

比较Firefox和谷歌的记住密码插件的实现区别

一，记住密码插件功能

1，存储用户密码

- 记住密码插件通常在保护用户隐私安全前提下存储用户密码信息，具体而言，插件将在本地存储密码的hash值。
- 加密存储能够提升安全性，即使插件中密码信息被泄露，也无法还原出原始的密码信息。当用户使用密码时，插件计算hash值，与存储的值进行比对。

2，自动填充表单

- 记住密码插件可以自动填充登录表单和其他网页表单中的用户名、密码以及其他个人信息。这样可以方便用户快速登录和填写表单。

3，密码同步与备份

- 一些插件允许用户将保存的密码信息进行同步和备份，确保用户在不同设备上可以方便地访问并使用已保存的密码。

4，安全提示与警告

- 一些插件会提供额外的安全功能，如检测弱密码、重复使用密码等，并给予用户相应的警报和提示，以增强密码的安全性。

二，Firefox记住密码插件

1，Firefox插件实现简化

1,创建插件清单文件（manifest.json）：

- 在插件项目目录中创建一个名为 `manifest.json` 的文件。该清单文件描述了插件的名称、版本、权限等信息。
- 下面是一个示例的 `manifest.json` 文件内容：

- ```
- {
- "manifest_version": 2,
- "name": "My Password Manager",
- "version": "1.0",
- "description": "A password manager extension for Firefox.",
- "permissions": ["storage", "<all_urls>"],
- "browser_action": {
- "default_popup": "popup.html"
- },
- "icons": {
- "48": "icon.png"
- }
- }
```

## 2,创建弹出窗口页面

- 在插件项目目录中创建一个名为 `popup.html` 的文件。该文件定义了插件弹出窗口的外观和交互。
- 下面是一个示例的 `popup.html` 文件内容：

```
1. <html>
 <head>
 <title>My Password Manager</title>
 <script src="popup.js"></script>
 </head>
 <body>
 <h1>Password Manager</h1>
 <div id="passwords"></div>
 </body>
</html>
```

### 3,创建脚本文件

- 在插件项目目录中创建一个名为 `popup.js` 的文件。该文件包含与插件交互的 JavaScript 代码。
- 下面是一个示例的 `popup.js` 文件内容：

```
- document.addEventListener('DOMContentLoaded', function() {
-
- // 获取保存的密码
-
- chrome.storage.local.get('passwords', function(result) {
-
- var passwords = result.passwords || [];
-
- // 在弹出窗口中显示密码列表
-
- var passwordsDiv = document.getElementById('passwords');
-
- for (var i = 0; i < passwords.length; i++) {
-
- var password = passwords[i];
-
- var passwordElement = document.createElement('div');
-
- passwordElement.textContent = password;
-
- passwordsDiv.appendChild(passwordElement);
-
- }
-
- });
```

### 4,打包插件

- 将插件项目目录打包成zip格式，确保清单文件和其他文件位于压缩包的根目录。

### 5,在Firefox中加载插件

- 打开 Firefox 浏览器，访问 `about:debugging` 页面，在左侧导航栏选择 "This Firefox"，然后点击 "Load Temporary Add-on" 按钮，选择刚才打包的压缩包。

## 三，Google chrome记住密码插件

### 1, chrome密码插件实现简化

## 1,创建插件文件夹

- 在计算机上创建一个新的文件夹，并为插件选择一个唯一的名称。

## 2,创建清单文件

- 在插件文件夹中创建一个名为 `manifest.json` 的文件，并添加以下基本配置信息：

```
- {
- "manifest_version": 2,
- "name": "My Password Manager",
- "version": "1.0",
- "description": "A simple password manager plugin.",
- "permissions": ["tabs", "storage"],
- "content_scripts": [
- {
- "matches": ["<all_urls>"],
- "js": ["content.js"]
- }
-]
- }
```

