July 29, 2024

Specula - Turning Outlook Into a C2 With One Registry Change

Written by Christopher Paschen and Oddvar Moe

Red Team Adversarial Attack Simulation

SKIP TO MAIN CONTENT

Table of contents

Share

There exist a few singular Registry changes that any non-privileged user can make that transform the Outlook email client into a beaconing C2 agent. Given that ***outlook.exe*** is a trusted process, this allows an attacker persistent access to a network that we have found often goes unnoticed. This technique has been reported on before and despite that continues to be a weak point in many otherwise very well-guarded networks.

Today, TrustedSec is releasing ***Specula*** (our previously internal framework) into the world, for leveraging this simple Registry change into an initial access platform. We have frequently leveraged ***Specula*** in our social engineering, p

ack paths. We hope our release

brings more light onto this attack path and helps a variety of organizations develop preventions for it. ***Specula's*** release will also assist our competitors in bringing further attention to it in their testing as well. If you do not care to read about this more and just want to see the code, it's available on GitHub or our vanity url http://specula.rocks.

History

Specula at its core is a C2 framework that operates via the Outlook home page feature. This is not anything specifically new, as other tooling (namely Ruler) exposes the functionality to create a home page that can attack this vector. The ability to abuse the Outlook home page was reported and listed as CVE-2017-11774. With that being the case, why are we releasing tooling related to the Outlook home page attack in 2024?

The Outlook home page was thought to have been patched in Knowledge Bases (KBs) listed under https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2017-11774. After the KB is installed, the UI elements related to Outlook's home page will be gone. This leads one to believe the associated functionality has been removed. Unfortunately, the Registry values that would have been set when the removed UI elements were used still get used by Outlook, even in current Office 365 installs.

Microsoft outlines this workaround to the missing UI elements. If an attacker can modify a single non-privileged Registry key, a C2 channel can be established despite it being thought to be a patched technique.

TrustedSec has been able to leverage this specific channel for initial access in hundreds of clients despite the existing knowledge and preventions available for this technique. For those reasons, we are releasing a cut-down version of our tooling to bring attention to this vector and hopefully close it for good. A graphical representation of how to set the needed Registry value for initial access is shown in the screenshot below.

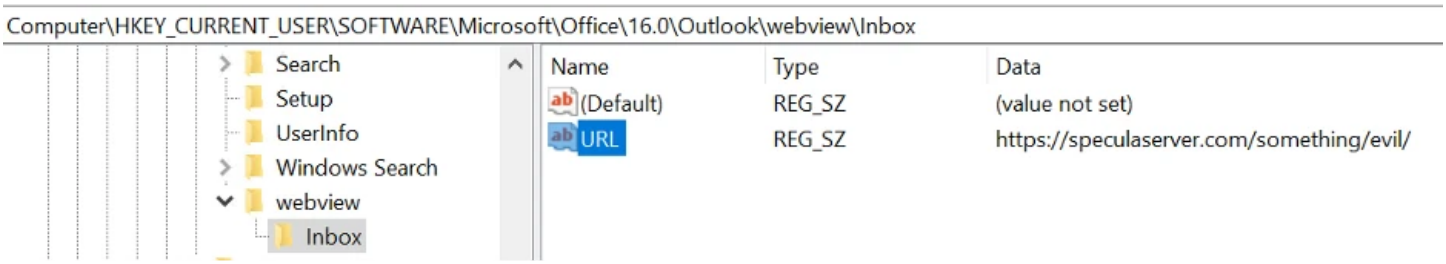


Figure 1 - Setting the Registry Value

How Specula Works

When a custom home page is set by any of the Registry keys outlined by Microsoft in their workaround, Outlook will download and display that HTML page instead of the normal mailbox element (inbox, calendar, sent, etc.) when the associated tab is selected. From the downloaded HTML page we're able to run vbscript or jscript within a privileged context with more or less full access to the local system as if we were running ***cscript / wscript.exe***. There are some controls that could impede this level of access, but those are also controlled by Registry keys a non-privileged user can access.

Microsoft also helpfully provides a COM object that can be embedded in the HTML page we create. This COM object (available via clsid 0006F063-0000-0000-C000-000000000046) renders the currently selected view for Outlook and properly communicates with Outlook to display what the user would expect from the default controls. There are a few other clsids that achieve the same goal, and we leave finding them as an exercise to the reader.

