lclevy Merge pull request #17 from cookiengineer/patc... 167eabf · 3 years ago 28 Commits

mozilla_db	real first commit	11 years ago
LICENSE	Initial commit	11 years ago
firepwd.py	Bugfix for wrong path usage in key3.db...	3 years ago
mozilla_pbe.pdf	mostly cosmetic changes, thanks to An...	9 years ago
mozilla_pbe.svg	mostly cosmetic changes, thanks to An...	9 years ago
readme.md	typos	4 years ago
requirements.txt	updated markdown doc	4 years ago

README GPL-2.0 license

Firepwd.py, an open source tool to decrypt Mozilla protected passwords

18apr2020

Introduction

This educational tool was written to illustrate how Mozilla passwords (Firefox, Thunderbird) are protected using contents of files key4.db (or key3.db), logins.json (or signons.sqlite).

NSS library is NOT used. Only python is used (PyCryptodome, pyasn1)

This code is released under GPL license.

Now part of LaZagne project: https://github.com/AlessandroZ/LaZagne

You can also read the related article, in french: http://connect.ed-diamond.com/MISC/MISC-069/Protection-des-mots-de-passe-par-Firefox-et-Thunderbird-analyse-par-la-pratique

or this poster for the password crypto of key3.db and signons.sqlite.

Versions supported

- Firefox <32 (key3.db, signons.sqlite)
- Firefox >=32 (key3.db, logins.json)
- Firefox >=58.0.2 (key4.db, logins.json)
- Firefox >=75.0 (sha1 pbkdf2 sha256 aes256 cbc used by key4.db, logins.json)
- at least Thunderbird 68.7.0, likely other versions

key3.db is read directly, the 3rd party bsddb python module is NOT needed.

About

firepwd.py, an open source tool to decrypt Mozilla protected passwords

Readme
GPL-2.0 license
Activity
597 stars
29 watching
113 forks
Report repository

Releases

No releases published

Packages

No packages published

Contributors 5

Languages

Python 100.0%

Usage

By default, firepwd.py processes key3.db (or key4.db) and signons.sqlite (logins.json) files in current directory, but an alternative directory can be provided using the -d option. Do not forget the '/' at the end.

If a master password has been set, provide it using the -p option.

