CYBERSECURITY BOOTCAMP

GDG Algiers

Women Techmakers Algiers

Reverse Engineering 101

Table of Contents

1. What is Reverse Engineering
    a. Introduction
    b. Why reverse engineering
2. How to perform reverse engineering
    a. Static analysis
    b. Dynamic analysis
3. Mitigations and obstacles

What is reverse engineering

Reverse Engineering is the process of analysing a product to understand how it works

Why reverse engineering

- Modifying software/hardware
- Finding bugs
- Finding security issues
- Analyzing malware
- Finding hidden features

How to perform reverse engineering

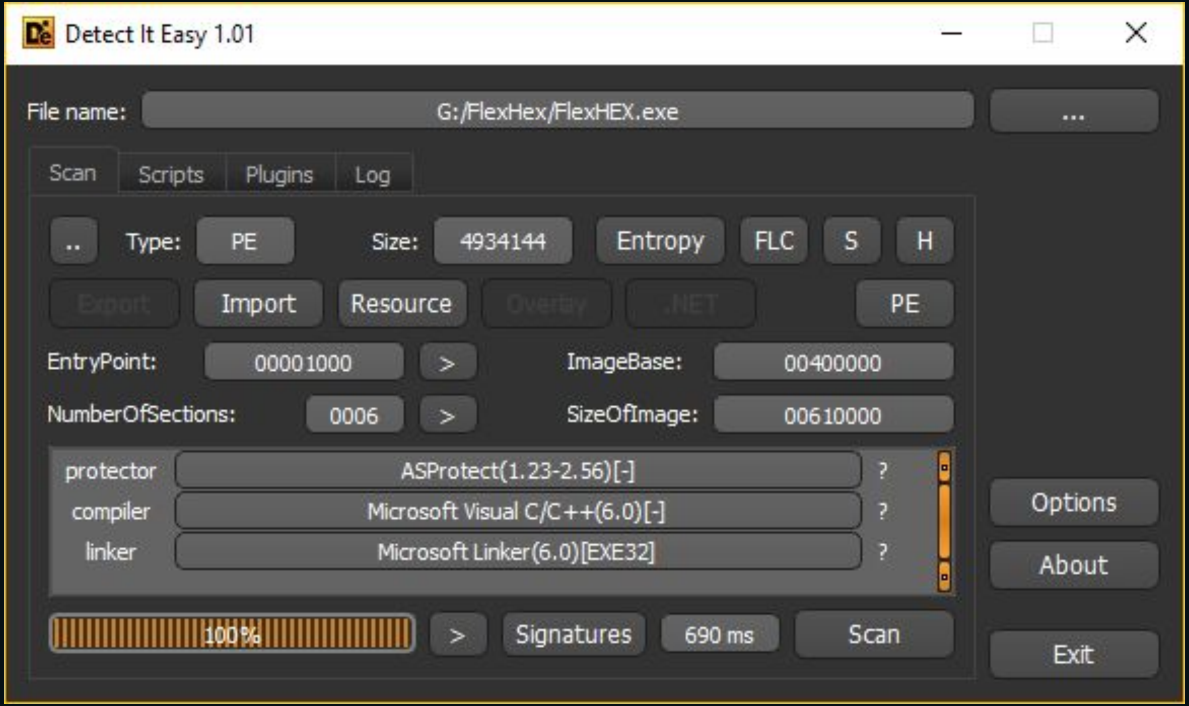
	Static analysis	Dynamic analysis
Identify	✓	✗
Execute	✗	✓

Static Analysis

The different techniques for analyzing the sample program without executing it

- Looking up VirusTotal to check if it’s a known malware
- Looking at the strings inside the file
- Trying to know the runtime, compiler used (as well as any other packers)
- Trying to look at the code disassembly / decompilation

Detect it Easy on Windows



File name: G:/FlexHex/FlexHEX.exe

Type: PE

Size: 4934144

EntryPoint: 00001000

ImageBase: 00400000

NumberOfSections: 0006

SizeOfImage: 00610000

protector ASProtect(1.23-2.56)[-]

