Remote Exploitation of Microsoft Office DLL Hijacking (MS15-132) via Browsers

A number of weeks back, security researcher Parvez Anwar posted a number of DLL hijacking vulnerabilities within Microsoft Office on Twitter (https://twitter.com/ParvezGHH/status/672433593558396929) [1]. The following week, Microsoft released MS15-132, which addressed some of these vulnerabilities, along with a large number of very similar bugs reported by others (https://code.google.com/p/google-security-research/issues/detail?id=556) in various guises (https://www.securify.nl/blog/SFY20151201/there_s_a_party_in_ole__and_you_are_invited.html). [2] [3].

The vulnerabilities that were uncovered were reported to affect various versions of Microsoft Office, on various versions of Windows. They could be triggered from within .docx, .rtf, and .pptx files, abusing the way in which Windows loads embedded OLE objects (and subsequently, any CLSID registered COM object), from within an Office document.

If you wish to learn more about this specific class of vulnerability, please refer to Haifei Li and Bing Sun's excellent "*Attacking Interoperability: An OLE Edition* (https://www.blackhat.com/docs/us-15/materials/us-15-Li-Attacking-Interoperability-An-OLE-Edition.pdf)" presentation from this year's BlackHat conference [4]. Additionally, NCC Group's Dominic Wang has written the whitepaper "*Understanding Microsoft Word OLE Exploit Primitives: Exploiting CVE-2015-1642 Microsoft Office CTaskSymbol Use-After-Free Vulnerability* (/uk/our-research/understanding-microsoft-word-ole-exploit-primitives/)" [5], which provides further background on how Microsoft Office handles embedded ActiveX controls.

However, to provide a basic overview of what is happening in these specific vulnerabilities:

- When loading a document containing an embedded OLE object, referenced by CLSID, or ProgID, Windows will look up the referenced ID from the registry, and try to load the corresponding DLL, calling the relevant code to create an instance of the object requested.
- Once the DLL is loaded into Office's address space, Office will *then* work out whether in fact the loaded object is suitable for the document (or is in fact, not an OLE object at all!).

- If a required DLL does not exist in the specified location, Windows will attempt to use pre-specified locations to search for the DLL (https://msdn.microsoft.com/en-us/library/7d83bc18.aspx) [6]. This includes the current directory.
- In the 0-day vulnerabilities, the specified CLSIDs (when embedded within an Office document), would cause the DLLs “elsext.dll, api-ms-win-core-winrt-l1-1-0.dll, OCIW32.DLL or oci.dll”, to be loaded from the current directory. If the attacker places one of these DLLs within the same directory as the document containing the embedded object, the attacker-supplied code will be executed by Office.

Now with that background out of the way, it’s time to talk about exploitation. Traditionally, DLL hijacking exploits have required that the user launches the malicious file from an SMB or WebDAV share. This is because, when launching from a remote share, the current directory for the document will be the share from which it was launched. The attacker obviously controls this, and thus can easily “plant” the malicious DLL within the same directory as the document. In this post we are going to discuss how an attacker can use a number of quirks in both Office and Firefox in order to exploit these issues remotely, without the need for SMB or WebDAV.

Office Temp folder file dropping

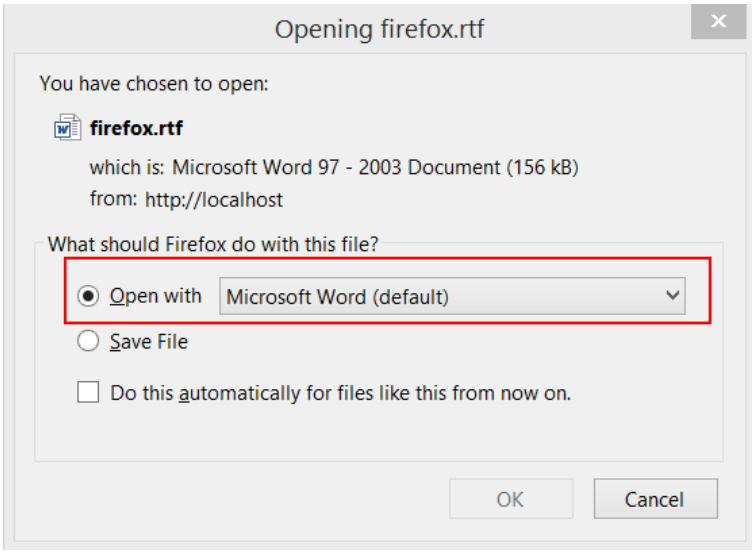
To exploit this issue remotely, we are going to use a number of known issues and tricks. The first of these relates to a temp file dropping issue in Microsoft Office. This issue was discovered by Haifei Li, after its potential use by attackers had been seen in the wild (https://blogs.mcafee.com/mcafee-labs/dropping-files-temp-folder-raises-security-concerns/) [7]. For this trick to work, an attacker simply has to embed their payload within an RTF file, using the Package ActiveX control. When the RTF file is opened, the attacker’s payload will be dropped into the user’s %TEMP% folder, retaining its original name (unless it already exists within the %TEMP% folder, in which case the name will be suffixed with a number, e.g. <filename> (2). <ext>).

