Outlook Today Homepage Persistence

By Ben Wilson

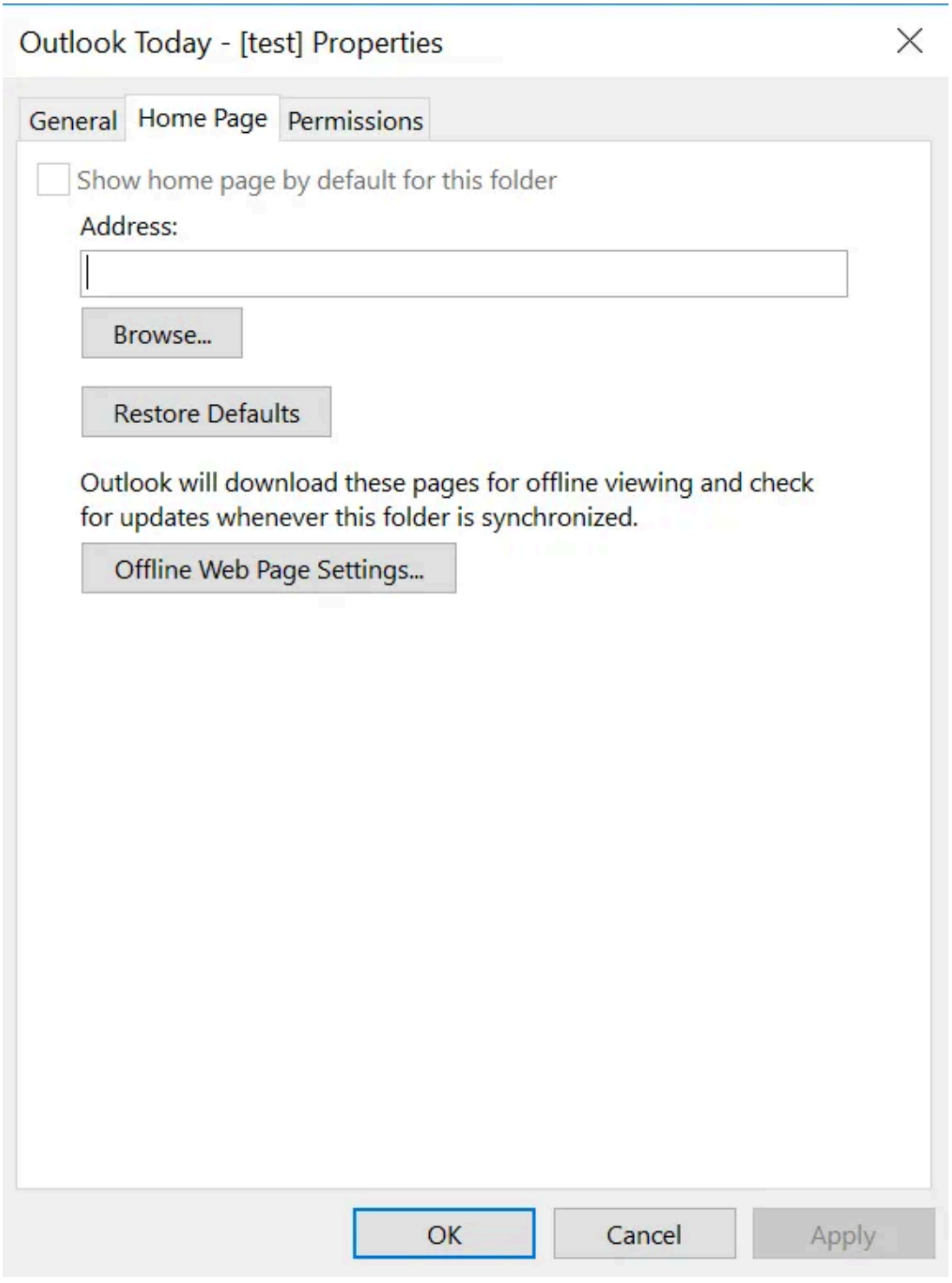
Published: 2018-09-15 · Archived: 2026-04-06 01:17:43 UTC

5 min read

Sep 15, 2018

Outlook Today

Under security update KB4011162, the homepage attack vector used by ruler was patched to prevent code execution. Therefore, I went investigating to see the extent of Microsoft's patch. While I was investigating, I came across Outlook Today, which according to Microsoft, is a handy way to get a quick interactive summary of your calendar, tasks, and messages for the current day. While I was looking through Outlook Today, I noticed that Outlook Today had a menu called data file properties (similar to properties under folders such as Inbox) and through that menu, you could once again set a homepage value.



