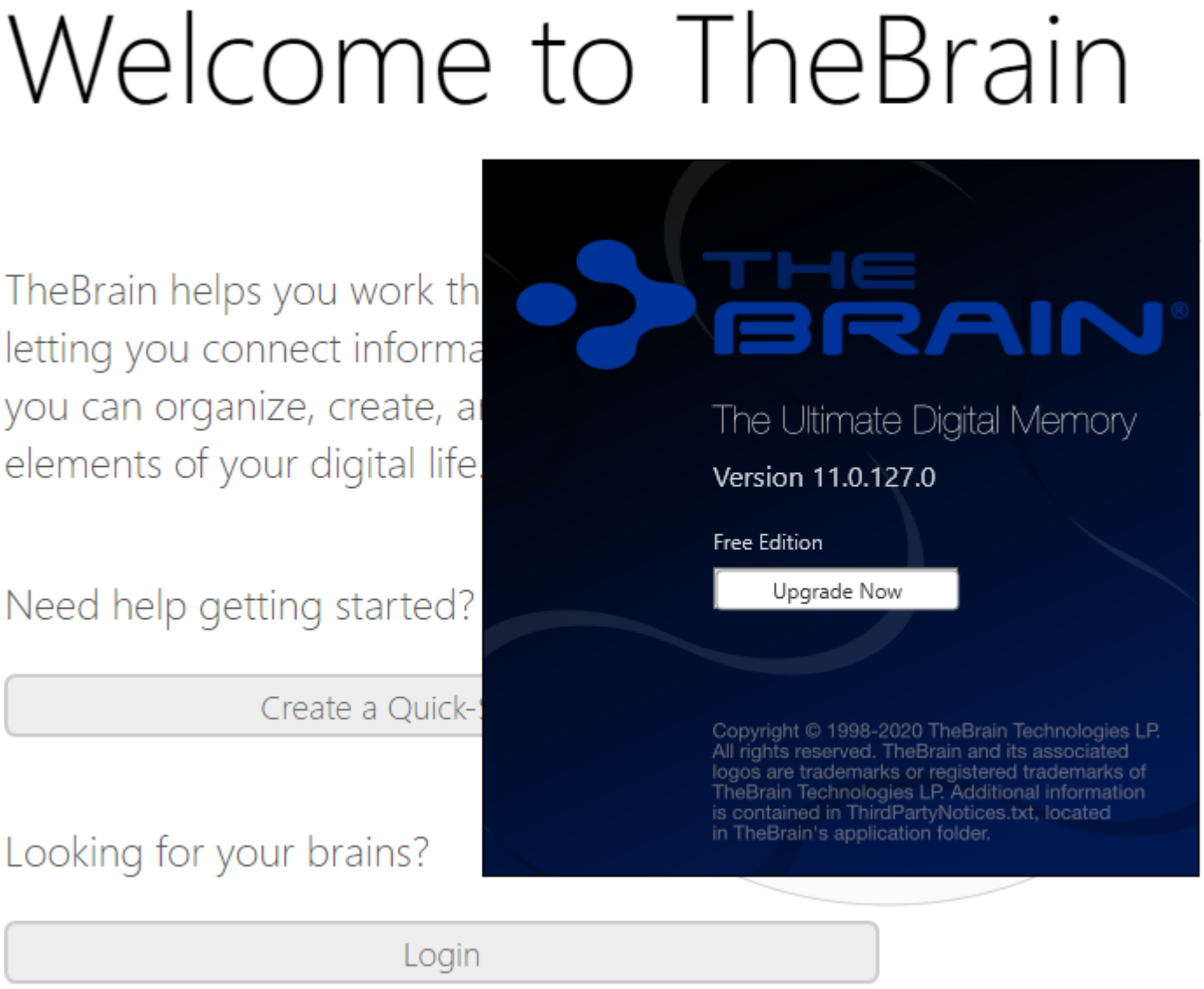
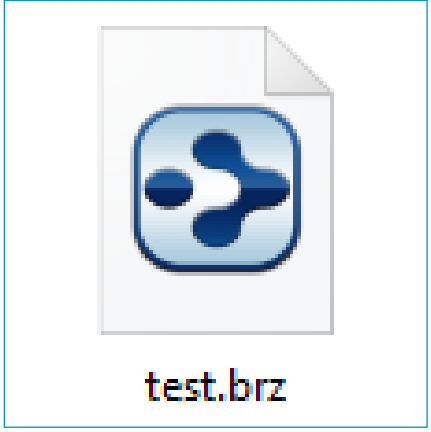
The target: a mind mapping software developed since 1997 and still very updated