In effect, we can load a web page that relies on a COM object provided by Outlook to render a view that looks and acts exactly like what a user would expect, all while being able to run additional code in the background. The screenshot below shows an empty harness of code that will run every 300 seconds.

```
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta http-equiv="refresh" content="300">
<title>Outlook</title>
<script id=clientEventHandlersVBS language=vbscript>
<!--
-->
</script>
</head>
<body>
 <object classid="clsid:0006F063-0000-0000-C000-000000000046" id="ViewCtl1" data=""
width="100%" height="100%"></object>
</body>
</html>
```

Figure 2 - Empty Harness of Code

Now that we can load code, we can set up a timer and trigger function containing any method that allows downloading a resource. In what we are releasing, this is accomplished with the MSXML2.ServerXMLHTTP COM object. After this resource is grabbed, it can be executed using the `ExecuteGlobal` command. Put that on a regularly occurring timer and you have your traditional beaconing C2.

Notably, the resources rendered and returned via ***Specula*** allow for execution of vbscript within a trusted context. Full access is also allowed to any COM object that exposes methods via an IDispatch interface.

The Specula Framework

Specula is different from existing tooling around the home page attack in that it is a whole framework dedicated to leveraging and executing commands on a continuing basis, as opposed to a one-off command. ***Specula*** removes the burden of manually changing out payloads. In addition, it was coded to be easily extendable without requd the framework.

Specula has features to allow a deployment to be pre-staged (bypassing validation checks) or staged (must check in x times before it can be tasked, or manually approved). This is intended to confuse an Incident Response team investigating the URLs used by ***Specula*** to host content. Once a client has reached the number of check-ins required (or manually approved), a unique per-host encryption key is generated and set on the target. The home page URL is also updated once approved.

From then on, the tasking is downloaded, decrypted, and executed via `ExecuteGlobal`. If there is no tasking, nothing is executed, and code is configured to wait and check again after the specified time delay has passed.

To create tasking for a given Outlook client, the operator will interact with that agent and then specify a module they would like to run. These are currently broken up into api, enumerate, execute, exploit, operation, and trolling. Once a module is selected the operator will configure the required options for that module and then execute it. On the back-end this uses ***jinja*** to transform a vbscript template into our payload and encrypt it with the associated key. Tasking is then executed and posted back to the ***Specula*** server where it can be enumerated via the agent's logs.

Extending Specula

For a detailed explanation, see: https://github.com/trustedsec/specula/wiki/Developing-New-Modules.

Specula's available actions are split into modules. This allows us to only send the code required for an individual task and not expose our entire toolset on any given command run. These modules are developed as a python file that creates a class named 'Spec'. Spec must subclass ***SpecModule***. There is an associated vbscript template written in the ***jinja*** template language that must have the same base name as the ***.py*** file and must end with ***.txt***. The function that the Outlook client should call is defined by the ***entry*** field of the Spec class. Any scripts with dependent functions are called out in the 'depends' list of the class. Finally, ***self.options*** is a dictionary of strings-to-dictionaries that specifies the options an operator can modify and set. The options key is the name used to reference the option. Its value is further spelled out in the linked wiki above. Pre-processing can be done if more complex

processing or additional work is required to generate the required task. Lastly, if post-processing of the return data is desired, the function `rethandler` can be overridden.

Preventing Home Page Attacks

First, if possible for your organization, start using the 'New Outlook'.

Figure 3 - New Outlook

This new version operates as a packaged web page more so than a full desktop Outlook client. It lacks compatibility with COM extensions and kills a few more attack paths related to Outlook beyond the home page attack. In this case, the associated Registry key is not checked, and the feature is well and truly removed from this version of Outlook.

Second, in upcoming versions of Windows 11, the vbscript engine will be a component that can be removed and will cripple this attack vector since vbscript is needed to run the code. As of Windows 11 24H2, you will be able to remove this feature of Windows, preventing abuse; however, it is ***on*** by default. Future Windows 11 versions will have this feature ***off*** by default. According to the timeline presented here, vbscript will be off by default as of 2027.

Third, Group Policy Object (GPO) can be used to configure the associated keys and set others that outright disable WebView. The recommended method to attempt using Group Policy is well outlined by Mandiant here. The associated section is titled Locking Down the Registry Keys Using GPO Enforcement. An important note is that the blog post uses a lot of Group Policy Preferences (GPP) settings, and these are just that, preferences. The correct way is to leverage Group Policy settings to prevent the user from manipulating the settings after being set. The most important setting is the ***Do not allow Home Page URL to be set in folder Properties*** policy setting. This policy setting will turn off WebView for the folders inside Outlook for the user. There is one exception though, and that is the

Outlook Today page. This page is displayed when you press the mailbox name in the navigation. The screenshot below shows the Outlook Today page.