This got me excited, but I needed to find out more about Outlook Today and explore the extent of my find. So opened up MFCMAPI and began exploring the properties. After hours and hours of searching for the original property PR_FOLDER_WEBVIEWINFO as well as other properties, which may hold the homepage value, it turned up nothing. After discussing with the developer of Ruler, Etienne Stalmans and the developer of MFCMAPI it was evident that the homepage value could not be set remotely and had to be set through the registry under:

“HKCU\Software\Microsoft\Office\16.0\Outlook\Today\UserDefinedUrl”

After hearing this, I decided not to give up and see if there was still a way to exploit this.

Persistence

I decided that if the Outlook Today Home Page could not be exploited remotely, it could still be used as a method of persistence, so I put my thinking cap back on and got back to work. Through research, I found that the Outlook Today page could be set as the startup folder in Outlook. This setting was located under File -> Options -> Advanced -> Outlook start and exit -> Browse -> Change Inbox to folder with your email address on it. This got me intrigued: if startup folders can be set through the Outlook GUI then there must be a way to edit the values programmatically. After doing some research and browsing the registry, I found the registry key located here:

“HKU\yoursid\Software\Microsoft\Office\16.0\Outlook\Profiles\Outlook\0a0d020000000000c000000000000046\001e0336”

It turns out this key is just a bunch of random numbers but through setting Outlook Today as the startup folder through the GUI and updating the registry we get this value: “IPF.TodaysOutlook”

Great, now we can set a custom URL for Outlook Today and set Outlook Today as the startup folder all through the registry, right? Not quite. I noticed that while trying to set the Outlook Today home page after it had been set back to default (using the “Restore Defaults” button in the GUI) it failed. For some reason the registry was unable to override the default settings put in place by Outlook.

Once again, I went back to the registry to try to find some answers. While I was changing the default homepage and restoring it while monitoring changes to the registry I realized that there was another value that was being changed, the stamp value found here:

“HKCU\Software\Microsoft\Office\16.0\Outlook\Today\Stamp”

Get Ben Wilson’s stories in your inbox

Join Medium for free to get updates from this writer.

Remember me for faster sign in

I found that every time you set a custom home page URL through Outlook GUI the Stamp value is set to 1, and every time you restore the default settings the Stamp value is set back to 0. This explains why the registry is unable to override the default settings as the Stamp value is acting as a lock and when the Stamp value is set to 1, the lock is open and new URLs are able to be set.

Sandbox Escape

Now that we are able to set a custom URL for Outlook Today and set Outlook Today as the startup folder programmatically, we need to find a way to abuse the exploit and escape the sandbox. Using some basic VBScript and HTML, I achieved this easily:

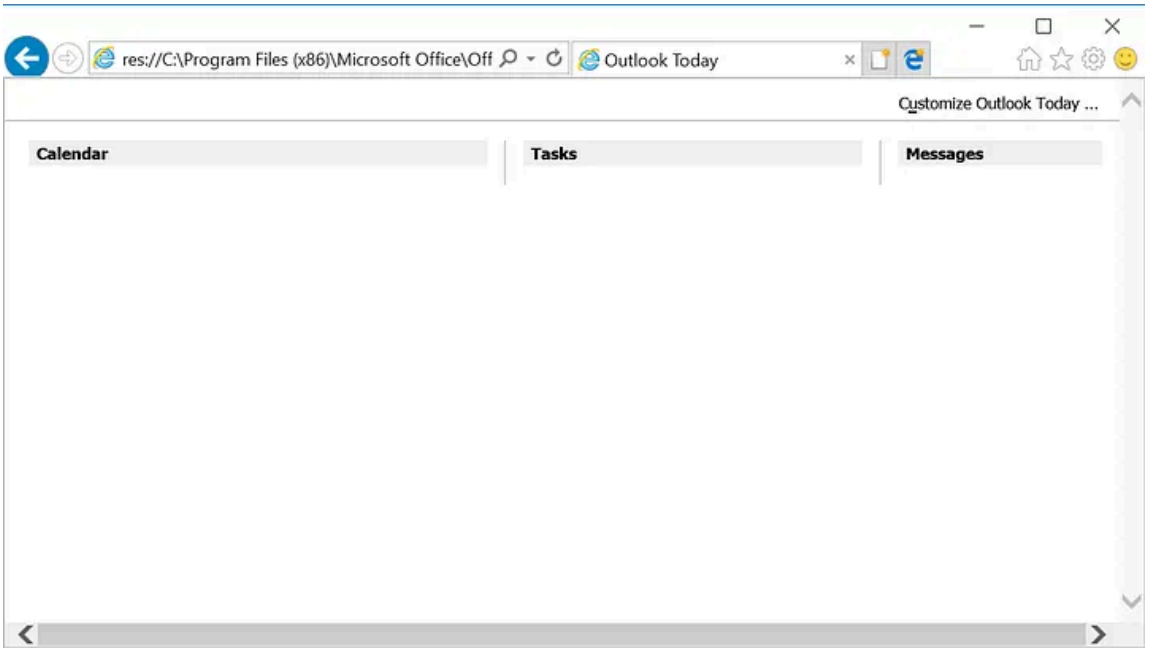
```
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>Outlook</title>
<script id=clientEventHandlersVBS language=vbscript>
<!--
```

```
Sub window_onload()
Set w = window.external.OutlookApplication
Set c = w.CreateObject("Wscript.Shell")
c.Run("calc.exe")
End Sub
-->
</script>
</head>
</html
```

