synchronization of data on all devices (laptop, smart phone, home and office PC) are enough motivational to use these services. Access to these storage services are protected with valid user credentials and the same can be accessed through web browsers or vendor specific desktop applications. From the home user's perspective, accessing cloud through browser is similar to that of email or banking website. Security on these different web accounts is ensured as long as login credentials are not compromised but there are number of traps through which it can slip away [1], [24], [33]. Before we discuss the browsers limitations, it is also very important to know the general behavior of user with his browser that further weaken the overall defense.

The number of passwords per user were limited with email or some business account in early days of internet. Over a period of time and with the induction of social websites, various email accounts and banking portals, this number has been raised to 6.5 per user in 2007 as stated by Microsoft research study [11]. Although the exact figure is unknown but it can be well imagined that from 2007 to 2013, the number of passwords per user must have been increased. In 2011 another empirical study on passwords [36] revealed that majority of users still opt for password with length less than 7 characters or select easily guessed words and very few include special characters or Greek words in it. Sharing of same password with more than one accounts and repetition of same password again and again is another poor practice of moderate users [12]. Selecting sophisticated passwords and then memorizing it came up with the idea of one Master Password [19], [17]. This idea never nourished mainly due to single point of failure or lack of user trust in these services. Facial recognition, iris scanning or finger printing are classified as biometric authentication [27], [29]. These are alternative ways to handle the authentication problem but as an additional cost of equipment this idea never achieved widely accepted popularity.

With the prevailing limitations on this issue, majority of users store passwords for different web accounts on their personal machines (Laptops and Home PCs) and at a first look there is no harm in doing this because stored password is not the only information which is critical; there is a lot more sensitive and personal information resident on the same machine which includes personal photographs, office documents and media files. Stealing few bytes of personal information from victim machine could be an easy job for some hacker but same attempt usually do not work well when the information size grows in gigabytes. Transferring bulk of data this way over the internet could slow down the network speed or can affect overall system response for other applications. This unexpected behavior of system can alarm its owner for some basic investigation or system restart as a last resort. Besides, the user has placed fairly a large contents of information on free cloud storage (for backup or synchronized replication) which can be accessed with valid login credentials. In this situation the hacking attempt is limited with key information (login credentials) alone from victim machine. The stolen credentials are then used without being noticed as an unauthorized user while accessing the information stored on cloud. Due to these reasons, motivation behind harvesting these stored credentials is becoming higher in hacking community.

The remainder of this paper is organized as follows. Section 2 describes browsers and their storage mechanism of users' credentials. Section 3 highlights exploitation trends. Section 4 is about related work. Testing environment is presented in Section 5. Section 6 concludes the paper.

2 STORAGE MECHANISM OF BROWSERS CREDENTIALS

Browsers are widely opted medium for accessing the internet and over the past few years' induction of new browsers are increasing the total count. Depending upon the usage share [37], [30] we have selected three popular browsers i.e. Google Chrome, Mozilla Firefox and Internet Explorer. Popularity of any browser application depends on its speed, look and feel, customization, plugins and various favors that can be used on number of devices like computers and smart phones. The feature of \remember password" in a browser is fairly a common knowledge and this password can be red easily in-case the owner has forgotten it and want to use it on some other machine. If the user machine is not protected with system password, physical possession of his/her machine will help anyone to steal stored password easily. In further discussion we will analyze the storage facility and mechanism of these passwords, their acquisition especially without acquiring the machine physically and its re-use to infiltrate into original user account.

Unauthorized acquisition of stored passwords is not a new practice as it can be witnessed even before the inception of cloud computing. Compromised credentials can help to view different web accounts but this access door now leads for cloud storage where volume of data is much bigger. If someone has access to cloud storage account, contents can be seen, copied or crippled very conveniently. This \someone" which can be labeled as a \hacker" has now bigger motivation to harvest stored credentials to try its luck whether he finds password especially for the cloud account. The complete scenario can be seen in figure 1 where interaction of user V (victim) with cloud storage and other web domains (including malicious or compromised legitimate websites) is shown. The possible attack scenarios is either through social engineering or having direct access of victim machine. Sequence of steps that are being followed by user H (hacker) to harvest credentials of user V are given in the section of testing environment, Section 5.

