Password management problems in IE and Firefox

The two parts of this article will present you an analysis of security techniques, risks, attacks and prevention of two widely used browser password management systems, the Internet. Explore and F

1. Introduction

The two parts of this article will present you an analysis of security techniques, risks, attacks and prevention of two widely used browser password management systems, the Internet. Explore and Firefox. The main object in this article covers two browsers IE 6 and 7, Firefox 1.5 and 2.0 and specifically the following:

1. ***Password saving technique*** : The meaning of protecting usernames and passwords on the local file system through encryption.
2. ***Types of attacks*** : Methods of destroying or overcoming protection.
3. ***Security mistakes*** : Users using passwords have no knowledge of risk.
4. ***Usability*** : Features to enhance or hinder the usability of security features.
5. ***Countermeasures and remedies*** : Actions necessary for users and organizations to reduce unnecessary risks.

Internet Explorer and Firefox have shared nearly 95% of all browsers. AutoComplete and Password Manager are features to store usernames, passwords and similar URLs of IE (from version 4) and Firefox (from version 0.7).

Each browser has its own features to support users by remembering different usernames and passwords as an authentication for websites. So when you go to a URL like http://www.gmail.com , where there are input fields, both Internet Explorer and Firefox will prompt the user if they want to save their username or password. When the user re-visits the site, the browser will automatically fill in those fields.

Although these features greatly simplify the user's responsibility, they also provide issues that need to be considered for security, which will be discussed in the following sections.

2. A case of password manager

The need for password managers is directly related to the difficulty of being able to remember a large number of usernames and passwords for specific websites. In fact, it should be noted that password managers can enhance the overall security because they have the right to allow greater entropy in the use of identifiers and passwords. So users can create different usernames instead of a username to make attackers more difficult to guess.

In terms of balance, users must rely on the application to perform its role (such as saving, handling safely and advanced capabilities to allow its existence). Password management is not a panacea, but they also work to promote technology, increase the barrier to attacks by improving the user interface to calculate lips. The regular school still needs evaluation.

Users as well as businesses need to be assured that password management systems must be used and properly implemented and knowledge of the risks involved. This article can be used as a basic knowledge for designing safer password managers by reviewing attacks, thereby building a solid solution to dealing with attacks. future.

3. First job

Using the same username and password in many websites will increase the likelihood of compromise, which is why an attacker only needs to discover a username and password to compromise all user resources. Using multiple passwords, memory techniques and the risk of reusing passwords are widely studied. In addition, extending to Firefox has also been studied to reduce the possibility of guessing passwords.

4. Password saving techniques

The locations and techniques for saving usernames and passwords are given below. This information is used as a study of the basic types of attacks used in Part 5.

4.1. Location saved

4.1.1. Internet Explorer 6 & 7

On Internet Explorer (versions 4 to 6) web format information AutoComplete is saved in the Registry at the following locations:

Encrypted usernames and passwords:

HKEY_CURRENT_USERSoftwareMicrosoftInternet ExplorerIntelliFormsSPW

Web addresses:

HKEY_LOCAL_MACHINESoftwareMicrosoftProtected Storage System Provider

Symmetric encryption keys:

HKEY_CURRENT_USERSoftwareMicrosoft Protected Storage System ProviderData

In Internet Explorer 7, AutoComplete information is also stored in the Registry but in another location.

Encrypted usernames and passwords:

HKEY_CURRENT_USERSoftwareMicrosoftInternet ExplorerIntelliFormsStorage2

Items in the registry are only created when the user performs a login (username and password) for a web page. SPW stands for *SavedPassWords* .

4.1.2. Firefox 1.5 and 2.0

In Firefox, Uniform Resource Locators, usernames and passwords are stored in a file *signons.txt* :

Encrypted usernames and passwords in the Windows system are stored in:

% userprofile% Application DataMozillaFirefox Profilesxxxxxxxx.defaultsignons.txt

When% userprofile%, is an environment variable in Windows, represents the path to the user's home directory.