In our exploit, we will take advantage of this known issue to embed our payload, oci.dll. This can be achieved simply by dragging and dropping our malicious DLL into our exploit document, and saving it as .rtf. Now when re-opening the document, you will see the file oci.dll get written into the current user’s %TEMP% directory:

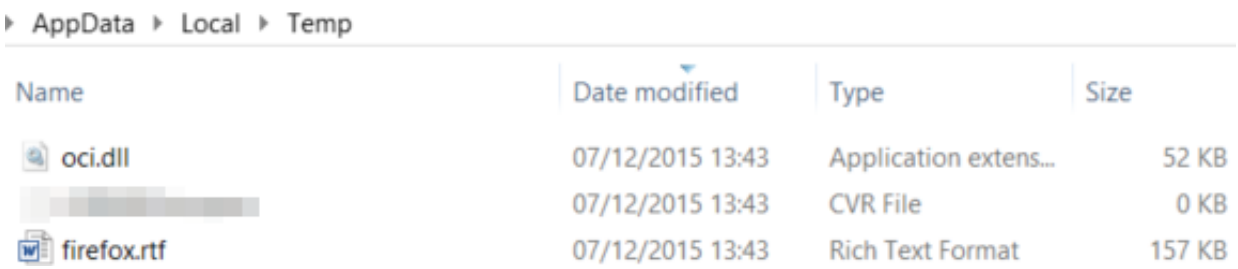
Firefox Temp Folder

The second trick we will take advantage of is the fact that Mozilla Firefox uses the current user’s %TEMP% directory as its default download directory when selecting “Open With” in Firefox’s download dialogue.

When a user downloads a file using Firefox, they are prompted with two choices (as seen below): “Save As”, and “Open With”.



When selecting “Open With”, Firefox will download the file into the user’s %TEMP% directory, before opening it with the specified program. At this point, the launched program’s current working directory becomes the user’s %TEMP% directory... see where we’re going with this?

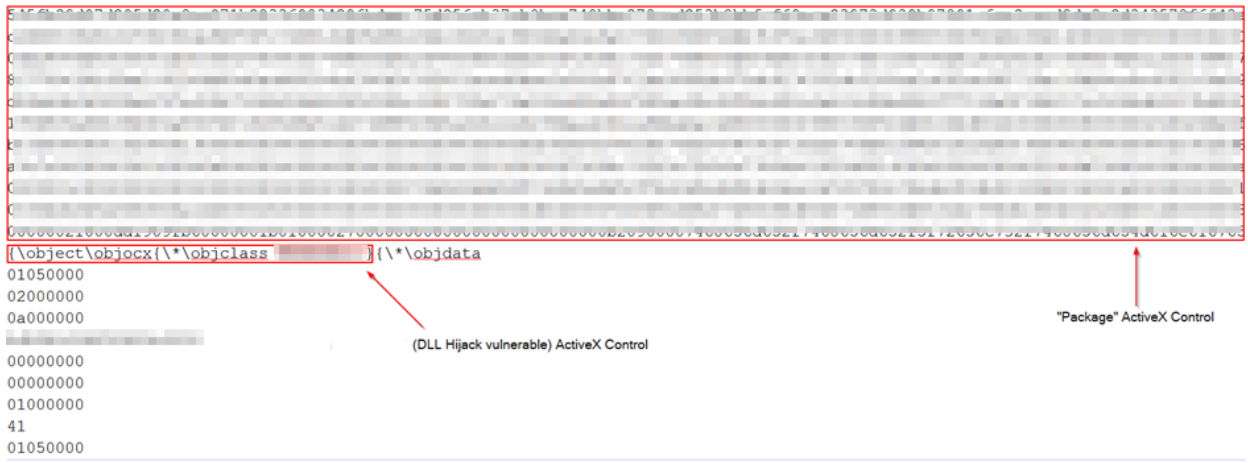


We can exploit this behaviour by crafting an RTF file which includes both our embedded “Package” ActiveX control, and the vulnerable object, one after the other. When the document is downloaded from the attacker’s web server (and the “Open With” (Microsoft Word) option is chosen) via Firefox, two things will happen:

Firstly, oci.dll will be dropped into the user’s %TEMP% folder.