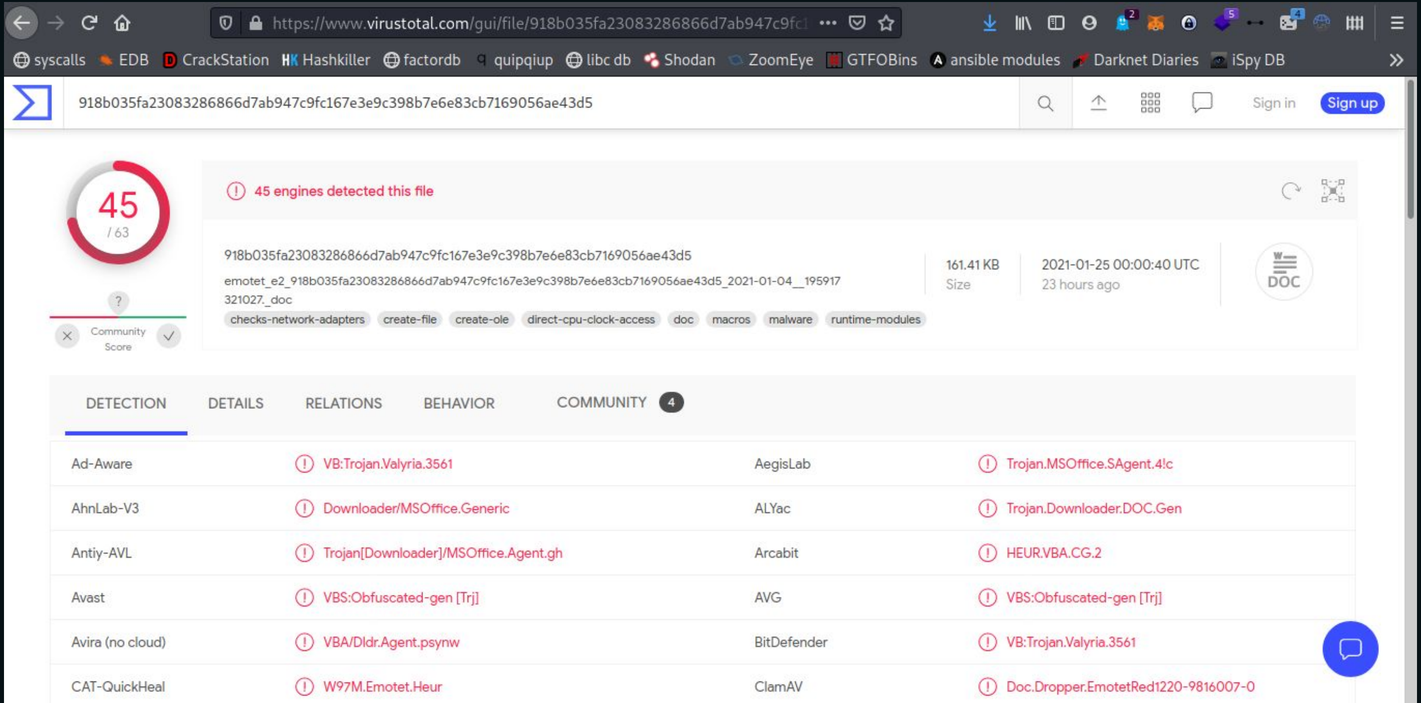
compiler Microsoft Visual C/C++(6.0)[-]

linker Microsoft Linker(6.0)[EXE32]

100%

690 ms

VirusTotal reporting malware



918b035fa23083286866d7ab947c9fc167e3e9c398b7e6e83cb7169056ae43d5

45 / 63

? Community Score

45 engines detected this file

918b035fa23083286866d7ab947c9fc167e3e9c398b7e6e83cb7169056ae43d5
emotet_e2_918b035fa23083286866d7ab947c9fc167e3e9c398b7e6e83cb7169056ae43d5_2021-01-04__195917321027._doc

checks-network-adapters create-file create-ole direct-cpu-clock-access doc macros malware runtime-modules