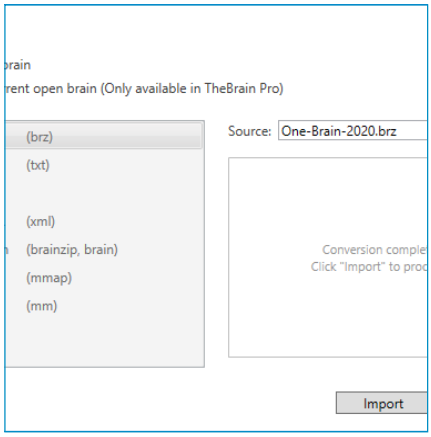


Found in July 2012 in version 7.0.4.4
Current version 11.0.127.0 is still affected
Reported on 5 August 2020, still not fixed by vendor
Found and reported by Luigi Auriemma



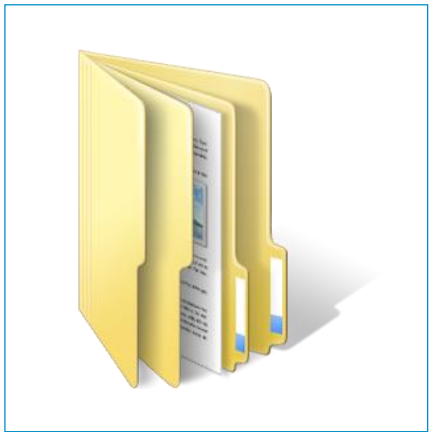
BRZ is a registered file type

BRZ files are opened by TheBrain when clicked

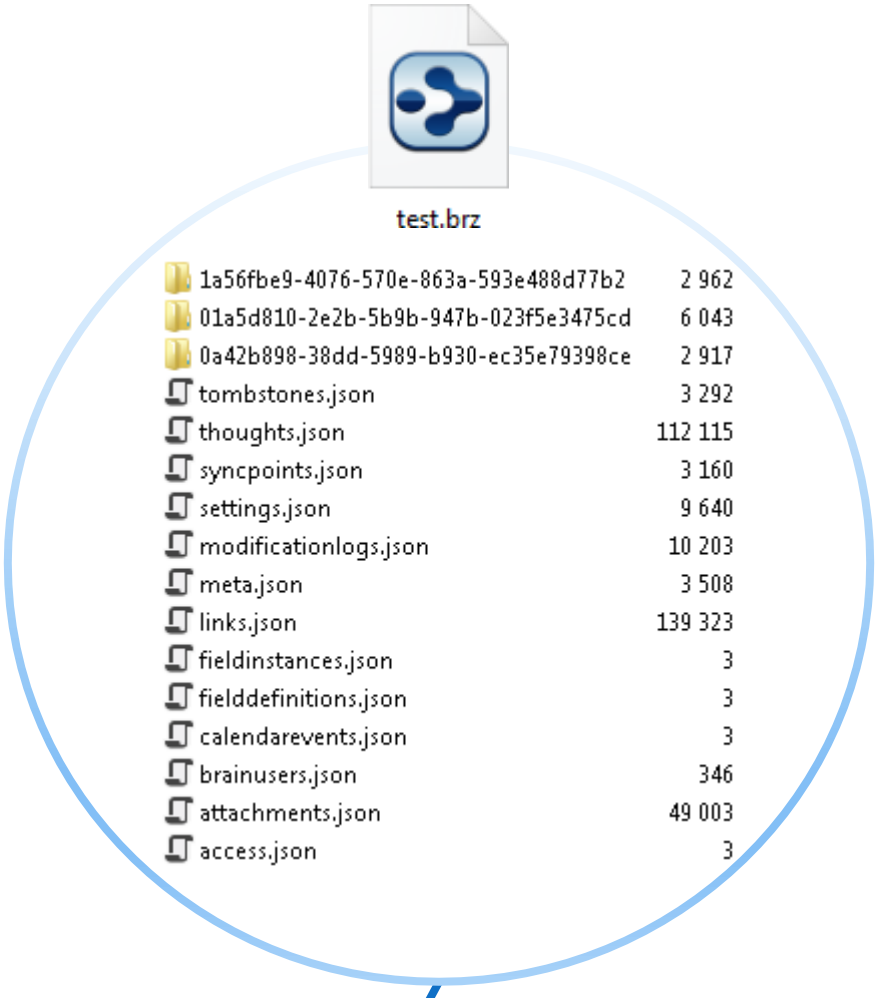


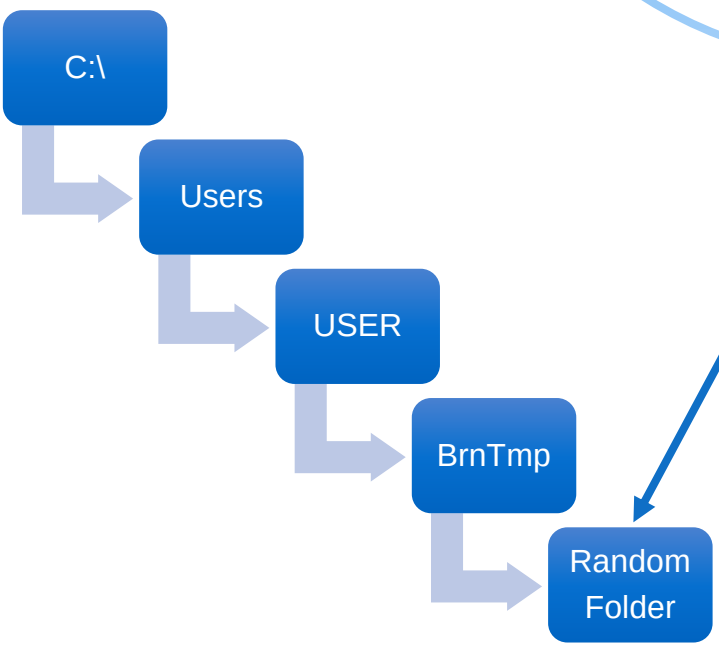
BRZ is just a ZIP file