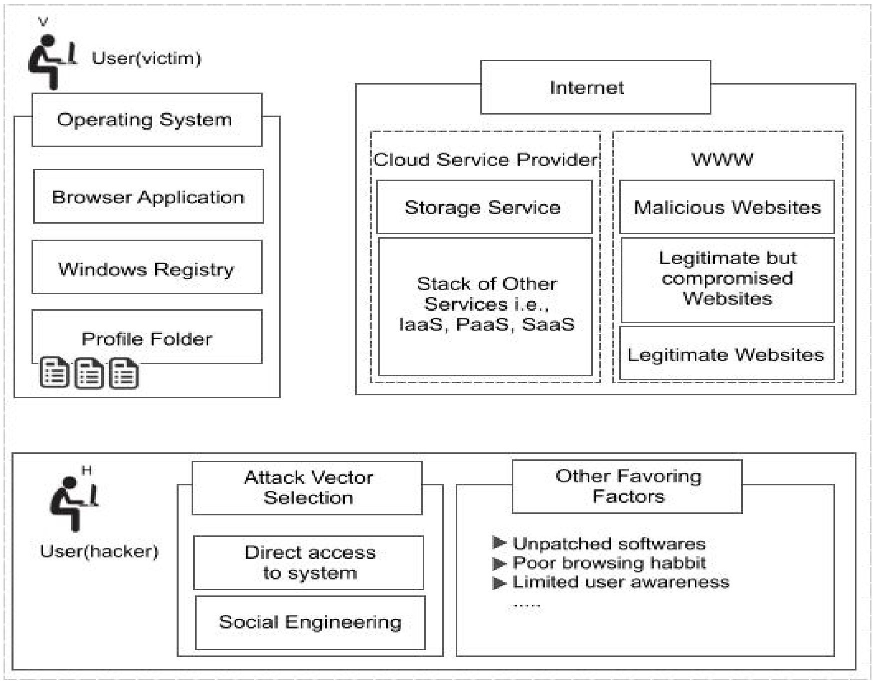

Figure 1: Abstract view of conventional internet access

The upcoming discussion on each browser is with respect to Windows-7 as an operating system and not in the order of browser popularity or ranking.

2.1 Mozilla Firefox

Any application which is installed on Windows Operating System copies its preferences, settings and profile folders at various locations including windows registry. These settings are used when the application starts or during its execution. The profile folder of Mozilla Firefox is created in the following directory.

C:\Users\(username[1])\AppData\Roaming\Mozilla\Firefox

With fresh installation of Mozilla Firefox (16.0.2 for windows), size of its folder is around 200 bytes. When the browser is launched its size grows in MBs (around 13 MB). This folder holds a detailed information on the browsing habit of a user, stored passwords, thumbnails, frequently used websites, bookmarks, downloads etc. in SQLite database files. SQLite is a server less DB library [31] which is used by Firefox to store. There are various tools and plugins available [18], [32] to browse SQLite files. We used the SQLite manager (plugin to Mozilla Firefox) [18] and SQLite Browser application [32]. Information on user credentials is stored in signons.sqlite along with the name of URL. At this point the only information revealed is the name of websites for which passwords are stored because the passwords are encrypted. Before moving on to the next step we have tested this hash value with popular hashing functions like MD5, SHA-1, SHA-2 to confirm the hashing algorithm used by Firefox but it implements its own algorithm to create this cryptic storage. On further analysis these encrypted values are not of fix length and change with every successful login.

For a particular website W, the valid credential C comprising password P and id K of user U, we have observed and confirmed successful login (over a number of times) with following behavior.

$FirstTimelogin => L\ (C_{P,K})$

$RememberPassword => FirefoxHashing(C_{P;K})$

For successive login, value of this password hash changes every time in SQLite database and

$Value\ (H_i\ (P))\ 6 = Value\ (H_{i1P;K}\ (P))$

After the value of this hash is changed, we used its previous value and updated it in SQLite database manually, the login was still a success. With this we concluded that a same value of password produces different cipher values and are not dependent with any other file or registry key. The later conclusion is confirmed when we tested same credentials (hash of user id and password) on another machine and it worked perfectly.