161.41 KB
Size

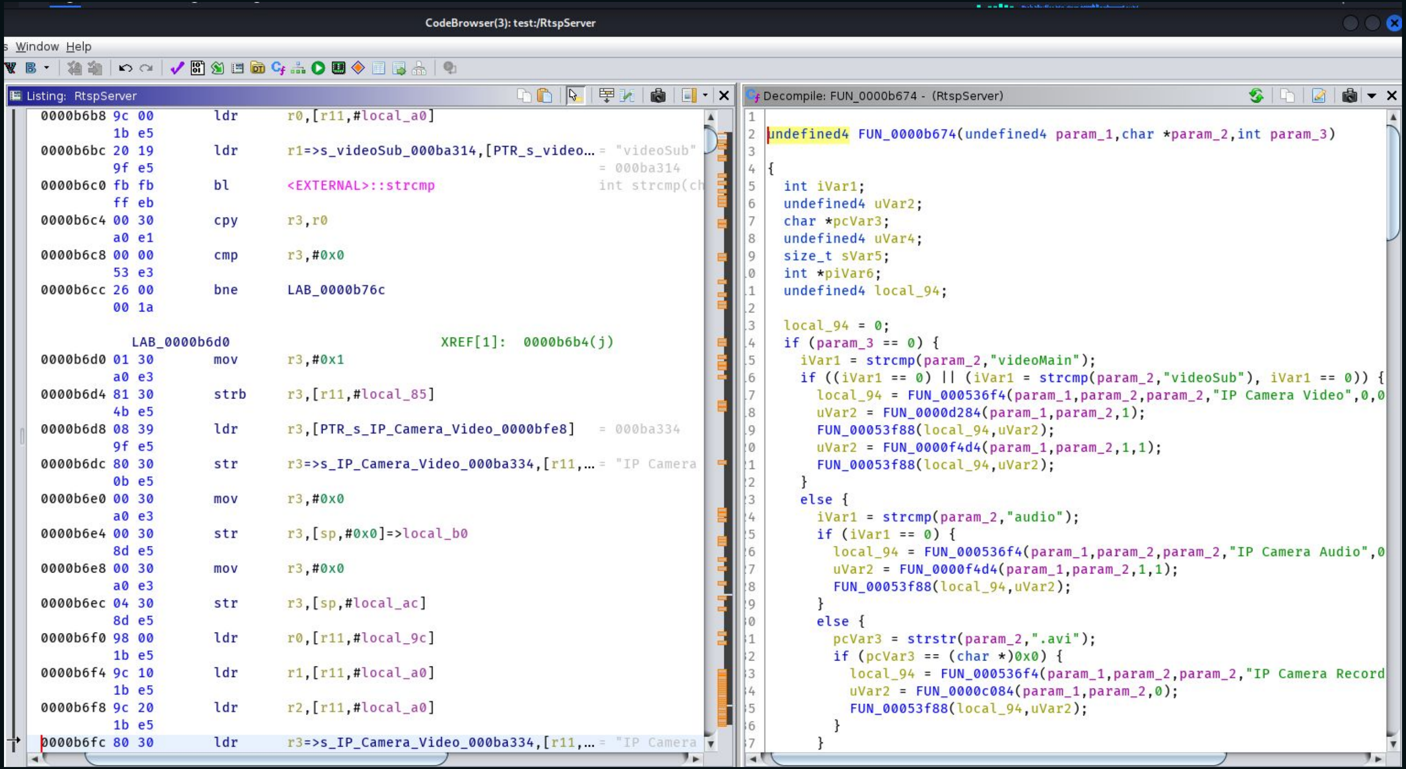
2021-01-25 00:00:40 UTC
23 hours ago

DOC

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 4

Ad-Aware	VB:Trojan.Valyria.3561	AegisLab	Trojan.MSOffice.SAgent.4!c
AhnLab-V3	Downloader/MSOffice.Generic	ALYac	Trojan.Downloader.DOC.Gen
Antiy-AVL	Trojan[Downloader]/MSOffice.Agent.gh	Arcabit	HEUR.VBA.CG.2
Avast	VBS:Obfuscated-gen [Trj]	AVG	VBS:Obfuscated-gen [Trj]
Avira (no cloud)	VBA/Dldr.Agent.psynw	BitDefender	VB:Trojan.Valyria.3561
CAT-QuickHeal	W97M.Emotet.Heur	ClamAV	Doc.Dropper.EmotetRed1220-9816007-0

