MDSec

ACTIVEBREACH

Exploiting CVE-2023-23397: Microsoft Outlook Elevation of Privilege Vulnerability

Date: 14th March 2023

Today saw Microsoft patch an interesting vulnerability in Microsoft Outlook. The vulnerability is described as follows:

> *Microsoft Office Outlook contains a privilege escalation vulnerability that allows for a NTLM Relay attack against another service to authenticate as the user.*

However, no specific details were provided on how to exploit the vulnerability.

At MDSec, we're continually looking to weaponise both private and public vulnerabilities to assist us during our red team operations. Having recently given a talk on leveraging NTLM relaying during red team engagements at FiestaCon, this vulnerability particularly stood out to me and warranted further analysis.

While no particular details were provided, Microsoft did provide a script to audit your Exchange server for mail items that might be being used to exploit the issue.

Review of the audit script reveals it is specifically looking for the *PidLidReminderFileParameter* property inside the mail items and offers the option to "clean" it if found:

```
try {
    if ($CleanupAction -eq "ClearItem") {
        $item.Delete([Microsoft.Exchange.WebServices.Data.DeleteMode]::HardDelete)
    } else {
        if (-not $item.RemoveExtendedProperty($mailInfo["PidLidReminderFileParameter"])) {
            Write-Host ("Failed to clear property for entry number: $entryCount, Line number: $($entryCount + 1)")
            $invalidEntries.Add($entryCount)
            continue
        }
```

Diving in to what this property is, we find the following definition:

Applies to: Outlook 2013 | Outlook 2016

Specifies the filename of the sound that a client should play when the reminder for that object becomes overdue.

Property	Value
Associated properties:	dispidReminderFileParam
Property set:	PSETID_Common
Long ID (LID):	0x0000851F
Data type:	PT_UNICODE
Area:	Reminder

This property controls what filename should be played by the Outlook client when the reminder for the mail item is triggered. This is of course particularly interesting as it implies that the property accepts a filename, which of could potentially be a UNC path in order to trigger the NTLM authentication.

Following further analysis of the available properties, we also note the PidLidReminderOverride property which is described as follows:

Applies to: Outlook 2013 | Outlook 2016

Specifies whether the client should respect the values of the **dispidReminderPlaySound** (PidLidReminderPlaySound) and **dispidReminderFileParam** (PidLidReminderFileParameter) properties.

Property	Value
Associated properties:	dispidReminderOverride
Property set:	PSETID_Common
Long ID (LID):	0x0000851C
Data type:	PT_BOOLEAN
Area:	Reminder

With this in mind, we should likely set the *PidLidReminderOverride* property in order to trigger Outlook to parse our malicious UNC inside *PidLidReminderFileParameter*.

Let's begin to build an exploit....

The first step to exploit this issue is to create an Outlook MSG file; these files are compound files in CFB format. To speed up the generation of these files, I leveraged the .NET MsgKit library.

Reviewing the MsgKit library, we find that the *Appointment* class defines a number of properties to add to the mail item before the MSG file is saved:

```
#region WriteToStorage
/// <summary>
///     Writes all the properties that are part of the <see cref="Appointment"/> object either as <see cref="CFStorage"/>'s
///     or <see cref="CFStream"/>'s to the <see cref="CompoundFile.RootStorage"/>
/// </summary>
2 references | 0 changes | 0 authors, 0 changes
private new void WriteToStorage()
{
    Class = MessageClass.IPM_Appointment;
    NamedProperties.AddProperty(NamedPropertyTags.PidLidLocation, Location);
    NamedProperties.AddProperty(NamedPropertyTags.PidLidAppointmentStartWhole, MeetingStart);
    NamedProperties.AddProperty(NamedPropertyTags.PidLidAppointmentEndWhole, MeetingEnd);
    NamedProperties.AddProperty(NamedPropertyTags.PidLidMeetingType, MeetingType.mtgRequest);
    NamedProperties.AddProperty(NamedPropertyTags.PidLidAppointmentSubType, AllDay);
    NamedProperties.AddProperty(NamedPropertyTags.PidLidAppointmentStateFlags, AppointmentState.asfMeeting);
    // Added for exploit
    NamedProperties.AddProperty(NamedPropertyTags.PidLidReminderFileParameter, PidLidReminderFileParameter);
    NamedProperties.AddProperty(NamedPropertyTags.PidLidReminderOverride, PidLidReminderOverride);
}
#endregion
```