2.1.1 *Corollary*

With the exercise described above, it can be stated that producing hash value of password is one step towards safeguarding the user password but it cannot be considered as a panacea. Moving complete profile folder from one machine to another will even open the same windows and tabs of Mozilla Firefox and can reveal the all; however, size of this profile folder is a big barrier in transporting it over the internet with some social engineering technique. A novice user or someone who is not literate enough to monitor this activity can be deceived by copying this stuff with some automatic (auto-play) feature of removable media. Further work reveled that acquisition of a file key3.db (which resides in the profile folder) and information from three columns (formSubmitURL encryptedUsername,encryptedPassword) of signons.sqlite table are sufficient for the compromise.

2.2 Internet Explorer

Internet explorer has its own mechanism to store passwords and as a product of Microsoft; it can take an extra advantage of windows credentials and more complex integrated mechanism for most secure encryption. There are tools and scripts available [8], [9], [25], [26] not only to reveal these stored passwords but also explain the internal ow of algorithms. These tools and probing scripts are not trivial neither everyone can come and reveal the secret book easily as it demands through understanding of implementation and a fair knowledge of computer architecture; however, for a hacker or aggressive user if things are available on a click then there is no need to be worried about the underlying Greek.

Internet Explorer stores two types of passwords for its users which are; auto complete and HTTP Basic authentication. The first one is very common which is used in daily emails, library account, banking or some forum website etc. In HTTP authentication the password is required to logon some website and it is controlled by some proxy server or router. For in-depth understanding, the reader can refer to RFC 2617 [15].

2.2.1 *Password Storage for Auto Complete*

A user visits a website and acknowledges the "remember password" offered by Internet Explorer. This entry is encrypted and get stored in windows registry. The storage section of windows registry prior to Internet explorer version 7.0 was

HKEY.CURRENT_USER\Software\Microsoft\Protected Storage System provide

which has been changed to

HKEY.CURRENT_USER\Software\Microsoft\Internet Explorer\IntelliForms\StorageX

from version 7 onward. With fresh installation of internet explorer 9, the last segmentation of above registry key is not created until \remember password" utility is invoked by the user. After the segment of \StorageX" (where X is an integer value) is created, all subsequent passwords are stored in it. Deleting these values manually from registry will incapacitate the browser to recover and would not be able to login with auto complete.

We set up two machines, V and H with windows 7 and Internet explorer 9. On machine V, login credentials for few websites have been stored. Registry values against stored passwords (which are encrypted) are then exported from Machine V to Machine H to check if these registry values work or not. Feature of auto complete on machine H requires URLs at first hand which are not readable from this encrypted information. The second slice of information required to check the auto complete feature can be extracted from the browser history (of machine V) or user surfing habit on his domains of interest. It is assumed that now we have exact name of URLs and the encrypted registry values. The outcome is still not a success, which implies that there are few other missing pieces of information required to solve this jigsaw of auto complete. Further, we tested it with same windows login credentials but it never worked. To deal with this encrypted storage we used tools [8], [9], [25], [26] which reveal all the stored passwords with enumeration and in plain text.

2.2.2 *Password Storage for Auto Complete*

This feature has been introduced with internet explorer 7 onward. The passwords stored here are encrypted with windows cryptography function, after salting them with the text generated from GUID [23]. Windows provide credentials management function to deal with this type of password and uses the function "CredEnumeration". The code snippet can be found on [8].

2.2.3 *Corollary*

In comparison to Mozilla Firefox, the process of deciphering Internet explorer stored password is coupled with windows credentials and system bindings. It gives further strength and resilience to storage mechanism of Internet explorer but these bindings cannot withstand against small utilities [25], [9],[26] which can reveal the stored passwords. Drive by download [24],[33], social engineering techniques or physical access to system are few ways that can be used for this exploit.