Invisibility

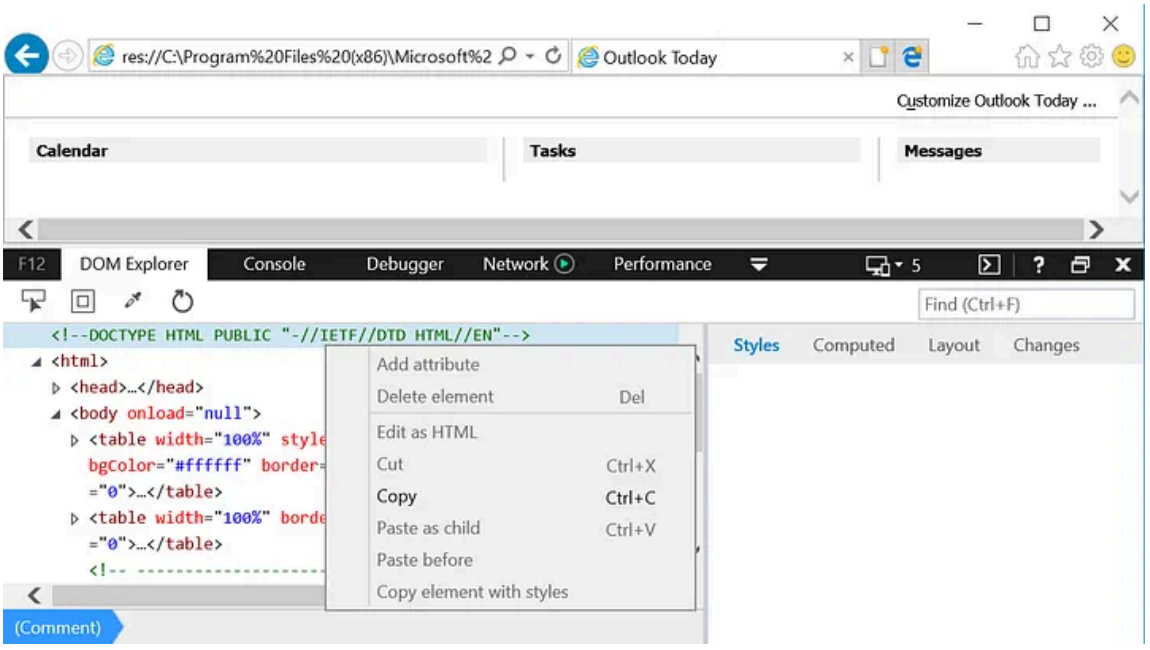
This is great and all but, if we want our persistence to remain invisible to the user then we are going to have to hide our shell better than this. Currently when the user opens Outlook there is no data provided, which may raise suspicion amongst observant users. To solve this we simply copy the default homepage URL for Outlook Today and paste it into Internet Explorer. It will look something like this.

Press enter or click to view image in full size



After that go to inspect element and then right click on the top of the HTML code and select copy