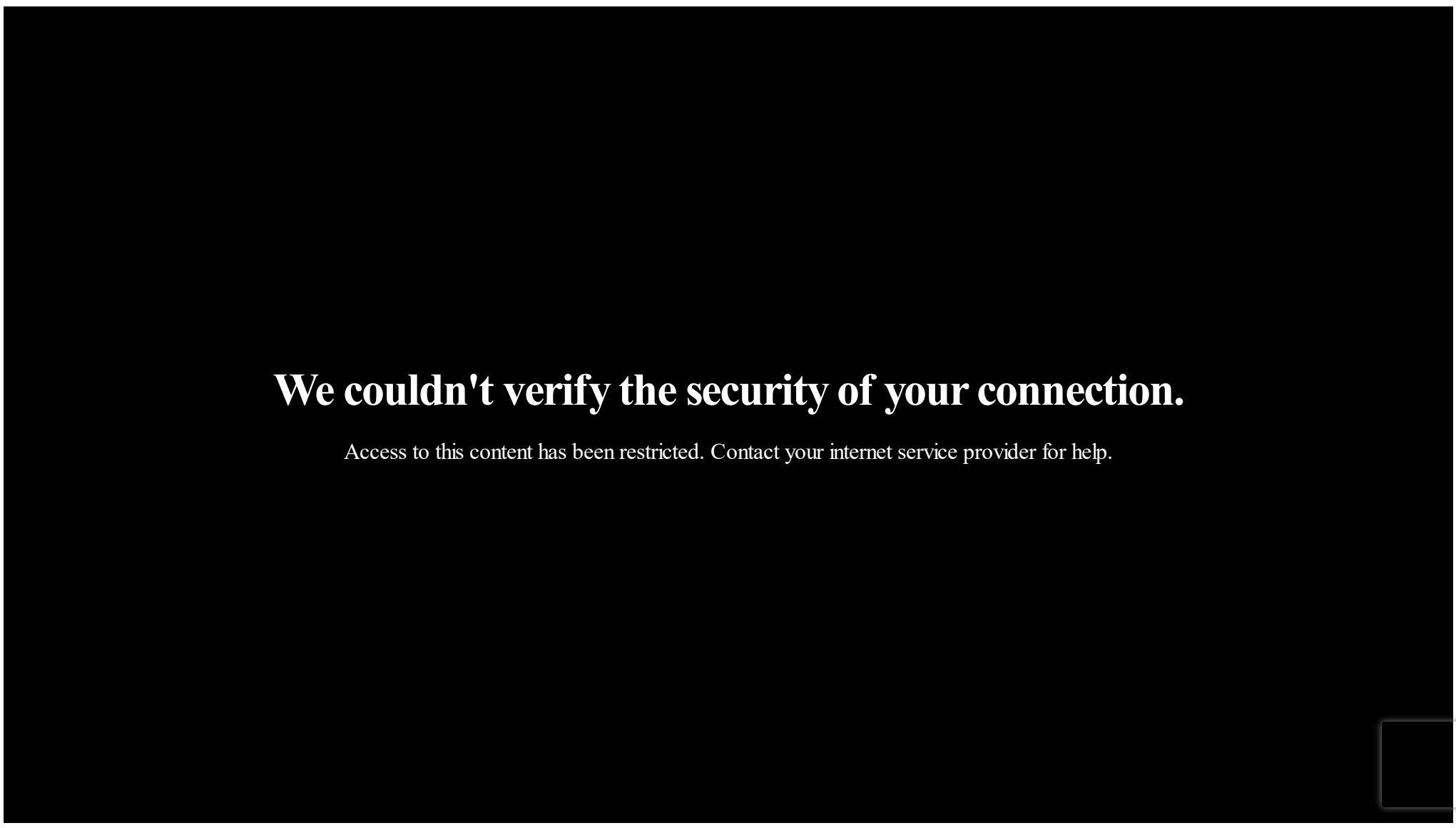
To create our malicious calendar appointment, I extended the Appointment class to add our required *PidLidReminderOverride* and *PidLidReminderFileParameter* properties, as shown above.

From that point, we simply need to create a new appointment and save it, before sending to our victim:

This vulnerability is particularly interesting as it will trigger NTLM authentication to an IP address (i.e. a system outside of the Trusted Intranet Zone or Trusted Sites) and this occurs immediately on opening the e-mail, irrespective of whether the user has selected the option to load remote images or not.

This one is worth patching as a priority as its incredibly easy to exploit and will no doubt be adopted by adversaries fast.

Here's a demonstration of our exploit which will relay the incoming request to LDAP to obtain a shadow credential:



This blog post was written by Dominic Chell.

WRITTEN BY
MDSec Research