2.3 Google Chrome

The Google Chrome; although being younger in its age has grabbed a large portion of browser market since 2008. Few efficient claims in its architecture and operative mode has really boosted itself amongst its peers. Unlike other browsers; each instance (Tab) of this browser runs in its own and as a separate process. Chrome managed it by placing each process in a sandbox where any abnormal behavior is dealt in isolation keeping other tabs to breathe normally. If we look at its methodology for storing passwords, it resembles with Firefox and Internet Explorer. The default installation folder of chrome profile is on this location

C:\Users\Username\AppData\Local\Google\Chrome\UserData\Default

Here, the complete user profile is stored in SQLite database files. Using the SQLite add-on [32] or standard browser [18] we can look into various SQLite files. The most important file is the login data which stores user name and password for different websites. This file can be investigated using SQLite browser [32]; however, passwords are encrypted and are not readable. Third party tools like ChromePassView [25] can be used to see actual values for stored password. The process of encryption and decryption of stored password has been secured in such a way that a user and machine which store it is the only combination for its retrieval. Strength of this feature has been achieved through Windows API of CryptProtectData [21]. On the other hand, if the machine is in physical possession and all stored passwords on Google Chrome are further protected with master password, the first attempt of viewing stored password using Chrome feature might fail even third party tools [25],[8] will not get any success. In this situation the feature of auto complete will still work but password field will show nothing except big dots for passwords. At this point, If password field is in focus with right click and option of inspect element is selected (built-in feature of Chrome), it will open the source code just below the web page. In this source code, changing type of password field to normal text will reveal the password in plaintext instantly.

2.3.1 *Corollary*

Winning the competition in browser market has focus on the look and feel with higher priority. Password protection scheme opted by Google Chrome has also proved itself to survive against basic hacking attempts. Other than third party tools the built in feature of inspect element is a simple provision to read stored passwords.

3 STORAGE MECHANISM OF BROWSERS CREDENTIALS

Computer administrator is the most privileged user of his machine having maximum permissions to execute programs (binaries), add, update or delete files and access to system registry. Guest users or remotely logged on users have lesser permissions and therefore cannot execute every program especially which requires access to system folder or manipulation with system registry (to survive system reboot). This feature of Operating System, supports avoiding installation of malicious or unwanted programs to some extent on user machine thus enhancing the overall security of system. Keeping in view this limitation as a barrier; the hackers somehow manage to transport malicious program on user (victim) machine and then wait for its execution by the system owner to exploit the maximum privileges. Social engineering is one popular example of this sugar coated trap. Here is a list of few techniques that can easily entice a moderate user to execute the program while being unaware of the consequences.

- Drive by download (Automatic execution of binaries by visiting malicious or compromised websites) [24], [33]
- Embedded macros in Microsoft office documents [22]
- Files with double extension (sample.exe.txt). By default windows hides file extensions, executable files can be displayed as text or image files with this technique.
- Alternate Data streams [20],[34]
- Plug and Play feature of removable devices like USB and CDs (Auto run and Auto play feature)
- Public documents on cloud that automatically gets synchronized on digital devices (Laptops, PCs, smart phones etc.)
- An executable file but having folder icon, where a user definitely gets tricked and double click it but it is too late now
- Un-patched vulnerabilities of browsers (Integer and Buffer overflow)[1], [2], [3]

Usage of any technique is equally effective for the transportation of stored credentials in user browser. In the next section we will present the practical demonstration where credentials from one machine are used on another machine.

4 RELATED WORK

The methodology of storing passwords in these browsers is same for the past few years even it has been highlighted as a security risk. The comparative analysis on browsers has been discussed in [6], where storage mechanism of login credentials and possible exploitation methods are discussed. The browser vulnerabilities invite attacks especially through Cross site scripting (XSS) and Reverse cross site scripting request (RCSR). The methodology of RCSR presented in [10] gives an overview that how stored credentials can be transported from a victim machine. The author in [6] has recommendations for the web developer to avoid auto-complete feature of password field while writing the code and disable password manager for the home user respectively. These recommendations appear as an escape solution of the problem. For better security, chromium segregates browser kernel and browser rendering separately as the browser kernel is responsible for managing persistent resources, such as cookies and the password database. The motive behind the chromium segregated architecture is to avoid execution of arbitrary code while visiting any malicious website [5]. This modular approach works until the user interaction is limited with chromium browser. Execution of malware by visiting dishonest websites from other browser is still a success to trespass into the information hive made by the chromium or any other browser.

