

Forum:[Hack Forums](#)

Thread: [Extract the contents of all types of compressed files using the password memo](#)

usa1010 - November 25, 2023, 1:23 pm

I want a program in the Python language, to extract the contents in compressed files of all kinds using a password memo. The program interface contains four buttons: the first to select compressed files. The second button to choose the password memo. The third button to choose the contents extraction file. The fourth button to start the process. How the program works: The program It tests the passwords in the note on the compressed files. When the password is correct and the contents are extracted, they are transferred to the extraction file and the program also displays the word Thank you. The process has completed successfully.

skull - November 25, 2023, 1:33 pm

Hope this helps

<https://www.datacamp.com/tutorial/zip-file>

fbirds5230 - November 25, 2023, 7:17 pm

(Nov 25, 2023 02:23 PM)usa1010 Wrote: [\[CITE\]](#) I want a program in the Python language, to extract the contents in compressed files of all kinds using a password memo. The program interface contains four buttons: the first to select compressed files. The second button to choose the password memo. The third button to choose the contents extraction file. The fourth button to start the process. How the program works: The program It tests the passwords in the note on the compressed files. When the password is correct and the contents are extracted, they are transferred to the extraction file and the program also displays the word Thank you. The process has completed successfully.

```
Code import tkinter as tk
import zipfile
import os
```

```
def extract_contents():
    zip_file = zip_file_entry.get()
    password_file = password_file_entry.get()
    extract_file = extract_file_entry.get()

    if not (zip_file and password_file and extract_file):
        status_label.config(text="Please fill in all fields.")
        return

    try:
        with open(password_file, 'r') as f:
            passwords = f.readlines()

        for password in passwords:
            password = password.strip()

            try:
                with zipfile.ZipFile(zip_file, 'r') as zip_ref:
                    zip_ref.extractall(path=extract_file, pwd=password.encode())
```

```
        status_label.config(text="Thank you. Process completed successfully.")
    return
except Exception as e:
    pass
```

```
        status_label.config(text="Extraction failed. Incorrect password or other error.")
except FileNotFoundError:
    status_label.config(text="File not found.")
```

```
def browse_zip():
    zip_file = tk.filedialog.askopenfilename()
    zip_file_entry.delete(0, tk.END)
    zip_file_entry.insert(0, zip_file)
```

```
def browse_password():
    password_file = tk.filedialog.askopenfilename()
    password_file_entry.delete(0, tk.END)
    password_file_entry.insert(0, password_file)
```

```
def browse_extract():
    extract_file = tk.filedialog.askdirectory()
    extract_file_entry.delete(0, tk.END)
    extract_file_entry.insert(0, extract_file)
```

```
root = tk.Tk()
root.title("Compressed File Extractor")
```

```
zip_label = tk.Label(root, text="Select Compressed File:")
zip_label.pack()
```

```
zip_file_entry = tk.Entry(root, width=50)
zip_file_entry.pack()
```

```
zip_button = tk.Button(root, text="Browse", command=browse_zip)
zip_button.pack()
```

```
password_label = tk.Label(root, text="Select Password Memo:")
password_label.pack()
```

```
password_file_entry = tk.Entry(root, width=50)
password_file_entry.pack()
```

```
password_button = tk.Button(root, text="Browse", command=browse_password)
password_button.pack()
```

```
extract_label = tk.Label(root, text="Select Extraction File:")
extract_label.pack()
```

```
extract_file_entry = tk.Entry(root, width=50)
extract_file_entry.pack()
```

```
extract_button = tk.Button(root, text="Browse", command=browse_extract)
extract_button.pack()
```

```
extract_button = tk.Button(root, text="Start Extraction", command=extract_contents)
extract_button.pack()

status_label = tk.Label(root, text="")
status_label.pack()

root.mainloop()
```

Hackers - November 26, 2023, 12:20 pm

(Nov 25, 2023 02:23 PM)usa1010 Wrote: [\[CITE\]](#) I want a program in the Python language, to extract the contents in compressed files of all kinds using a password memo. The program interface contains four buttons: the first to select compressed files. The second button to choose the password memo. The third button to choose the contents extraction file. The fourth button to start the process. How the program works: The program It tests the passwords in the note on the compressed files. When the password is correct and the contents are extracted, they are transferred to the extraction file and the program also displays the word Thank you. The process has completed successfully.