Ghidra CodeBrowser view


CodeBrowser(3): test:/RtspServer
Window
Help
Listing: RtspServer
Decompile: FUN_0000b674 - (RtspServer)

```
0000b6b8 9c 00        ldr       r0,[r11,#local_a0]
         1b e5
0000b6bc 20 19        ldr       r1=>s_videoSub_000ba314,[PTR_s_video… = "videoSub"
         9f e5                                                        = 000ba314
0000b6c0 fb fb        bl        <EXTERNAL>::strcmp                     int strcmp(ch
         ff eb
0000b6c4 00 30        cpy       r3,r0
         a0 e1
0000b6c8 00 00        cmp       r3,#0x0
         53 e3
0000b6cc 26 00        bne       LAB_0000b76c
         00 1a

              LAB_0000b6d0                        XREF[1]:   0000b6b4(j)
0000b6d0 01 30        mov       r3,#0x1
         a0 e3
0000b6d4 81 30        strb      r3,[r11,#local_85]
         4b e5
0000b6d8 08 39        ldr       r3,[PTR_s_IP_Camera_Video_0000bfe8]   = 000ba334
         9f e5
0000b6dc 80 30        str       r3=>s_IP_Camera_Video_000ba334,[r11,… = "IP Camera
         0b e5
0000b6e0 00 30        mov       r3,#0x0
         a0 e3
0000b6e4 00 30        str       r3,[sp,#0x0]=>local_b0
         8d e5
0000b6e8 00 30        mov       r3,#0x0
         a0 e3
0000b6ec 04 30        str       r3,[sp,#local_ac]
         8d e5
0000b6f0 98 00        ldr       r0,[r11,#local_9c]
         1b e5
0000b6f4 9c 10        ldr       r1,[r11,#local_a0]
         1b e5
0000b6f8 9c 20        ldr       r2,[r11,#local_a0]
         1b e5
0000b6fc 80 30        ldr       r3=>s_IP_Camera_Video_000ba334,[r11,… = "IP Camera
```

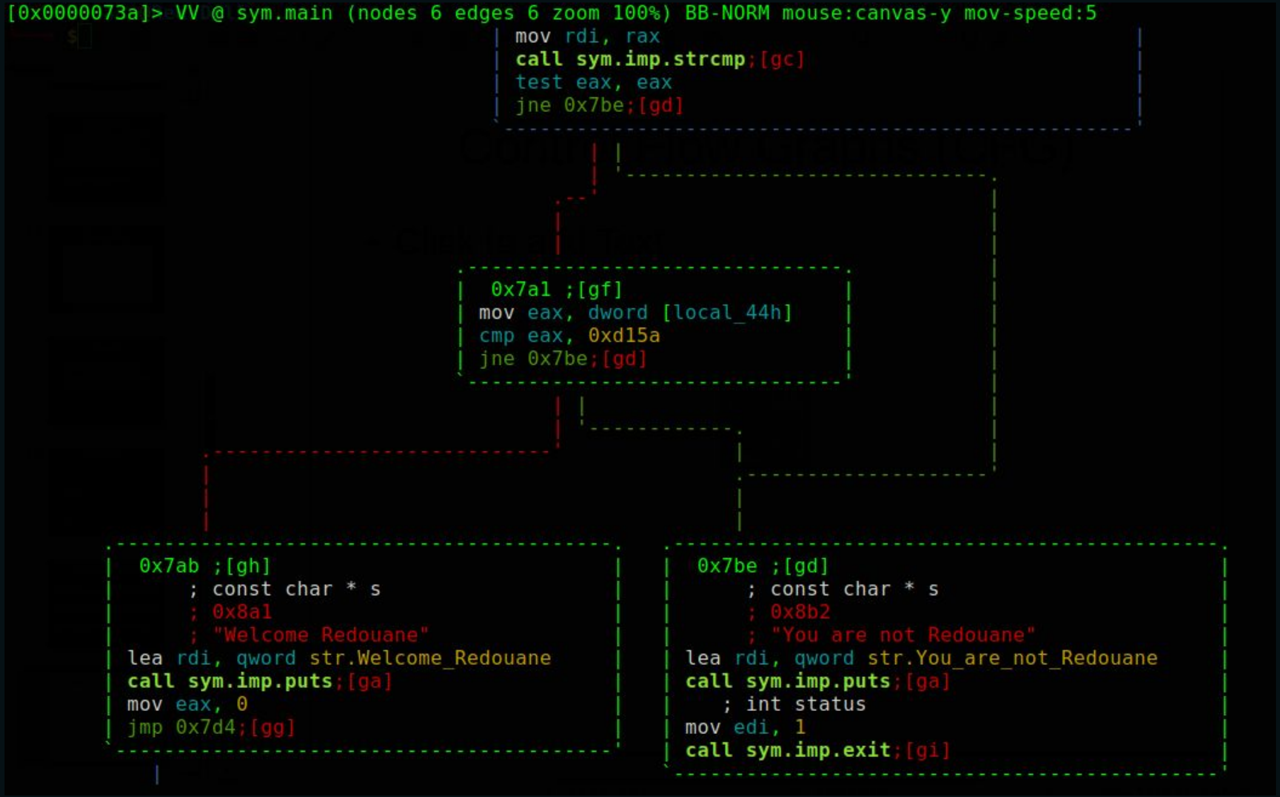
```c

undefined4 FUN_0000b674(undefined4 param_1,char *param_2,int param_3)

{
  int iVar1;
  undefined4 uVar2;
  char *pcVar3;
  undefined4 uVar4;
  size_t sVar5;
  int *piVar6;
  undefined4 local_94;

  local_94 = 0;
  if (param_3 == 0) {
    iVar1 = strcmp(param_2,"videoMain");
    if ((iVar1 == 0) || (iVar1 = strcmp(param_2,"videoSub"), iVar1 == 0)) {
      local_94 = FUN_000536f4(param_1,param_2,param_2,"IP Camera Video",0,0
      uVar2 = FUN_0000d284(param_1,param_2,1);
      FUN_00053f88(local_94,uVar2);
      uVar2 = FUN_0000f4d4(param_1,param_2,1,1);
      FUN_00053f88(local_94,uVar2);
    }
    else {
      iVar1 = strcmp(param_2,"audio");
      if (iVar1 == 0) {
        local_94 = FUN_000536f4(param_1,param_2,param_2,"IP Camera Audio",0
        uVar2 = FUN_0000f4d4(param_1,param_2,1,1);
        FUN_00053f88(local_94,uVar2);
      }
      else {
        pcVar3 = strstr(param_2,".avi");
        if (pcVar3 == (char *)0x0) {
          local_94 = FUN_000536f4(param_1,param_2,param_2,"IP Camera Record
          uVar2 = FUN_0000c084(param_1,param_2,0);
          FUN_00053f88(local_94,uVar2);
        }
      }
```