### 3,创建内容脚本

- 在插件文件夹中创建一个名为 `content.js` 的 JavaScript 文件，其中包含与密码管理相关的逻辑。
- 以下是一个简单的示例来捕获和保存密码：

```
- // 监听登录表单的提交事件

- document.addEventListener('submit', function(event) {

- var form = event.target;

- // 获取用户名和密码字段的值

- var username = form.querySelector('input[name="username"]').value;

- var password = form.querySelector('input[name="password"]').value;

- // 使用 Chrome 的存储 API 将用户名和密码保存起来

- chrome.storage.local.set({ 'username': username, 'password': password },
function() {

- console.log('保存密码成功! ');

- });

- });
```

### 4,加载插件

- 打开 Chrome 浏览器并进入扩展程序管理界面 (`chrome://extensions`)。在顶部启用开发者模式，然后点击 "加载已解压的扩展程序" 按钮，选择创建的插件文件夹。

## 四，比较Firefox和谷歌的记住密码插件的实现区别

### 1, 安全性

## 1, 密码存储和加密

- Firefox: Firefox 使用一个称为 "Login Data" 的 SQLite 数据库来保存用户的密码。该数据库文件默认情况下位于用户的个人配置文件文件夹内。密码存储在本地计算机上, 并使用用户设置的主密码进行加密。只有在正确输入主密码后, 才能解密并访问保存的密码。
- Chrome: Chrome 使用一个类似的 SQLite 数据库来存储用户的密码, 称为 "Login Data"。同样, 密码被保存在本地计算机上, 并使用操作系统的登录密码作为加密密钥进行加密。只有在正确输入登录密码后, 才能解密并访问保存的密码。

## 2, 密码同步

- Firefox: Firefox 提供了一个称为 Firefox Sync 的功能, 允许用户跨设备同步其保存的密码和其他数据。Firefox Sync 使用加密同步协议来保护用户数据的传输和存储。
- Chrome: Chrome 也提供了类似的功能, 称为 Chrome 同步。通过 Chrome 同步, 用户可以在不同的设备上同步其保存的密码和其他浏览器数据。Chrome 同步使用加密协议来保护数据的传输和存储。

## 3, 安全更新与漏洞修复

- Firefox: Firefox 的开源性质使得安全研究人员能够审查其代码, 并报告发现的漏洞。Mozilla 组织会及时发布安全更新, 修复已确认的漏洞, 并向用户推送自动更新。
- Chrome: Chrome 由 Google 开发, 也会定期发布安全更新以修复漏洞。由于 Chrome 是封闭源代码的浏览器, 安全问题通常只有在发现后才会公开, 并由 Google 进行修复和发布更新。

## 4, 安全性能与扩展

- Firefox: Firefox 可以使用各种安全插件和扩展来增强其安全性, 例如密码生成器、多因素身份验证等。Mozilla 组织还专注于隐私保护, 通过阻止跟踪器和提供增强的隐私功能来提高用户的在线安全性。
- Chrome: Chrome 也支持各种安全扩展和功能, 例如密码生成器和广告拦截器。Google 在 Chrome 中实施了一些安全措施, 如安全浏览器警示, 帮助用户避免访问可能存在恶意软件或欺诈网站。

## 2, 存储位置

- Firefox使用称为"密码管理器"的内部工具来存储和管理保存的密码。这些密码以加密形式存储在Firefox的配置文件中。而Chrome使用Google账户进行同步, 保存的密码将与Google账户关联, 并且可以在不同设备间同步。

### 3, 主密码功能

- Firefox提供了一个主密码功能，可以为保存在浏览器中的所有密码设置额外的层级保护。这样，用户必须输入主密码才能访问保存的密码列表。Chrome没有类似的内置主密码功能。

### 4, 安全性控制

- Firefox提供了一些设置选项，例如启用或禁用自动填充密码，启用或禁用密码同步等。而Chrome在安全性控制方面提供了更多的选项，例如可选的账户验证、指纹识别等。