Encrypted usernames and passwords in Linux systems running Firefox are saved in the following location:

~ / .mozilla / firefox / xxxxxxxx.default / signons.txt

The location of *xxxxxxxx* is selected randomly when Firefox is installed. The *signons.txt* file is created when any login for a website is saved. Following login with URLs inserted in the file. It is completely unrelated to the password manager if the page accesses using HTTP or HTTPs. URLs are not encrypted because they are used as a reference (lookup) for matching login. More specifically, when a browser password manager needs to auto-login for a specific page, there is the URL of the page referenced with the *signons.txt* file (if that URL exists), the corresponding username and password are filled in. go to the login page.

4.2. Access and storage techniques

4.2.1. Internet Explorer 6 & 7

1. **Architecture saves** : Registry
2. **Format** : Binary, and saved as a pair of hex values ??in a REG_BINARY data type.
3. **Encryption** : DES- three levels.
4. **Access** : Storage API protection (for Internet Explorer 4-6); Data protection API (for Internet Explorer 7)
5. **Requests for access** : User login.
6. **Temporary storage** : Symmetric keys are typed 0 into memory after use.

Internet Explorer 4-6 uses a storage protection provider (PStore) to save and access user information including usernames and passwords, passwords entered on web format in Internet Explorer. Pstore as defined by MSDN, is an application programming interface used to store information securely. In a recent announcement by Microsoft it was given:

". Protected Storage service, sooner or later will be taken care of as a secure way to store secrets. The most significant Windows application still uses P-store is Microsoft Internet Explorer, saving Auto information. - Complete includes names and passwords used for authentication based on forms. "

PStore data is encrypted with DES-three levels and stored in a binary architecture. Unencrypted data cannot be accessed directly through the registry. However, data access and security need to be tightened with users' Windows logon capabilities. When a user logs in, any program running under the user's content can increase access to unencrypted PStore data using the correct API calls. However, other Windows user accounts cannot access other PStore data.

The PStore is not only used exclusively on Internet Explorer but it is also used for other techniques of Microsoft products such as Outlook and MSN Explorer. These programs are also susceptible to vulnerabilities in security design. Some Spyware programs have learned how to undermine PStore security through its easily programmable API and increase unwarranted access.

Internet Explorer 7 uses the Data Protection Application Programming Interface (DPAPI), but these capabilities can still exist and be published to extended programs through API calls.

The password for AutoComplete in IE7 shown below is using the standard terminology:

EK - Encryption Key
RK - Record Key
CRC - Cyclical Redundancy Check (Check cyclical excess)
Hash - Secure Hash Algorithm (SHA) (Hash security algorithm)

Storage capacity :
EK: URL
RK: Hash (EncryptionKey)
C: CRC (Record Key)
V: {data} EK
Archive (C, V) is indexed by RK in the Registry, invalidating EK

Ability to recover :
EK: URL
RK: Hash (EK)
Look up RK in the Registry, see if it matches the encrypted data and data {V} EK or not

So the URL is required to restore the capabilities (data) as it ranks into the *EncryptionKey* (EK).

a. Concerns about access of Internet Explorer .

IE *AutoComplete* works under the assumption that a specific Windows user account is fully accessible to the password database. Therefore, if a user is not allowed to have reasonable access to the computer and the account is logged in or it is not a protected password, the attacker may abuse the account privileges and use Use passwords illegally. Reasonable access can be done directly on that computer or using remote access clients (VNC, remote workstation, .).

Thus, if there is no respect for the use of the device (such as the rooms being separated by keys, or passwords to protect the login of screen savers), anyone can use directly on the computer to access any website that allows password management. If you manage your computer well, an unreliable person who wants to increase access to anything (from a person's personal email to a bank account) will be difficult.

In addition, in cases where many people have the same logical user account (a poor security practice), problems skyrocket with unreasonable users. The remote control technique to increase access is given in the following section and is also valid for additional additions.

4.2.2. Firefox 0.7-1.5 and 2.0

Archived architecture : Text file format (signons.txt)
Format : ASCII, by using Base64 encoding (except URLs and fields)

URL (e.g. www.gmail.com)
Name field (in cleartext, for example: username, email, userid, .)
Base64 encoding for the above information
Name field (eg password, pass, .)
Base64 encoding for the above information
. (There may be multiple items per URL)

(Each URL entry lasts for a split line)

Encryption : DES three levels (CBC mode)
Access : Network security services API (NSS)
Requests for access : User logged in and master password (if set)
Related files : Certificates (signed Public Key) are stored in certN.db, and Private Key databases are stored in keyN.db and security modules stored in secmod.db

Note that the file locations are predefined in section 4.1.