ZIP files contain files and folders

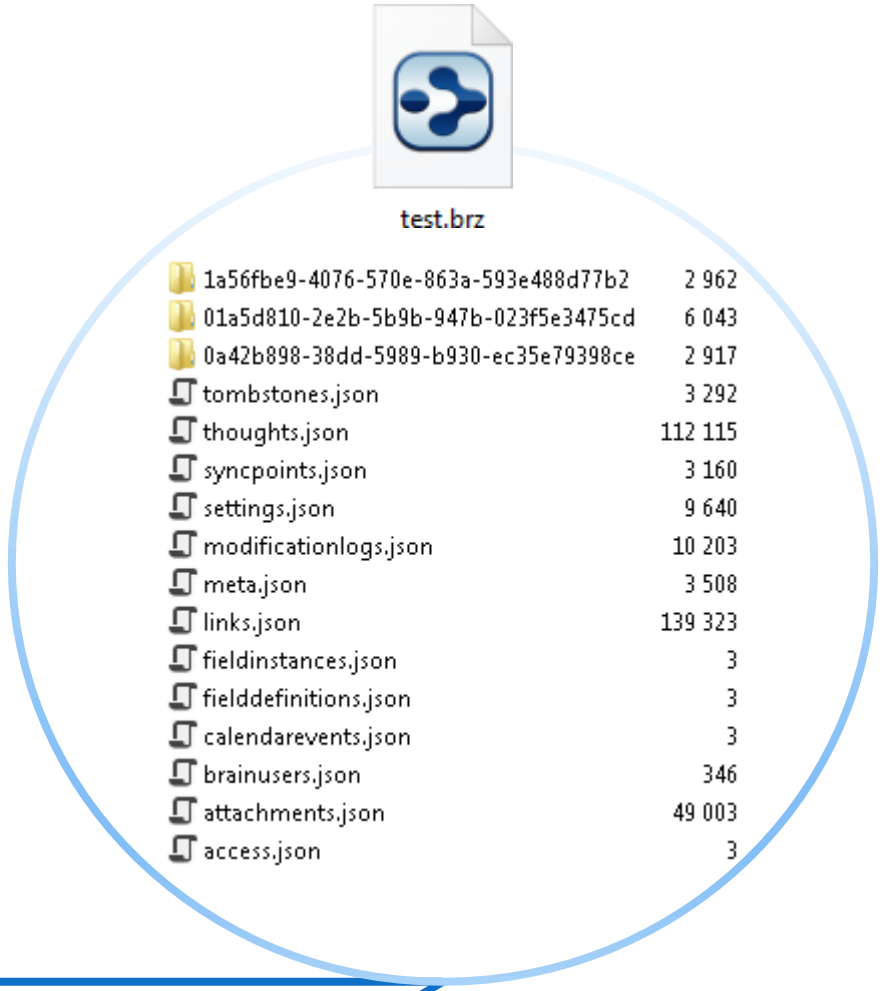


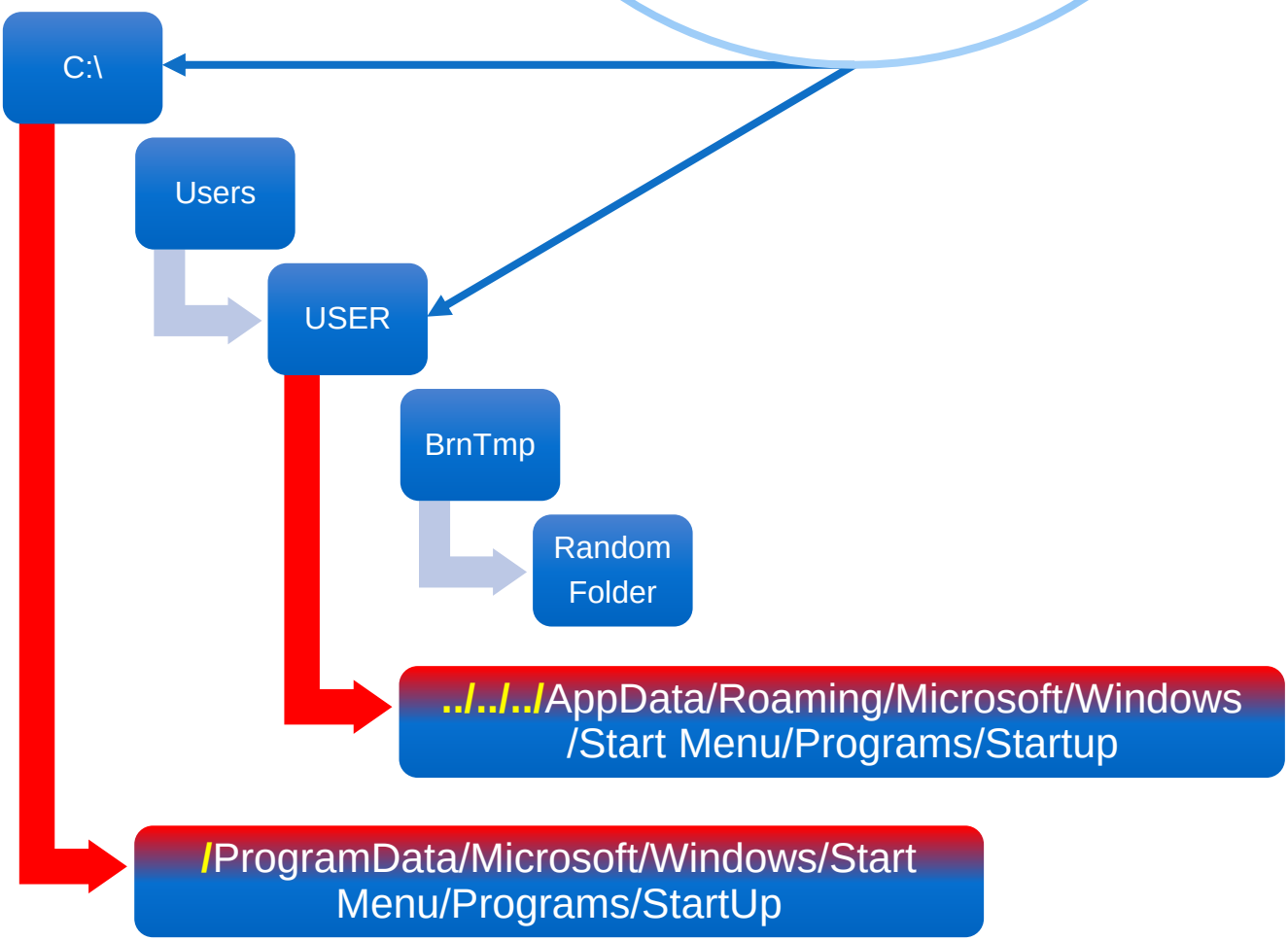
The content of the BRZ (ZIP) file is extracted by the software into a new temporary folder



