

Forum: [Hack Forums](#)

Thread: [\[source\]](#) Simple Batch File Obfuscator

Robert Rouge - March 23, 2025, 6:13 pm

hello hachforums,

made this line by line batch obfuscator specifically for future projects as learn batch. it is written in C# as a console app, just build the exe navigate to the folder is stored in via command line and then execute command (batchObfuscater.exe [path to batch file])

```
Code using System;  
using System.Collections.Generic;  
using System.IO;
```

```

class BatchFileObfuscator
{
    static void Main(string[] args)
    {
        // Check for correct number of arguments
        if (args.Length != 1)
        {
            Console.WriteLine("Usage: BatchFileObfuscator <inputFilePath>");
            return;
        }

        string inputFilePath = args[0];

        try
        {
            // Generate the output file path by prepending "o-" to the input file name
            string directory = Path.GetDirectoryName(inputFilePath);
            string fileNameWithoutExtension = Path.GetFileNameWithoutExtension(inputFilePath);
            string fileExtension = Path.GetExtension(inputFilePath);
            string outputFilePath = Path.Combine(directory, "o-" + fileNameWithoutExtension +
fileExtension);

            // Read all lines from the input batch file
            string[] lines = File.ReadAllLines(inputFilePath);
            // Create a new list to hold the modified lines
            var modifiedLines = new List<string>();

            // Add the lines to the modified list
            modifiedLines.Add("");
            modifiedLines.Add("");
            modifiedLines.Add("");
            modifiedLines.Add("");
            modifiedLines.Add("");
            modifiedLines.Add("");

```

```

modifiedLines.Add("");
modifiedLines.Add("@echo off");
modifiedLines.Add("rem *****");
modifiedLines.Add("rem *****CREATED BY ROUGE
SOFTWARE*****");
modifiedLines.Add("rem *****");
modifiedLines.Add("set Rouge=echo && set Software=msg && set flamingW0rm=pause");
modifiedLines.Add("");

```

```

// Dictionary to keep track of label renaming
var labelMap = new Dictionary<string, string>();
int labelCounter = 1;

```

```

// Process the lines and replace 'echo' with '%Rouge%'
foreach (var line in lines)
{
    // Skip the original @echo off line
    if (line.Trim() != "@echo off")
    {
        // Check for goto statements
        if (line.TrimStart().StartsWith("goto", StringComparison.OrdinalIgnoreCase))
        {
            // Extract the label name
            var labelName = line.Substring(4).Trim();
            if (labelMap.ContainsKey(labelName))
            {
                // Replace with the new label name
                modifiedLines.Add(line.Replace(labelName, labelMap[labelName]));
            }
            else
            {
                // Create a new label name
                string newLabelName = "RougeSoftware" + labelCounter++;
                labelMap[labelName] = newLabelName;
                modifiedLines.Add(line.Replace(labelName, newLabelName));
            }
        }
        else if (line.TrimStart().StartsWith(":"))
        {
            // Rename the label
            string originalLabel = line.Trim();
            string labelName = originalLabel.Substring(1); // Remove the colon
            if (labelMap.ContainsKey(labelName))
            {
                modifiedLines.Add(":" + labelMap[labelName]);
            }
            else
            {
                // Keep the original label if it wasn't referenced by a goto
                modifiedLines.Add(originalLabel);
            }
        }
    }
}

```

```

        else
        {
            // Replace 'echo' with '%Rouge%'
            modifiedLines.Add(ReplaceEchoWithRouge(line));
        }
    }
}

// Write the modified content to the output file
File.WriteAllLines(outputFilePath, modifiedLines);
Console.WriteLine("Replacement complete. Output written to: " + outputFilePath);
}
catch (Exception ex)
{
    Console.WriteLine("An error occurred: " + ex.Message);
}
}

static string ReplaceEchoWithRouge(string line)
{
    // Replace 'echo' with '%Rouge%' (case-sensitive)
    line = line.Replace("echo", "%Rouge%");
    line = line.Replace("msg", "%Software%");
    line = line.Replace("pause", "%flamingW0rm%");
    line = line.Replace("@echo off", "");

    return line;
}

```

} make string encrypter to based on: <https://www.instructables.com/Batch-text-encrypter/>

here is code if wanna try to go crazy encrypt strings too

Code @echo off

setlocal ENABLEEXTENSIONS ENABLEDELAYEDEXPANSION

title Encrypt/Decrypt

color 0a

:: Character mappings for encryption

```

(set CHAR[a]=UDFM45) & (set CHAR[b]=H21DGF) & (set CHAR[c]=FDH56D) & (set
CHAR[d]=FGS546) & (set CHAR[e]=JUK4JH)
(set CHAR[f]=ERG54S) & (set CHAR[g]=T5H4FD) & (set CHAR[h]=RG641G) & (set
CHAR[i]=RG4F4D) & (set CHAR[j]=RT56F6)
(set CHAR[k]=VCBC3B) & (set CHAR[l]=F8G9GF) & (set CHAR[m]=FD4CJS) & (set
CHAR[n]=G423FG) & (set CHAR[o]=F45GC2)
(set CHAR[p]=TH5DF5) & (set CHAR[q]=CV4F6R) & (set CHAR[r]=XF64TS) & (set
CHAR[s]=X78DGT) & (set CHAR[t]=TH74SJ)
(set CHAR[u]=BCX6DF) & (set CHAR[v]=FG65SD) & (set CHAR[w]=4KL45D) & (set
CHAR[x]=GFH3F2) & (set CHAR[y]=GH56GF)
(set CHAR[z]=45T1FG) & (set CHAR[1]=D4G23D) & (set CHAR[2]=GB56FG) & (set
CHAR[3]=SF45GF) & (set CHAR[4]=P4FF12)
(set CHAR[5]=F6DFG1) & (set CHAR[6]=56FG4G) & (set CHAR[7]=USGFDG) & (set
CHAR[8]=FKHFDG) & (set CHAR[9]=IFGJH6)
(set CHAR[0]=87H8G7) & (set CHAR[@]=G25GHF) & (set CHAR[#]=45FGFH) & (set