Figure 4 - Outlook Today Page

The Outlook Today page shares the feature set of WebView, and you just have to create a ***REG_SZ*** value named URL under ***HKCU\Software\Microsoft\Office\16.0\Outlook\Today***. This can be remediated with setting the Group Policy setting named Outlook Today availability to ***disabled***.

Figure 5 - Outlook Today - Disabled Setting

Note: If you do not see the Office settings, you will need to add the appropriate ADMX files from Microsoft for the version of Office you are using in your environment.

The fourth option is to leverage the baselines inside the Microsoft Security Compliance Toolkit. In our testing, the baselines seem to have locked down the web engine that Outlook uses for rendering HTML and vbscript, causing it not to run the scripts.

Detecting Home Page Attacks

Detecting home page attacks should be straightforward. All you need to do is set up monitoring if a URL value is added or present under these locations:

- HKCU\Software\Microsoft\Office\16.0\Outlook\WebView\Inbox
- HKCU\Software\Microsoft\Office\16.0\Outlook\WebView\Calendar
- HKCU\Software\Microsoft\Office\16.0\Outlook\WebView\Contacts
- HKCU\Software\Microsoft\Office\16.0\Outlook\WebView\Deleted Items
- HKCU\Software\Microsoft\Office\16.0\Outlook\WebView\Drafts
- HKCU\Software\Microsoft\Office\16.0\Outlook\WebView\Journal
- HKCU\Software\Microsoft\Office\16.0\Outlook\WebView\Junk E-mail
- HKCU\Software\Microsoft\Office\16.0\Outlook\WebView\Notes
- HKCU\Software\Microsoft\Office\16.0\Outlook\WebView\Outbox
- HKCU\Software\Microsoft\Office\16.0\Outlook\WebView\RSS
- HKCU\Software\Microsoft\Office\16.0\Outlook\WebView\Sent Mail
- HKCU\Software\Microsoft\Office\16.0\Outlook\WebView\Tasks
- HKCU\Software\Microsoft\Office\16.0\Outlook\Today

- HKCU\Software\Microsoft\Office\15.0\Outlook\WebView\Inbox
- HKCU\Software\Microsoft\Office\15.0\Outlook\WebView\Calendar
- HKCU\Software\Microsoft\Office\15.0\Outlook\WebView\Contacts
- HKCU\Software\Microsoft\Office\15.0\Outlook\WebView\Deleted Items
- HKCU\Software\Microsoft\Office\15.0\Outlook\WebView\Drafts
- HKCU\Software\Microsoft\Office\15.0\Outlook\WebView\Journal
- HKCU\Software\Microsoft\Office\15.0\Outlook\WebView\Junk E-mail
- HKCU\Software\Microsoft\Office\15.0\Outlook\WebView\Notes
- HKCU\Software\Microsoft\Office\15.0\Outlook\WebView\Outbox
- HKCU\Software\Microsoft\Office\15.0\Outlook\WebView\RSS
- HKCU\Software\Microsoft\Office\15.0\Outlook\WebView\Sent Mail
- HKCU\Software\Microsoft\Office\15.0\Outlook\WebView\Tasks
- HKCU\Software\Microsoft\Office\15.0\Outlook\Today
- HKCU\Software\Microsoft\Office\14.0\Outlook\WebView\Inbox
- HKCU\Software\Microsoft\Office\14.0\Outlook\WebView\Calendar
- HKCU\Software\Microsoft\Office\14.0\Outlook\WebView\Contacts
- HKCU\Software\Microsoft\Office\14.0\Outlook\WebView\Deleted Items
- HKCU\Software\Microsoft\Office\14.0\Outlook\WebView\Drafts
- HKCU\Software\Microsoft\Office\14.0\Outlook\WebView\Journal
- HKCU\Software\Microsoft\Office\14.0\Outlook\WebView\Junk E-mail
- HKCU\Software\Microsoft\Office\14.0\Outlook\WebView\Notes
- HKCU\Software\Microsoft\Office\14.0\Outlook\WebView\Outbox
- HKCU\Software\Microsoft\Office\14.0\Outlook\WebView\RSS
- HKCU\Software\Microsoft\Office\14.0\Outlook\WebView\Sent Mail
- HKCU\Software\Microsoft\Office\14.0\Outlook\WebView\Tasks