Firefox uses the Network Security Services API to perform encryption. It relates to *Password Manager* Firefox to create the Public Key Cryptography Standard (PKCS) # 11 (defining the API for third party security modules including software and hardware). Use PKCS # 5 to encrypt passwords. Firefox also has an option of using alternate security modules for password managers, which is the national information processing standard (FIPS) 140-1. *The Master Password* is used in conjunction with a part in the keyN.db file that is often used to provide a *Master Key* . *Master Key is* then used to decrypt the username and password stored in *Password Manager* .

Although not easy to resolve, the NSS API has several important functions for Firefox or a related program to increase access to a password database. Password settings are held by (PK11_SetPasswordFunc), 64-base data decoding (NSSBase64_DecodeBuffer), decoding (PK11SDR_Decrypt) allows a program to relate to related usernames and passwords; This is indeed a simple problem. The actual code needs to launch NSS, declare variables, manage the cache, . Even so, the security of the entire system is based on the encryption power of Master Password (created by the user) and the ability to Access to key3.db file (including important parts) is saved in the user's profile.

FIPS 140-1 security module can be allowed by navigating the following placement:

Firefox 1.5 on Windows:

Tools | Options | Advanced | Security Devices | NSS Internal FIPS PKCS # 11

Firefox 2.0 on Windows:

Tools | Options | Advanced | Encryption | Security Devices | NSS Internal FIPS PKCS # 11

5. Attacks on the password manager

This section will look at several attacks to undermine password managers. The two attacks were discussed and then we would combine them into one part of the series.

One of the most common technologies for finding all the paths of a system is to use a tree diagram. The purpose of this diagram shown below in Figure 1 is the complete compromise of the password database.

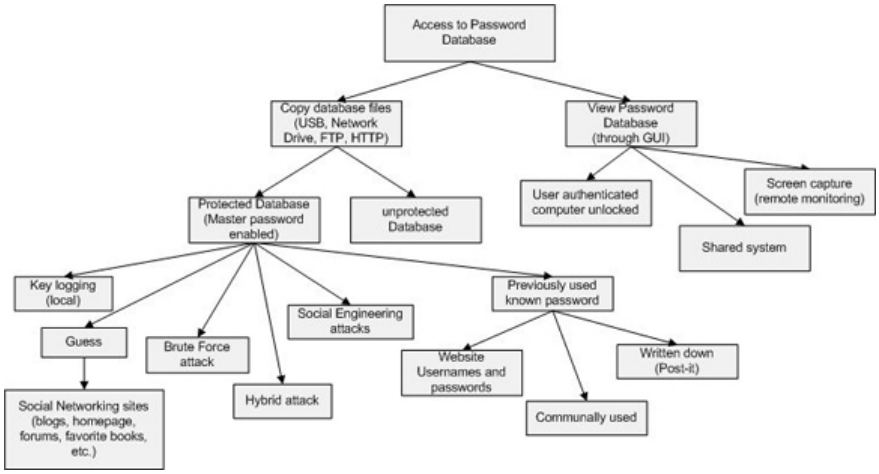

Figure 1 : Attack the tree with password manager in Firefox.

The result of increasing access to the database will allow an attacker to gain all the URLs, usernames, and passwords that were used to authenticate in the pages. The compromised password manager allows an attacker to access anything from e-mail to insurance, banking or even corporate intranet information. A small problem (not listed in the above tree) will compromise with the ability to log in for special pages.

A password management system developed for applications and user components. To compromise with a password database or a special login, just attack the weakest component of the system. In this case, the weakest link is always the user (no coding or additional software). Attacks are based on the interface between the user and the password manager or between the password manager and the browser.

5.1. The browser does not suffer from JavaScript attacks

Two JavaScript attacks will not be discussed in this section before concluding: a standard attack and an Ajax application.