```

```
CHAR[$]=75FG45) & (set CHAR[*]=54GDH5)
(set CHAR[(]=45F465) & (set CHAR[.]=HG56FG) & (set CHAR[,]=DF56H4) & (set
CHAR[-]=F5JHFH) & (set CHAR[_]=SGF4HF)
(set CHAR[\]=45GH45) & (set CHAR[/]=56H45G)
```

:: Character mappings for decryption

```
(set CHAR_DECRYPT[UDFM45]=a) & (set CHAR_DECRYPT[H21DGF]=b) & (set
CHAR_DECRYPT[FDH56D]=c) & (set CHAR_DECRYPT[FGS546]=d) & (set
CHAR_DECRYPT[JUK4JH]=e)
(set CHAR_DECRYPT[ERG54S]=f) & (set CHAR_DECRYPT[T5H4FD]=g) & (set
CHAR_DECRYPT[RG641G]=h) & (set CHAR_DECRYPT[RG4F4D]=i) & (set
CHAR_DECRYPT[RT56F6]=j)
(set CHAR_DECRYPT[VCBC3B]=k) & (set CHAR_DECRYPT[F8G9GF]=l) & (set
CHAR_DECRYPT[FD4CJS]=m) & (set CHAR_DECRYPT[G423FG]=n) & (set
CHAR_DECRYPT[F45GC2]=o)
(set CHAR_DECRYPT[TH5DF5]=p) & (set CHAR_DECRYPT[CV4F6R]=q) & (set
CHAR_DECRYPT[XF64TS]=r) & (set CHAR_DECRYPT[X78DGT]=s) & (set
CHAR_DECRYPT[TH74SJ]=t)
(set CHAR_DECRYPT[BCX6DF]=u) & (set CHAR_DECRYPT[FG65SD]=v) & (set
CHAR_DECRYPT[4KL45D]=w) & (set CHAR_DECRYPT[GFH3F2]=x) & (set
CHAR_DECRYPT[GH56GF]=y)
(set CHAR_DECRYPT[45T1FG]=z) & (set CHAR_DECRYPT[D4G23D]=1) & (set
CHAR_DECRYPT[GB56FG]=2) & (set CHAR_DECRYPT[SF45GF]=3) & (set
CHAR_DECRYPT[P4FF12]=4)
(set CHAR_DECRYPT[F6DFG1]=5) & (set CHAR_DECRYPT[56FG4G]=6) & (set
CHAR_DECRYPT[USGFDG]=7) & (set CHAR_DECRYPT[FKHFDG]=8) & (set
CHAR_DECRYPT[IFGJH6]=9)
(set CHAR_DECRYPT[87H8G7]=0) & (set CHAR_DECRYPT[G25GHF]=@) & (set
CHAR_DECRYPT[45FGFH]=#) & (set CHAR_DECRYPT[75FG45]=$) & (set
CHAR_DECRYPT[54GDH5]=*)
(set CHAR_DECRYPT[45F465]=() & (set CHAR_DECRYPT[HG56FG]=.) & (set
CHAR_DECRYPT[DF56H4]=,) & (set CHAR_DECRYPT[F5JHFH]=-) & (set
CHAR_DECRYPT[SGF4HF]= )
(set CHAR_DECRYPT[45GH45]=\ ) & (set CHAR_DECRYPT[56H45G]=/)
```

:mainmenu

cls

echo Choose an option:

echo 1. Encrypt

echo 2. Decrypt

echo 3. Exit

set /p choice=Enter your choice (1-3):

if "%choice%"=="1" goto encrypt

if "%choice%"=="2" goto decrypt

if "%choice%"=="3" exit

goto mainmenu

:encrypt

cls

echo Enter a string to encrypt:

set /p Encrypt=Input:

set Encrypt2=%Encrypt%

```

set "EncryptOut="
:encrypt_loop
set char=%Encrypt2:~0,1%
set Encrypt2=%Encrypt2:~1%
if defined CHAR[%char%] (
    set EncryptOut=%EncryptOut%!CHAR[%char%]!
) else (
    set EncryptOut=%EncryptOut%%char% :: If character not found, keep it as is
)
if not "%Encrypt2%"==" " goto encrypt_loop
echo Input string: %Encrypt%
echo Output string: %EncryptOut%
echo.
pause
goto mainmenu

:decrypt
cls
echo Enter a string to decrypt:
set /p Decrypt=Input
set Decrypt2=%Decrypt%
set "DecryptOut="
:decrypt_loop
set char=%Decrypt2:~0,6%
set Decrypt2=%Decrypt2:~6%
set DecryptOut=%DecryptOut%!CHAR_DECRYPT[%char%]!
if not "%Decrypt2%"==" " goto decrypt_loop
echo Input string: %Decrypt%
echo.
echo Output string: %DecryptOut%
set string=%DecryptOut%
set temp_str=%string%
set str_len=0
:lengthloop
if defined temp_str (
    set temp_str=%temp_str:~1%
    set /A str_len += 1
    goto lengthloop
)
echo.
echo Output string is %str_len% characters long!
echo.
pause
goto mainmenu

```

cote - March 26, 2025, 10:50 pm

Interesting. Thank you for the contribution, I like the obfuscation style!