In further discussion the author agrees that the chromium browser does not support for XSS or Cross site request forgery (CSRF) on websites that are safe but have vulnerabilities in them. The identified vulnerabilities of browsers has made its vendors to release patches or even a new release. It has been observed in [16] that Google Chrome and Mozilla Firefox releases new version after every six (06) weeks. In [6] various vulnerabilities of browsers have been discussed and in [14] a new design of browser has been propose but it lacks how to incapacitate usage of browser credentials on other machines. Keeping encrypted data on user machine does not guarantee that information is secure forever. Other than acquiring information of stored credentials which are encrypted, information available in readable format like bookmarks, downloads, frequently visited websites and cookies can be used to craft behavior oriented emails in social engineering. The realization and importance of user credentials has also been discussed in [28] but there is a lenient focus for auto-complete and stored passwords.

5 TESTING ENVIRONMENT

Discussion on credentials compromise with respect to each browser has already been covered. In this section we will present the attacking scenario in relevance with our previous discussion. The testing environment has been created with two machines V and H on two different networks and Windows 7 as an operating system. PC V will act as victim machine whose browser profile will be used on the second PC H, which will act as a hacking machine.

Both machines V and H have been prepared with fresh installation of Mozilla Firefox (version 16), Internet Explorer (version 9) and Google Chrome (version 23). Few websites that ask for login credentials have been visited only on machine 'V' so that stored credentials can be used on H as it is. It is assumed that the victim machine V has been injected with the exploit code that will actually transport the selected information from browser profile and registry keys of machine V to H. The Fig-1 shows an overall attack scenario where the user 'V' has placed contents on cloud `Step

1' followed by an attack in `Step 2'. Deployment of malicious code and fetching information from V to H falls in `Step 2' which is the acquisition of stored credentials. Unauthorized access of cloud contents and websites is part of `Step 3'. These websites are those ones, which have been visited on V at the start of this exercise Figure goes here After the profile folder and registry information has been ported on 'H' in step-2; it is only Mozilla Firefox that has been breached and the other two browsers survived this simple attempt. Table 1 shows the initial results with this trivial methodology of transporting the desired credentials from 'V' to 'H'. Results shown in table 1, are effective only for Mozilla Firefox and not for the other two browsers.

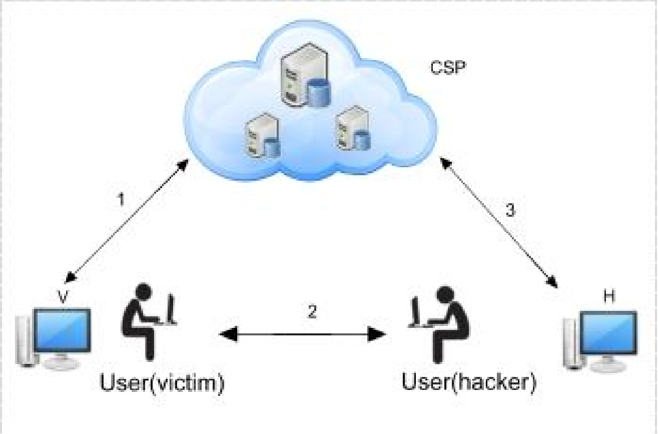

Figure 2: Attacking flow

Table 1: Credentials Effectiveness on remote-machine

Browser	**Target Credentials**	**Delivery Mechanism**	**Effectiveness**
Mozilla Firefox	SQLite and Key File	Email	Yes
Internet Explorer	Windows Registry	Email	No
Google Chrome	Registry and Profile	Email	Yes
Mozilla Firefox	SQLite and Key File	USB	Yes
Internet Explorer	Windows Registry	USB	No
Google Chrome	Registry and Profile	USB	No

In case of Internet Explorer (on machine V) values of registry keys have been made identical but the auto complete feature never worked. To check it’s binding with the windows

Logon credentials we changed the windows login password same on both machines but it never worked. For Mozilla Firefox, the received profile folder behaved the same. In next step we enhanced the exploit mechanism that initially meant to transfer desired credentials as it is, now it can work with stored credentials in windows registry for both IE and Chrome. It retrieves these passwords in plain while being on V, and with some basic encryption it can transforms them into non-readable (optional) stream while sending through email. The size of desired information was less than 50kb and it took un-noticeable time on machine V for retrieving, encrypting and sending back on H. Methodology opted in this step resembles as that of[8],[9] [25].