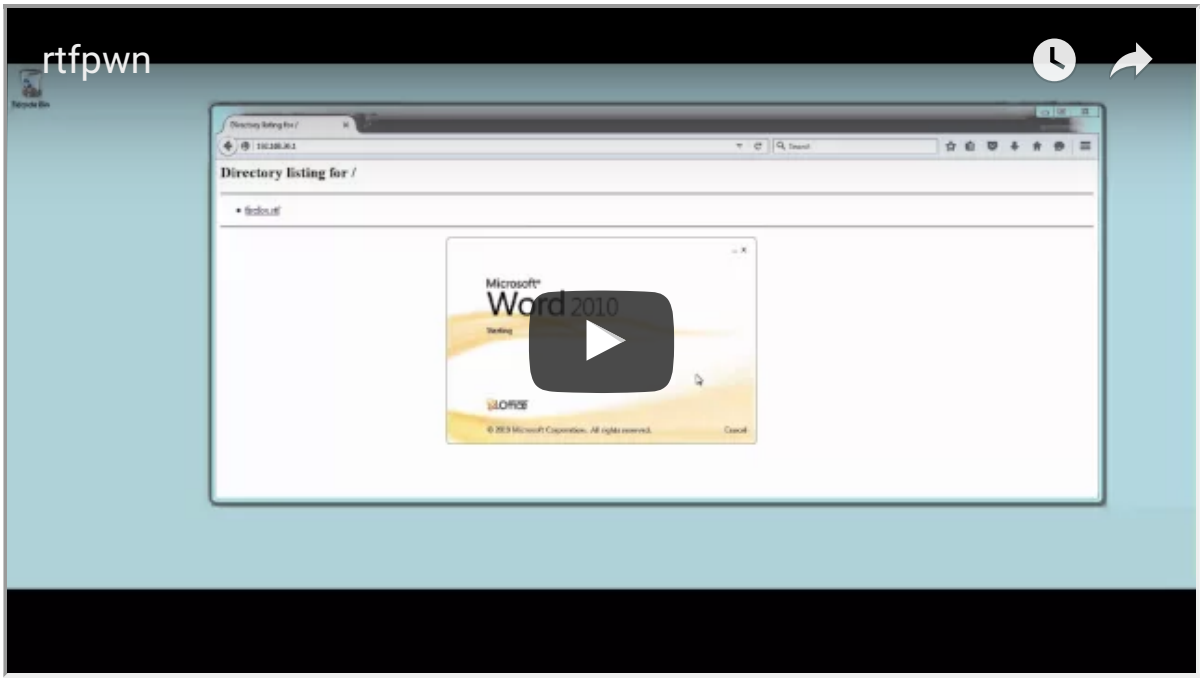
Secondly, the object is loaded, causing Microsoft Word to attempt to load the DLL oci.dll from the current directory.

As the current directory is now %TEMP%, the attacker’s payload will be executed by Word.



Demo

The following demo video shows how it was possible to take advantage of these issues in order to gain remote code execution via Firefox:



Microsoft Edge and Google Chrome

Microsoft Edge and Google Chrome both use the user's "Downloads" directory for storing downloaded files, and do not prompt before downloading files. This can be misused by an attacker, using an "auto-download" attack; whereby the attacker can force the user's browser to download the attacker-supplied DLL in addition to the crafted document. When the document is opened, the DLL will be loaded from the "Downloads" directory (where it has just been downloaded to, using the auto-download method).

The example below shows how it's possible to force a DLL to be downloaded using the <iframe src="oci.dll"> method:

```
<html>
<script>
var iframe = document.createElement('iframe');
iframe.src = 'oci.dll';
iframe.id = "dllframe";
iframe.width = 0;iframe.height = 0;
iframe.style.visibility = "hidden";
document.body.appendChild(iframe);
function dropDoc() {
      window.location = "exploit.docx";
}
aaa = setTimeout(dropDoc, 3000);
</script>
</html>
```

This technique has also been documented recently by Haifei Li (http://justhaifei1.blogspot.com/2015/10/watch-your-downloads-risk-of-auto.html) [8].

Note that on both Google Chrome and Microsoft Edge, an additional warning will be shown if the downloaded file (i.e. attacker's DLL) has a low reputation. Note that on Edge, the warning is only shown until the next file is downloaded and doesn't actually prevent the file from being downloaded. So using the PoC above, the warning message is only shown briefly until the .docx file is dropped, at which point it's masked by the "Open" dialogue.