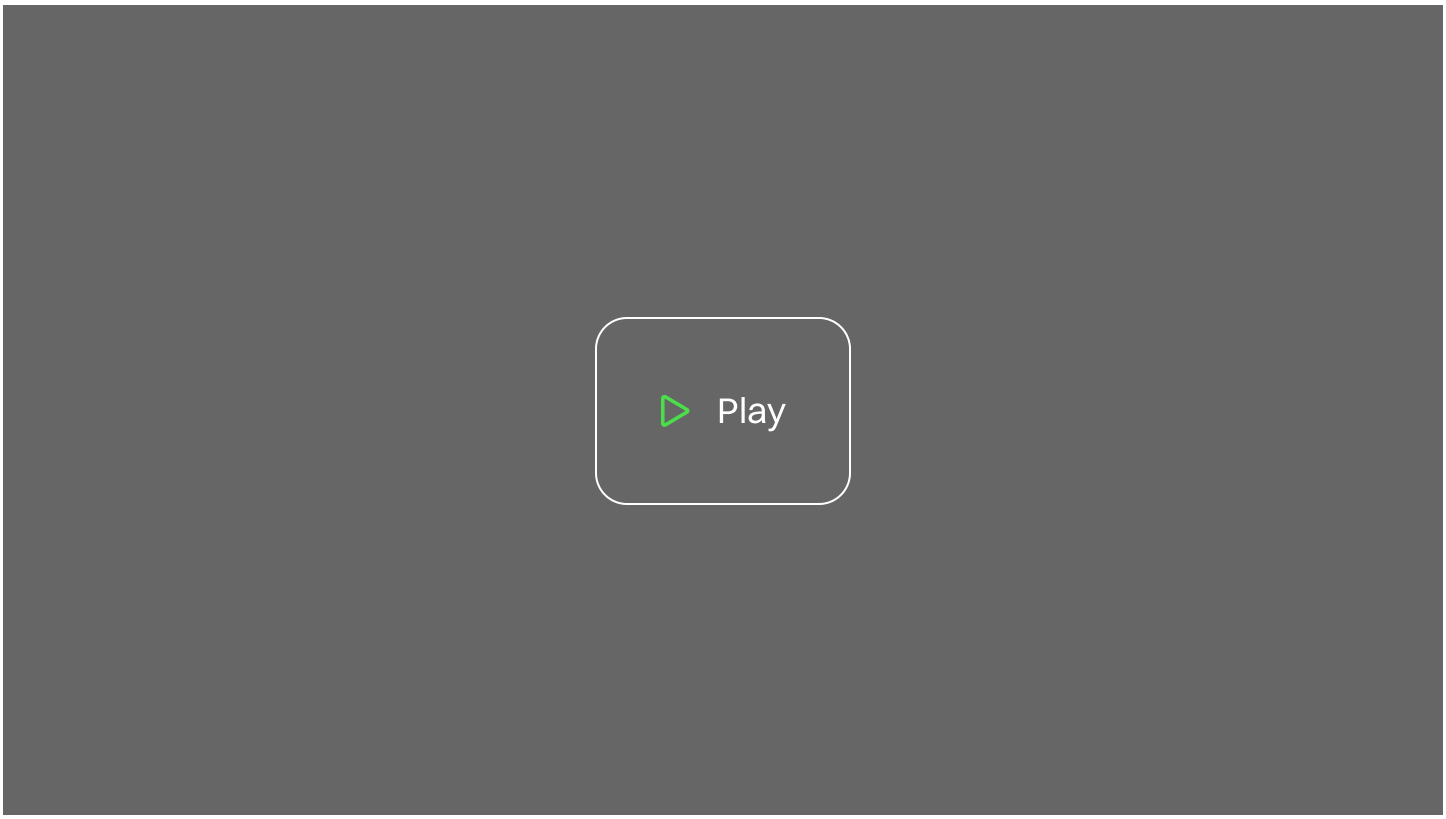
- HKCU\Software\Microsoft\Office\14.0\Outlook\Today

We have not seen any organization that has ever used the Outlook WebView for legitimate reasons, but we still recommend that you check before you start monitoring and preventing access to the URL value.

Conclusion and More Info

If you would like to dive into Specula more, we've made a collection of usage and development videos for specula that you can find on our YouTube channel.

If you have questions about Specula, you can find the developers on the Trustedsec Discord account.



TRUSTEDSEC

3485 Southwestern Boulevard
Fairlawn, OH 44333

1-877-550-4728