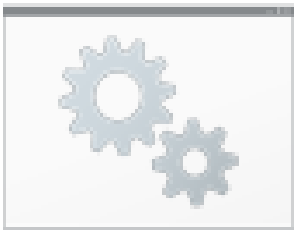


We can force it to extract the files in other folders for executing malware and editing files



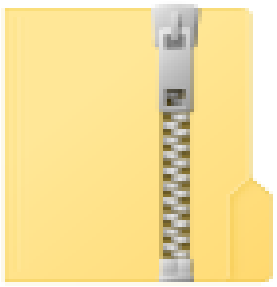


How to test



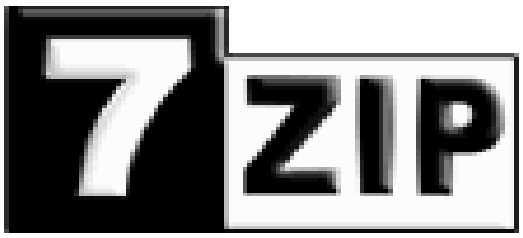
test.bat

Create a file

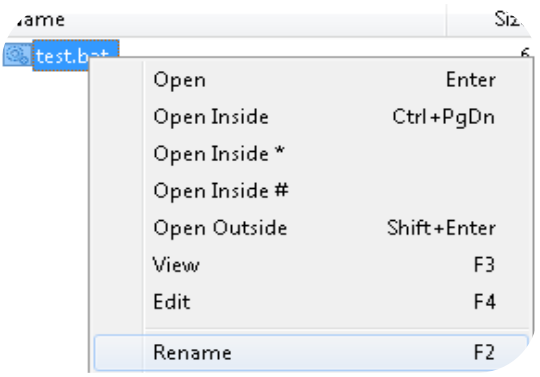


test.zip

ZIP the file



Open the ZIP file



Rename the archived file

../../../DIR/FILE

test.brz

Rename the ZIP to BRZ

Technical details

FileTypesMan

Extension	Type Name	Description	MIME Type
	LaunchWinApp.exe		
.BrainTheme	TheBrain.Theme	Brain Theme	application/BrainTheme
.brz	TheBrain.Document	TheBrain Archive	application/BrainArchive