With this step of our exercise, the login credentials worked for the selected browsers. Mozilla Firefox has a limited resistance against trivial malicious attempt whereas Internet Explorer and Google Chrome survived it initially.

5.1 Offline Acquisition of Credentials

Acquiring victim machine physically has no protection if it is not safeguarded with system password. In case if the system is protected with password, still there is a way to bypass it. This can happen if the first boot device is the removable media (USB or CD). A small utility of UNetbootin [35] has been used with Ubuntu as an OS to boot the system V in Linux environment. The UNetbootin and utilities like it has the ability to convert removable media into live USB or CD to boot a system with desired OS. After the system V has been booted with Ubuntu, we have the access for the profile folder of Mozilla and Chrome. Acquiring registry information was tough though but still it was of no use which helped the Internet explorer to survive this offensive technique. The Chrome browser escaped too but only for its table storing encrypted data for login credentials; however, other information regarding bookmarks, downloads, cookies and history can be used for social engineering. Firefox behaved the same and showed minimal resistance amongst three.

Table 3: Summary Results

	IE	Firefox	Chrome
Offer password storage	Yes	Yes	No
Storage access with external tools	Yes	Yes	Yes
Storage access with external tools	Yes	Yes	Yes
Encrypted password storage	Yes	Yes	Yes
Profile effectiveness on other machines	No	Yes	No
Built-in password manager	No	Yes	Yes
Encryption binding with Windows credentials	Yes	No	Yes
Credentials storage (File system)	No	Yes	Yes
Credentials storage (Windows registry)	Yes	No	No
Access to credentials with other OS (Ubuntu)	No	Yes	No

Table 2: Credentials Effectiveness on remote machine

Browser	Target Credentials	Delivery mechanism	Effectiveness
Internet Explorer	Windows Registry	Email	Yes
Google Chrome	Registry and Profile	Email	Yes
Internet Explorer	Windows Registry	USB	Yes
Google Chrome	Registry and Profile	USB	Yes

5 CONCLUSION

The comparative analysis of Google Chrome, Mozilla Firefox and Internet Explorer just highlighted the impact and volume of disaster which has been amplified when a regular user is an active entity on cloud. Before opting for the cloud storage, the compromise on stored passwords was a little lesser which has now been increased to manifold with massive and organized cloud storage. The motivation to launch an attack is escalated but the modus operandi is almost the same. The paradigm shift with cloud computing completes with the triangular communication involving the data owner, service provider and the end user. Demand of security has been focused mainly on the other two and ignoring the end user. It is the time when a regular user needs to update his general awareness while moving on to the cloud and at the same time, the mechanism to avoid hazards associated with the stored passwords need to be re-engineered.

6 ACKNOWLEDGEMENT

This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency)" (NIPA-2014-(H0301-14-1003) and by the Industrial Core Technology Development Program (10049079), Develop of mining core technology exploiting personal big data) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea)"
.