So basically you want a bruteforcer for password protected zips? Is this for Windows or another oS?

usa1010 - November 26, 2023, 12:33 pm

(Nov 26, 2023 01:20 PM)nodes Wrote: [\[CITE\]](#) So basically you want a bruteforcer for password protected zips? Is this for Windows or another oS?

For Windows, I have the passwords for the compressed files, but I want to extract the contents automatically using a program

(Nov 25, 2023 08:17 PM)fbirds5230 Wrote: [\[CITE\]](#) Code import tkinter as tk
import zipfile
import os

```
def extract_contents():
    zip_file = zip_file_entry.get()
    password_file = password_file_entry.get()
    extract_file = extract_file_entry.get()

    if not (zip_file and password_file and extract_file):
        status_label.config(text="Please fill in all fields.")
        return

    try:
        with open(password_file, 'r') as f:
            passwords = f.readlines()

        for password in passwords:
            password = password.strip()

            try:
                with zipfile.ZipFile(zip_file, 'r') as zip_ref:
                    zip_ref.extractall(path=extract_file, pwd=password.encode())
                    status_label.config(text="Thank you. Process completed successfully.")
                    return
            except Exception as e:
```

pass

```
status_label.config(text="Extraction failed. Incorrect password or other error.")  
except FileNotFoundError:  
    status_label.config(text="File not found.")
```

```
def browse_zip():  
    zip_file = tk.filedialog.askopenfilename()  
    zip_file_entry.delete(0, tk.END)  
    zip_file_entry.insert(0, zip_file)
```

```
def browse_password():  
    password_file = tk.filedialog.askopenfilename()  
    password_file_entry.delete(0, tk.END)  
    password_file_entry.insert(0, password_file)
```

```
def browse_extract():  
    extract_file = tk.filedialog.askdirectory()  
    extract_file_entry.delete(0, tk.END)  
    extract_file_entry.insert(0, extract_file)
```

```
root = tk.Tk()  
root.title("Compressed File Extractor")
```

```
zip_label = tk.Label(root, text="Select Compressed File:")  
zip_label.pack()
```

```
zip_file_entry = tk.Entry(root, width=50)  
zip_file_entry.pack()
```

```
zip_button = tk.Button(root, text="Browse", command=browse_zip)  
zip_button.pack()
```

```
password_label = tk.Label(root, text="Select Password Memo:")  
password_label.pack()
```

```
password_file_entry = tk.Entry(root, width=50)  
password_file_entry.pack()
```

```
password_button = tk.Button(root, text="Browse", command=browse_password)  
password_button.pack()
```

```
extract_label = tk.Label(root, text="Select Extraction File:")  
extract_label.pack()
```

```
extract_file_entry = tk.Entry(root, width=50)  
extract_file_entry.pack()
```

```
extract_button = tk.Button(root, text="Browse", command=browse_extract)  
extract_button.pack()
```

```
extract_button = tk.Button(root, text="Start Extraction", command=extract_contents)  
extract_button.pack()
```

```
status_label = tk.Label(root, text="")
status_label.pack()
```

```
root.mainloop()
```

There is a problem in the code, brother

Hackers - November 26, 2023, 12:47 pm

(Nov 26, 2023 01:33 PM)usa1010 Wrote: [\[CITE\]](#) For Windows, I have the passwords for the compressed files, but I want to extract the contents automatically using a program

There is a problem in the code, brother

Ah gotcha, gotcha

Can you share the error code? What you're asking for is fairly complicated and will require you to install dependencies on your machine as well as test it out and debug. I mean, it's not that complicated, but I don't know if anyone is going to hand you a completely working script here for free, or they'll use GPT or similar and may not debug it for you

If you share any code and errors, I can help out here, just quote me in your reply so I'm notified