Process Monitor

Operation	Path
CreateFile	C:\Users\test\BmTmp\kLDvL\M
CreateFile	C:\Users\test\BmTmp\kLDvL\M
CreateFile	C:\Users\test\BmTmp\kLDvL
CreateFile	C:\Users\test\BmTmp
QueryNetwork...	C:\Users\test\BmTmp
CloseFile	C:\Users\test\BmTmp
CreateFile	C:\Users\test\BmTmp\kLDvL
CloseFile	C:\Users\test\BmTmp\kLDvL
CreateFile	C:\Users\test\BmTmp\kLDvL\M
CloseFile	C:\Users\test\BmTmp\kLDvL\M
CreateFile	C:\Users\test\BmTmp\kLDvL\I
CreateFile	C:\Users\test\BmTmp\kLDvL\I
CreateFile	C:\Users\test\BmTmp\kLDvL
QueryNetwork...	C:\Users\test\BmTmp\kLDvL
CloseFile	C:\Users\test\BmTmp\kLDvL
CreateFile	C:\Users\test\BmTmp\kLDvL\I
CloseFile	C:\Users\test\BmTmp\kLDvL\I
CreateFile	C:\Users\test\BmTmp\kLDvL\I\6eca60c9-f6fa-5004-ab9d-bf0d4999659d\.data
CreateFile	C:\Users\test\BmTmp\kLDvL\I\6eca60c9-f6fa-5004-ab9d-bf0d4999659d\.data
CreateFile	C:\Users\test\BmTmp\kLDvL\I\6eca60c9-f6fa-5004-ab9d-bf0d4999659d
CreateFile	C:\Users\test\BmTmp\kLDvL\I
QueryNetwork...	C:\Users\test\BmTmp\kLDvL\I
CloseFile	C:\Users\test\BmTmp\kLDvL\I
CreateFile	C:\Users\test\BmTmp\kLDvL\I\6eca60c9-f6fa-5004-ab9d-bf0d4999659d
CloseFile	C:\Users\test\BmTmp\kLDvL\I\6eca60c9-f6fa-5004-ab9d-bf0d4999659d
CreateFile	C:\Users\test\BmTmp\kLDvL\I\6eca60c9-f6fa-5004-ab9d-bf0d4999659d\.data
CloseFile	C:\Users\test\BmTmp\kLDvL\I\6eca60c9-f6fa-5004-ab9d-bf0d4999659d\.data
CreateFile	C:\Users\test\BmTmp\kLDvL\I\6eca60c9-f6fa-5004-ab9d-bf0d4999659d\.data\Icon.png
CreateFile	C:\Users\test\BmTmp\kLDvL\I\6eca60c9-f6fa-5004-ab9d-bf0d4999659d\.data\Icon.png
WriteFile	C:\Users\test\BmTmp\kLDvL\I\6eca60c9-f6fa-5004-ab9d-bf0d4999659d\.data\Icon.png
CloseFile	C:\Users\test\BmTmp\kLDvL\I\6eca60c9-f6fa-5004-ab9d-bf0d4999659d\.data\Icon.png

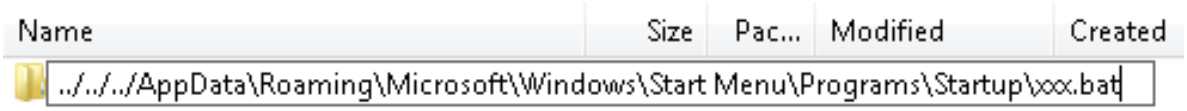
Source code via ILSpy

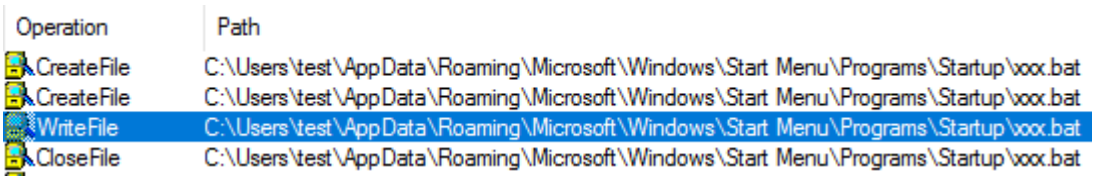
```
// TheBrain.Sys.FileUtility
using Ionic.Zip;
using System;
using System.IO;

public static void Unzip(string file, string dir, Action<string> logDelegate, bool
skipIfUpToDate)
{
	using (ZipFile zipFile = ZipFile.Read(file, new ReadOptions
	{
		Encoding = SlashConverterUtf8Encoding.Instance
	}))
	{
		foreach (ZipEntry item in zipFile)
		{
			string text = Utils.FilterIllegalFilenameCharacters(item.FileName,
			allowSlash: true);
			if (text != item.FileName)
			{
				logDelegate("Replacing illegal filename " + item.FileName + "
			with " + text + ".");
			}
			string text2 = Path.Combine(dir, text);
			if (item.IsDirectory)
			{
				Directory.CreateDirectory(text2);
				continue;
			}
			Directory.CreateDirectory(Directory.GetParent(text2).FullName);
			bool flag = false;
			bool flag2 = File.Exists(text2);
			if (skipIfUpToDate && flag2 && item.LastModified <=
			File.GetLastWriteTime(text2))
			{
				flag = true;
			}
			if (!flag)
			{
				if (flag2)
				{
					new FileInfo(text2).IsReadOnly = false;
				}
				using (FileStream stream = File.Create(text2))
				{
					item.Extract(stream);
				}
			}
		}
	}
}
```