Control flow graph on Radare2

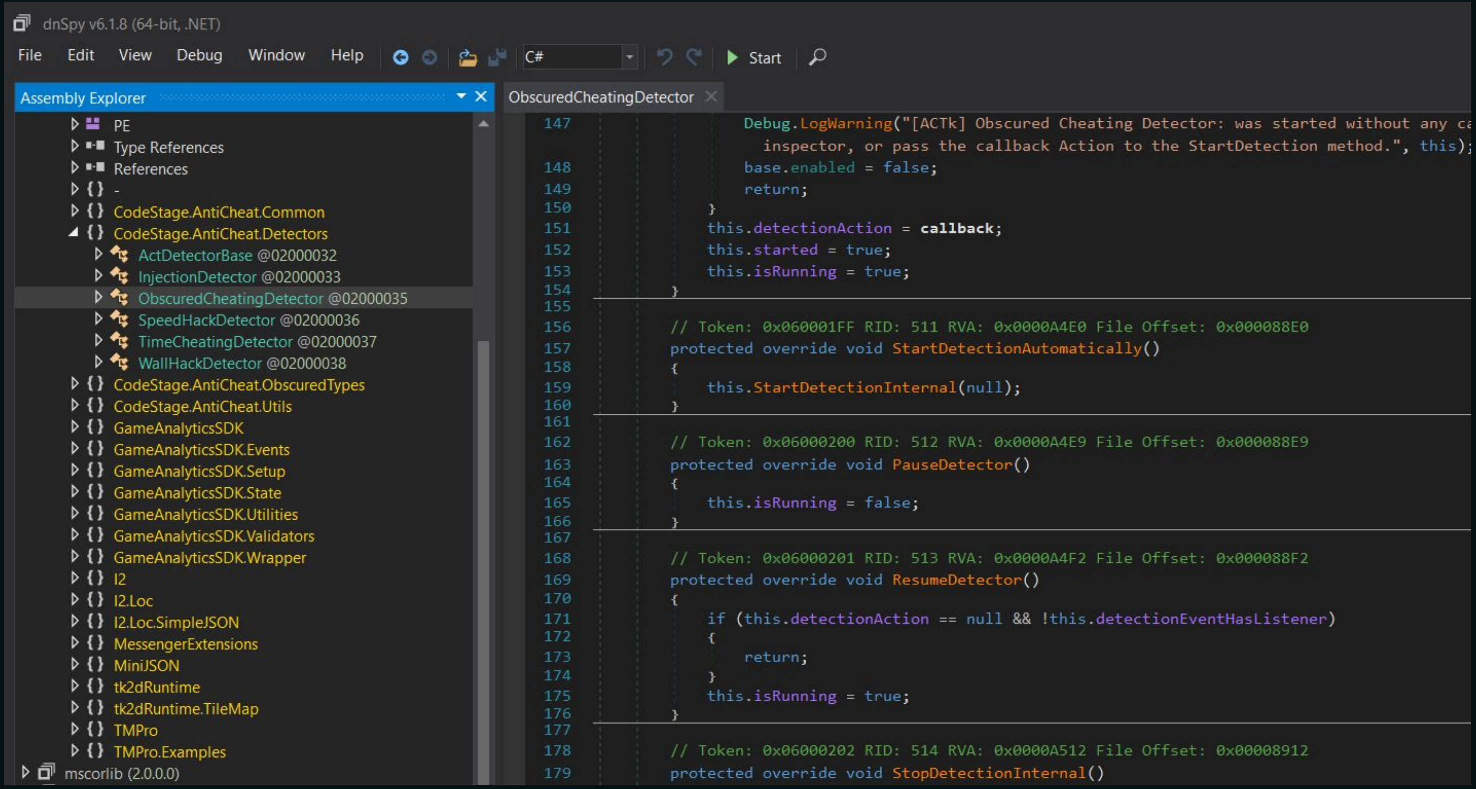


```
[0x0000073a]> VV @ sym.main (nodes 6 edges 6 zoom 100%) BB-NORM mouse:canvas-y mov-speed:5
                                      | mov rdi, rax                                   |
                                      | call sym.imp.strcmp;[gc]                       |
                                      | test eax, eax                                  |
                                      | jne 0x7be;[gd]                                 |
                                      `------------------------------------------------'

                            .--------------------------------.
                            |  0x7a1 ;[gf]                   |
                            | mov eax, dword [local_44h]     |
                            | cmp eax, 0xd15a                |
                            | jne 0x7be;[gd]                 |
                            `--------------------------------'

   .--------------------------------------.   .------------------------------------------.
   |  0x7ab ;[gh]                         |   |  0x7be ;[gd]                             |
   |       ; const char * s               |   |       ; const char * s                   |
   |       ; 0x8a1                        |   |       ; 0x8b2                            |
   |       ; "Welcome Redouane"           |   |       ; "You are not Redouane"           |
   | lea rdi, qword str.Welcome_Redouane  |   | lea rdi, qword str.You_are_not_Redouane  |
   | call sym.imp.puts;[ga]               |   | call sym.imp.puts;[ga]                   |
   | mov eax, 0                           |   |     ; int status                         |
   | jmp 0x7d4;[gg]                       |   | mov edi, 1                               |
   `--------------------------------------'   | call sym.imp.exit;[gi]                   |
                                              `------------------------------------------'
```