5.1.1. Standard JavaScript attack

Assumption : An attacker has a valid user account

Attack results : An attacker has the ability to penetrate a saved page and 1) Increase access 2) Use JavaScript to detect username and password.

Other damage : use the exposed password to access the password manager or other applications and pages with the same password.

Using JavaScript can completely detect saved passwords on any page via the DOM. When someone visits a page that stores a username and password, the password is always hidden with *. That's what users see; but the browser saves the password in real *ASCII code* and submits it when the submission action is needed. The use of * is perfectly good in design to prevent out-of-interface.

To work around this preventive scope, an intelligent attacker can use both embedded JavaScript in an HTML page or run a script after loading the page, we assume the attacker has identified the username and password.

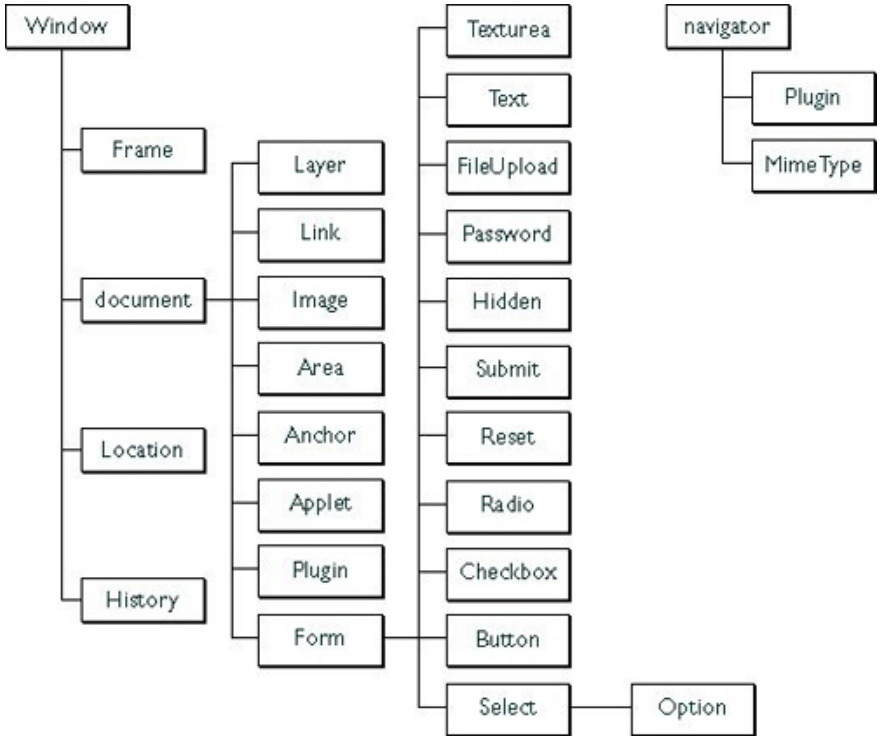

Figure 2 : JavaScript document object model

The JavaScript code is completely easy to access online. It can be embedded in HTML or run in a bookmarklet. A bookmarklet is a small JavaScript program that is saved as a URL and runs locally in the website after downloading.

Using programming logic, an attacker can iterate through all password-based elements (as shown in Figure 2) of the DOM; values ??corresponding to these password objects will be retrieved later, breaking the * is not possible. Anyone or a logical program with a Web client (Internet Explorer or Firefox) can 'click' on a link and find the password.

JavaScript code to retrieve password:

```
javascript :(
function () {
var s, F, j, f, i;
s = "";
F = document.forms;
for (j = 0; jf = F [j];
for (i = 0; i if (f [i] .type.toLowerCase () == "password")
s + = f [i] .value + "n";
}
}
if (s) alert ("Passwords in forms: nn" + s);
else alert ("No passwords in forms on this page.");}) ();
```

5.1.2. Getting the password using Ajax

Assumption : An attacker accesses a transparent web proxy or is configured for the client web.

Attack results : An attacker has the ability to insert, delete or change page content, allowing JavaScript to get the username and password on any page that has HTTP connections (even if SSL submit is used at the time). points later).