7 REFERENCES

1. Mitre. cve-2006-7228, 2006. http://cve.mitre.org/cgi bin/cvename.cgi?name=CVE 2006-7228, 2006.

2. Mitre. cve-2007-3743, 2007. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3743, 2007.
3. Mitre. cve-2008-3360, 2008. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-3360, 2008.
4. M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A view of cloud computing. Communications of the ACM, 53(4):50{58, 2010.
5. A. Barth, C. Jackson, C. Reis, and T. Team. The security architecture of the chromium browser, 2008.
6. C. Boja. Security survey of internet browsers data managers. arXiv preprint arXiv:1112.5760, 2011.
7. R. Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6):599{616, 2009.
8. S. Exploded. Exposoing the password secretes of internet explorer. securityxploded.com, 2013.
9. S. Exploded. Internet explorer password revealer. http://www.ie-password revealer.com/, 2013.
10. M. Felker. Password management concerns with ie and Firefox, 2010.
11. D. Florencio and C. Herley. A large-scale study of web password habits. In Proceedings of the 16th international conference on World Wide Web, pages 657{666. ACM, 2007.
12. S. Gaw and E. Felten. Password management strategies for online accounts. In Proceedings of the second symposium on Usable privacy and security, pages 44{55. ACM, 2006.
13. E. Hamburger. Google drive vs. dropbox, skydrive, sugarsync, and others: a cloud sync storage face-of. http://www.theverge.com, OPTurldate = 10 January, 2013 " 2012.
14. S. Hangai, T. Hamamoto, and M. Kawamoto.Perceptual color on internet browser. In Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on, volume 1, pages 319 {322 vol.1, 2000.
15. IETF. Http authentication: Basic and digest access authentication. http://tools.ietf.org/html/rfc2617/, 2012.
16. F. Khomh, T. Dhaliwal, Y. Zou, and B. Adams. Dofaster releases improve software quality? an empirical case study of mozilla firefox. In Mining Software Repositories (MSR), 2012 9th IEEE Working Conference on, pages 179 {188, june 2012.
17. LastPass. The last password you'll have to remeber. http://lastpass.com/, 2013.
18. Lazierthanthou. Sqlite manager (firefox add on). https://addons.mozilla.org/en-us/firefox/ addon/sqlite-manager/, 2011.
19. H. Luo and P. Henry. A common password method for protection of multiple accounts. In Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on, volume 3, pages 2749 { 2754 vol.3, sept. 2003.

20. A. Martini, A. Zaharis, and C. Ilioudis. Detecting and manipulating compressed alternate data streams in a forensics investigation. In Digital Forensics and Incident Analysis, 2008. WDFIA '08. Third International Annual Workshop on, pages 53 {59, oct. 2008.
21. Microsoft. Cryptprotectdata function. http://msdn.microsoft.com/en-us/library/aa380261.aspx , 2013.
22. Microsoft. Office macros. http://office.microsoft.com/en-001/support/using-macros-to-speed-up-your-work-HA001019230.aspx , 2013.
23. M. MSDN. Credenumeratefunction. http://msdn.microsoft.com/en-us/library/aa374794%28VS.85%29.aspx , 2013.
24. J. Narvaez, B. Endicott-Popovsky, C. Seifert, C. Aval, and D. Frincke. Drive by-downloads. In System Sciences (HICSS), 2010 43rd Hawaii International Conference on, pages 1 {10, jan. 2010.
25. Nirsoft. Internet explorer and chrome password view. http://www.nirsoft.net ,2013.
26. Oxid.IT. Cain and able. http://www.oxid.it/cain.html , 2011.
27. N. K. Ratha, J. H. Connell, and R. M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. IBM Systems Journal, 40(3):614 {634, 2001.
28. B. Ross, C. Jackson, N. Miyake, D. Boneh, andJ. Mitchell. Stronger password authentication using browser extensions. In Proceedings of the 14th Usenix Security Symposium, volume 5, 2005.
29. S. Sanderson and J. Erbetta. Authentication forsecure environments based on iris scanning technology. In Visual Biometrics (Ref.No. 2000/018), IEE Colloquium on, pages 8/1 {8/7, 2000.
30. N. Share. Browser market share. http://marketshare.hitslink.com/browser-market-share.aspx?qprid=0&qpcustomd=0/ , 2012.
31. SQLite. Sqlite database. http://www.sqlite.org/ ,2012.
32. SQLiteBrowser. Sqlite database browser. http://sqlitebrowser.sourceforge.net/ , 2012.
33. B. Stone-Gross, M. Cova, C. Kruegel, and G. Vigna.Peering through the iframe. In INFOCOM, 2011 Proceedings IEEE, pages 411 {415, april 2011.
34. Symantec. Windows ntfs alternate data streams. http://www.symantec.com/connect/articles/windows-ntfs-alternate-data-streams , 2013.
35. UNetbootin. Unetbootin. http://unetbootin.sourceforge.net/ , 2012.
36. A. Voyiatzis, C. Fidas, D. Serpanos, and N. Avouris. An empirical study on the web password strength in greece. In Informatics (PCI), 2011 15th Panhellenic Conference on, pages 212{216. IEEE, 2011.
37. W3School. Browser statistics. http://www.w3schools.com/browsers/browsers_stats.asp , 2012.