Valid verbose levels (-v) are from 0 (default) to 2.

```
$ python firepwd.py -h
Usage: firepwd.py [options]

Options:
  -h, --help            show this help message and exit
  -v VERBOSE, --verbose=VERBOSE
                        verbose level
  -p MASTERPASSWORD, --password=MASTERPASSWORD
                        masterPassword
  -d DIRECTORY, --dir=DIRECTORY
                        directory

$ python firepwd.py -d /c/Users/lclevy/AppData/Roaming/Mozilla/Firef
no stored passwords

$ python firepwd.py -p 'MISC*' -d mozilla_db/
 SEQUENCE {
   SEQUENCE {
     OBJECTIDENTIFIER 1.2.840.113549.1.12.5.1.3
     SEQUENCE {
       OCTETSTRING a8db682ac51cfad8c06664fe9deb5283073b33ee
       INTEGER 01
     }
   }
   OCTETSTRING 72d5636049d4af9eeadaf7eb0dc1710a62d5362fe4086dcc0495e
 }
decrypting privKeyData
 SEQUENCE {
   INTEGER 00
   SEQUENCE {
     OBJECTIDENTIFIER 1.2.840.113549.1.1.1
     NULL 0
   }
   OCTETSTRING 3042020100021100f8000000000000000000000000000001020100
 }
decoding 3042020100021100f8000000000000000000000000000001020100021811
 SEQUENCE {
   INTEGER 00
   INTEGER 00f8000000000000000000000000000001
   INTEGER 00
   INTEGER 13c1e53d51a1e60bc79419f7d59107ef97976d075832a45b
   INTEGER 00
   INTEGER 00
   INTEGER 00
   INTEGER 00
   INTEGER 15
 }
decrypting login/password pairs
http://challenge01.root-me.org: 'login\x03\x03\x03' , 'password\x08\

$ python firepwd.py -d /c/Users/laurent/AppData/Roaming/Thunderbird/
 SEQUENCE {
   SEQUENCE {
     OBJECTIDENTIFIER 1.2.840.113549.1.12.5.1.3
     SEQUENCE {
       OCTETSTRING 10540ef85fb7e198d41884c8c9c90cf3bc065482
       INTEGER 01
     }
   }
   OCTETSTRING 082fe34f23eae209334d53be2c85ea62d0242a722d452da5b0f27
 }
decrypting privKeyData
 SEQUENCE {
```

```
    INTEGER 00
    SEQUENCE {
      OBJECTIDENTIFIER 1.2.840.113549.1.1.1
      NULL 0
    }
    OCTETSTRING 3042020100021100f8000000000000000000000000000000102010
  }
decoding 3042020100021100f8000000000000000000000000000000102010002187
  SEQUENCE {
    INTEGER 00
    INTEGER 00f8000000000000000000000000000001
    INTEGER 00
    INTEGER 75a873cdb39783ecf1fedcea3d010dd9732a01a8b30451e9
    INTEGER 00
    INTEGER 00
    INTEGER 00
    INTEGER 00
    INTEGER 15
  }
decrypting login/password pairs
[censored]


$ python firepwd.py -d /c/Users/laurent/AppData/Roaming/Mozilla/Fire
  SEQUENCE {
    SEQUENCE {
      OBJECTIDENTIFIER 1.2.840.113549.1.12.5.1.3
      SEQUENCE {
        OCTETSTRING c6581e1fbdb50b4265ab11f54861fdbb62cb4abd
        INTEGER 01
      }
    }
    OCTETSTRING cecb819cb612dccfc2265121aa38ed5d4b7cfc6f06f92f4fb4830
  }
decrypting privKeyData
[...]

>python firepwd.py -v 2 -p MISC* -d ff50\
globalSalt: b'5ed0adce15d896b84115f530be4e259f72beda91'
  SEQUENCE {
    SEQUENCE {
      OBJECTIDENTIFIER 1.2.840.113549.1.5.13 pkcs5 pbes2
      SEQUENCE {
        SEQUENCE {
          OBJECTIDENTIFIER 1.2.840.113549.1.5.12 pkcs5 PBKDF2
          SEQUENCE {
            OCTETSTRING b'f92dde91809b8b00c6607b73f3d0321c80f930aa13f
            INTEGER b'01'
            INTEGER b'20'
            SEQUENCE {
              OBJECTIDENTIFIER 1.2.840.113549.2.9 hmacWithSHA256
            }
          }
        }
        SEQUENCE {
          OBJECTIDENTIFIER 2.16.840.1.101.3.4.1.42 aes256-CBC
          OCTETSTRING b'd7f6eef452a0becb5227af2e175c'
        }
      }
    }
    OCTETSTRING b'9ef5288ba19326df7188f1f0d1811c2a'
  }
clearText b'70617373776f72642d636865636b0202'
password check? True
  SEQUENCE {
    SEQUENCE {
      OBJECTIDENTIFIER 1.2.840.113549.1.5.13 pkcs5 pbes2
      SEQUENCE {
        SEQUENCE {
          OBJECTIDENTIFIER 1.2.840.113549.1.5.12 pkcs5 PBKDF2
          SEQUENCE {
            OCTETSTRING b'86535fdbbc242465d6e8477094b93221c9cc45bb363
            INTEGER b'01'
            INTEGER b'20'
            SEQUENCE {
```

```
              OBJECTIDENTIFIER 1.2.840.113549.2.9 hmacWithSHA256
            }
          }
        }
        SEQUENCE {
          OBJECTIDENTIFIER 2.16.840.1.101.3.4.1.42 aes256-CBC
          OCTETSTRING b'4de278f3bc4cf8e503ce0b8672ec'
        }
      }
    }
    OCTETSTRING b'62093ca8bb60c0416b5e7bee18402b99c21e780985ff75737fb8
  }
clearText b'7f914a642a4552b0e0c7a87061fe5d9437a41968c4a7d35e08080808(
decrypting login/password pairs
[...]
```

Installation

```
pip install -r requirements.txt
```

Tested with python 3.7.3, PyCryptodome 3.9.0 and pyasn 0.4.8

Modules required:

- pyasn1, https://pypi.python.org/pypi/pyasn1/, for ASN1 decoding
- PyCryptodome, https://www.pycryptodome.org/en/latest/, for 3DES and AES decryption

Reference documents

- Into the Black Box: A Case Study in Obtaining Visibility into Commercial Software, D. Plakosh, S. Hissam, K. Wallnau, March 1999, Carnegie Mellon University : http://www.sei.cmu.edu/library/abstracts/reports/99tn010.cfm
- Dr. Stephen Henson, August 4th 1999 : http://arc.info/?l=openssl-