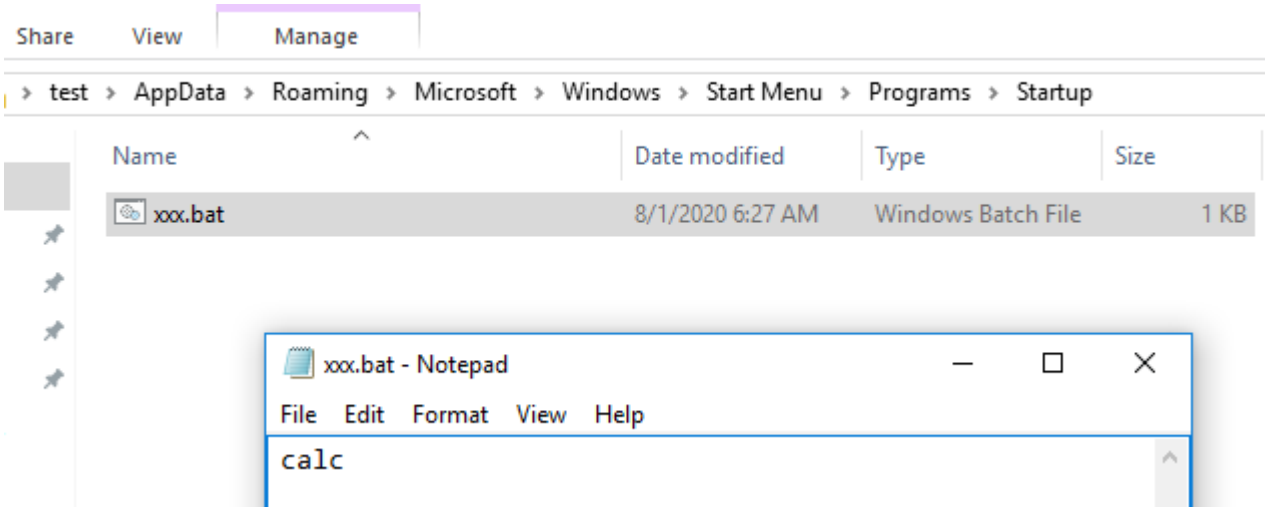
Source code via ILSpy

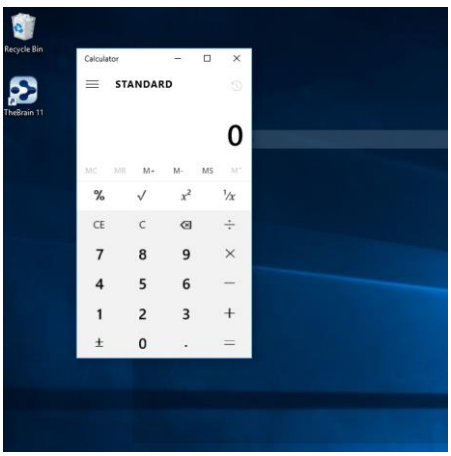
```
// TheBrain.Sys.Utils
using System.Text;

public static string FilterIllegalFilenameCharacters(string name, bool allowSlash =
false, bool clipName = true)
{
	string text = allowSlash ? "\\:*?<>|\"" : "/\\:*?<>|\"";
	StringBuilder stringBuilder = new StringBuilder(name);
	for (int i = 0; i < stringBuilder.Length; i++)
	{
		char c = stringBuilder[i];
		if (text.IndexOf(c) != -1 || c < ' ')
		{
			stringBuilder[i] = '_';
		}
	}
	string text2 = stringBuilder.ToString();
	string[] rESERVED_FILE_NAMES = RESERVED_FILE_NAMES;
	foreach (string value in rESERVED_FILE_NAMES)
	{
		if (text2.ToUpper().Equals(value))
		{
			return text2 + "_";
		}
	}
	if (text2.EndsWith("."))
	{
		text2 = text2.Substring(0, text2.Length - 1);
	}
	if (clipName && text2.Length > 255)
	{
		text2 = text2.Substring(0, 255);
	}
	return text2;
}
```

Proof-of-concept









Notes:

- Files can be overwritten
- Any file and extension is allowed
- Simple slides for beginners and for being viewed on mobile
- Just for teaching directory traversal vulnerabilities using a real example