Other Damage : Allow to use the same login to access the computer system, other applications and pages using the same username and password.

In the problem presented in Figure 4 below, we have a user who is opening a web browser and wants to access his bank information on a remote server. The client requires the provider's main website (such as American Express). However, the information in the answer has been changed through a proxy server. Proxy servers are usually set to protect identity of IP addresses, filtering; they act as a communication medium between client and server.

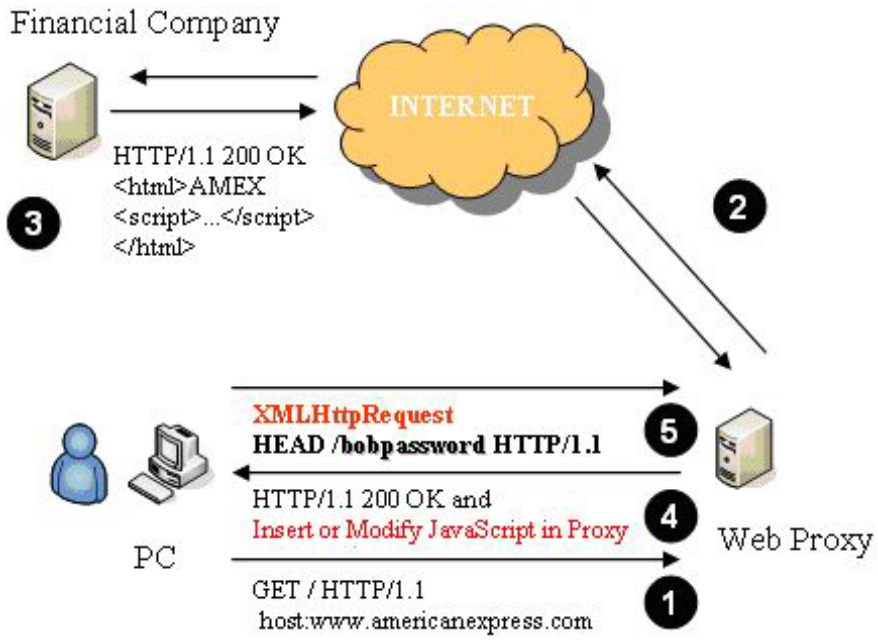

Figure 4 : Getting a password using Ajax

When an attacker controls the proxy, they can embed JavaScript to retrieve and send the username and password through a sync request to the server (XMLHttpRequest). Username and password can be obtained through JavaScript (password manager that automatically fills in login) or through timing technology to wait for a reasonable time (eg 5 seconds) to allow users to enter information believe and then JavaScript runs and sends the login data to the attacker. Figure 4 shows that the browser is requesting an XML file containing login information (bobpassword). The server will ignore the request because it is invalid but the attacker raises the login level at this point.

Identifying processes is very important, this process will authenticate users to the server and distribute malicious code that can send the username and password to the attacker. On a certain page, the login information is entered before the SSL connection is established. This is the main point of the attack, if there is an SSL connection established before logging in, the proxy server cannot see the encrypted traffic. However, some sites that use

SSL submit (such as Yahoo, AMEX, .) form the encrypted authentication connections after the original page is loaded, they will have vulnerabilities with these types of attacks.

You will notice a lot of important things when comparing the differences in password management features of two major browsers:

Characteristic

Internet Explorer 7

Firefox 2.0

Save username, password and URL
yes

yes

Password accessed via JavaScript
yes

yes

Access password via access software
yes

yes

Password protection (not tied to user accounts)
yes

Password Prompt when starting the session to save passwords
yes

Easily export username / password data
yes

Encode
yes

yes

Decryption
yes

Password Manager select "Show Passwords"
yes

With the last few items in the table above, note that it is necessary to pay attention to the issues of debate whether it is a trust characteristic or a security hole, though, there are sometimes no methods for password retrieval when forgotten.

Conclusion of part 1

Part 1 of this article provides a fundamental job for addressing password management issues, but actually addresses the first two points given at the beginning of the article, it has Technical explanation of password saving for both Internet Explorer and Firefox, and then presented two JavaScript attacks that can be used to circumvent browsers.

Please see part 2!