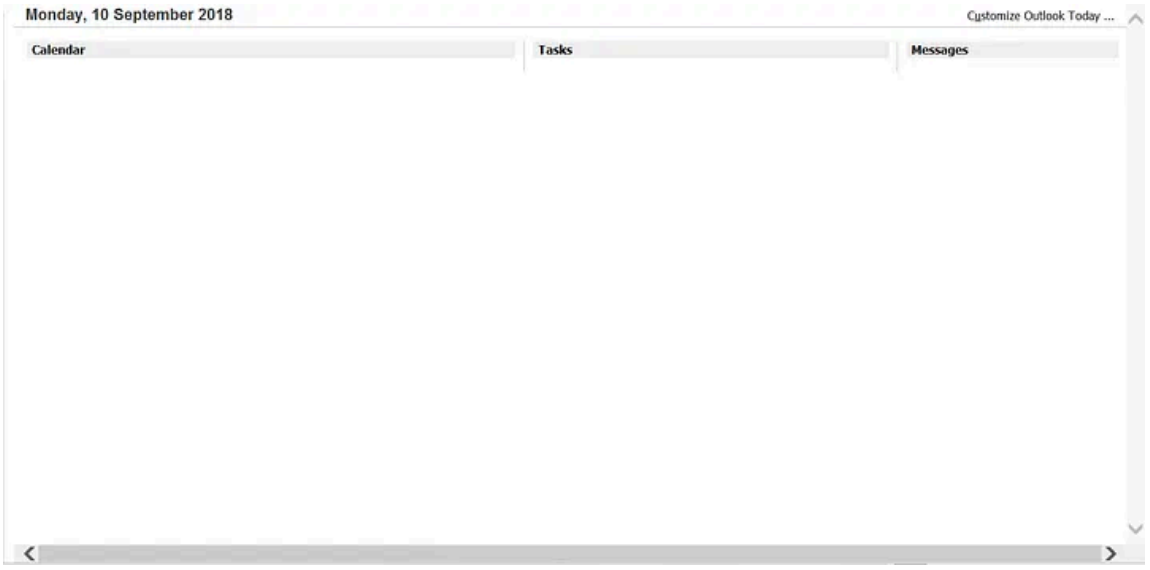
Press enter or click to view image in full size



Then paste this code above the current HTML sandbox escape code and remove duplicate the tags and now we have an invisible shell inside Outlook Today. Notice that the only difference between the two is the fact that our malicious page does not include the Inbox, Drafts and Outbox links shown under the Messages tab in the default Outlook Today home page.

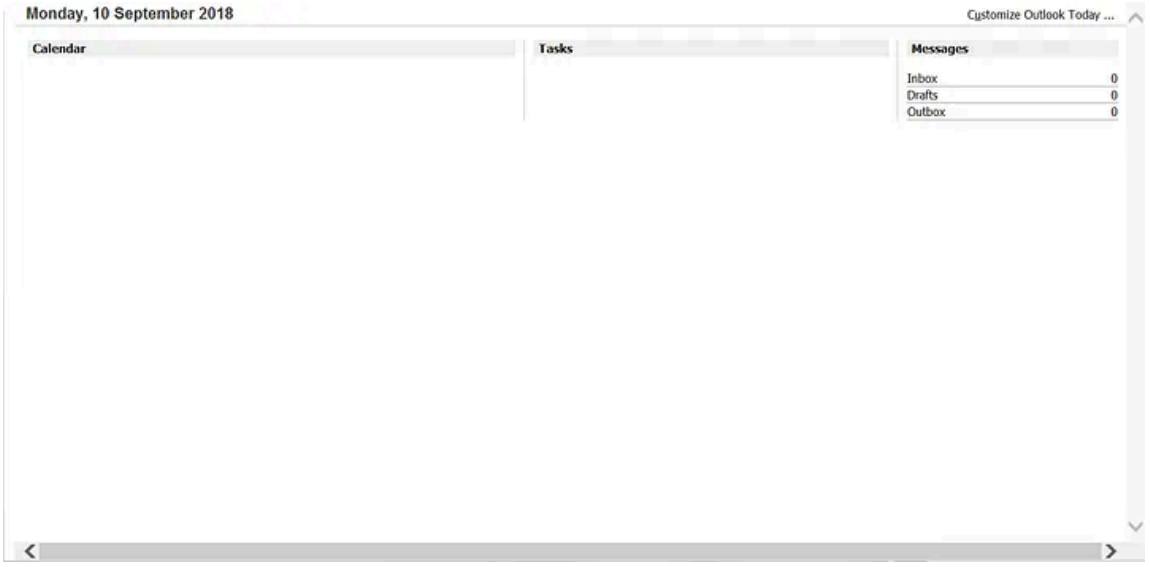
Malicious:

Press enter or click to view image in full size



Default:

Press enter or click to view image in full size



Conclusion

I have reported this issue to Microsoft and they said that because this exploit requires some form of initial access they have deemed it not serious enough to require a patch.

Although this exploit requires either initial access or physical access to the target, it is still a great technique for persistence if you are having trouble gaining persistence access on a target. In addition, because the registry keys are located under user editable directories, only basic privileges are required on the target to execute this exploit.

Source: https://medium.com/@bwtech789/outlook-today-homepage-persistence-33ea9b505943