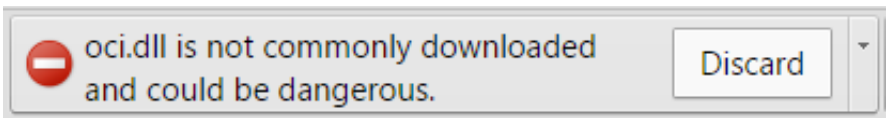


Figure 1 - Chrome Warning

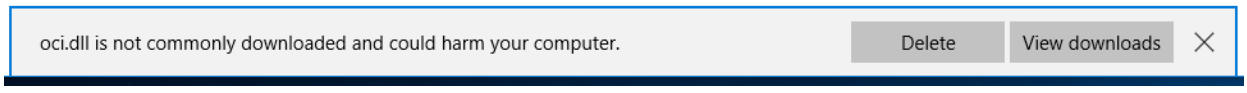


Figure 2 - Edge Warning

Figure 3 - Edge "Open" dialogue

Mitigations

Microsoft have released MS15-132, which addresses the DLL Hijacking vulnerabilities discussed in this blog post, however this patch does not address all of the issues that have been publically disclosed. Furthermore, due to the wide attack surface exposed by OLE, specifically for this class of vulnerability - we can expect more vulnerabilities of this nature to be discovered, and potentially exploited by threat actors in the near-future.

We contacted Mozilla for a workaround/fix for the shared %TEMP% folder and they responded saying it is a bug in Microsoft Windows/Office as opposed to Firefox.

Google Chrome can be configured to prompt before downloading files. To enable this, select: “Settings > Show Advanced > Ask where to save each file before downloading.”

References

[1] - https://twitter.com/ParvezGHH/status/672433593558396929 (https://twitter.com/ParvezGHH/status/672433593558396929)

[2] - https://code.google.com/p/google-security-research/issues/detail?id=556 (https://code.google.com/p/google-security-research/issues/detail?id=556)

[3] - https://www.securify.nl/blog/SFY20151201/there_s_a_party_in_ole__and_you_are_invited.html (https://www.securify.nl/blog/SFY20151201/there_s_a_party_in_ole__and_you_are_invited.html)

[4] - https://www.blackhat.com/docs/us-15/materials/us-15-Li-Attacking-Interoperability-An-OLE-Edition.pdf (https://www.blackhat.com/docs/us-15/materials/us-15-Li-Attacking-Interoperability-An-OLE-Edition.pdf)

[5] - https://www.nccgroup.trust/uk/our-research/understanding-microsoft-word-ole-exploit-primitives (/uk/our-research/understanding-microsoft-word-ole-exploit-primitives/)/

[6] - https://msdn.microsoft.com/en-us/library/7d83bc18.aspx (https://msdn.microsoft.com/en-us/library/7d83bc18.aspx)

[7] - https://blogs.mcafee.com/mcafee-labs/dropping-files-temp-folder-raises-security-concerns/ (https://blogs.mcafee.com/mcafee-labs/dropping-files-temp-folder-raises-security-concerns/)

[8] - http://justhaifei1.blogspot.com/2015/10/watch-your-downloads-risk-of-auto.html (http://justhaifei1.blogspot.com/2015/10/watch-your-downloads-risk-of-auto.html)

[9] - https://technet.microsoft.com/en-us/library/security/ms15-132.aspx (https://technet.microsoft.com/en-us/library/security/ms15-132.aspx)

Published date: 05 January 2016

Written by: Richard Warren

- [ ] Security Consulting
- [ ] Software Testing
- [ ] Website Performance
- [ ] Domain Services
- [ ] Corporate
- [ ] Business Insights

Filter By Date

January (2) (/uk/about-us/newsroom-and-events/blogs/?Year=2016&Month=1)

Call us on:
+44(0)161 826 7555 (tel:+441612095200)

Share Prices (/uk/about-us/investor-relations/share-price/)
Results & Presentations (/uk/about-us/investor-relations/results-and-presentations/)
Stock Exchange Announcements (/uk/about-us/investor-relations/stock-exchange-announcements/)

Legal

Terms & Conditions (/uk/about-us/terms-and-conditions/)
Privacy Policy (/uk/about-us/privacy-policy/)
Accessibility (/uk/about-us/accessibility/)
Sitemap (/uk/sitemap/)

Latest from @NCCGroupplc (https://twitter.com/NCCGroupplc)

NCC Group plc 2h
@NCCGroupplc
We work hard to recruit the best. Take a look at our current vacancies if you fancy a new challenge nccgroup.trust/uk/about-us/ca…
Expand

NCC Group plc 3h
@NCCGroupplc
#Phishing the simple and effective blended attack nccgroup.trust/uk/about-us/ne…