DnSpy decompiling a game anticheat



- PE
- Type References
- References
- { } -
- { } CodeStage.AntiCheat.Common
- { } CodeStage.AntiCheat.Detectors
  - ActDetectorBase @02000032
  - InjectionDetector @02000033
  - ObscuredCheatingDetector @02000035
  - SpeedHackDetector @02000036
  - TimeCheatingDetector @02000037
  - WallHackDetector @02000038
- { } CodeStage.AntiCheat.ObscuredTypes
- { } CodeStage.AntiCheat.Utils
- { } GameAnalyticsSDK
- { } GameAnalyticsSDK.Events
- { } GameAnalyticsSDK.Setup
- { } GameAnalyticsSDK.State
- { } GameAnalyticsSDK.Utilities
- { } GameAnalyticsSDK.Validators
- { } GameAnalyticsSDK.Wrapper
- { } I2
- { } I2.Loc
- { } I2.Loc.SimpleJSON
- { } MessengerExtensions
- { } MiniJSON
- { } tk2dRuntime
- { } tk2dRuntime.TileMap
- { } TMPro
- { } TMPro.Examples

mscorlib (2.0.0.0)

```
            Debug.LogWarning("[ACTk] Obscured Cheating Detector: was started without any ca
              inspector, or pass the callback Action to the StartDetection method.", this);
            base.enabled = false;
            return;
        }
        this.detectionAction = callback;
        this.started = true;
        this.isRunning = true;
    }

    // Token: 0x060001FF RID: 511 RVA: 0x0000A4E0 File Offset: 0x000088E0
    protected override void StartDetectionAutomatically()
    {
        this.StartDetectionInternal(null);
    }

    // Token: 0x06000200 RID: 512 RVA: 0x0000A4E9 File Offset: 0x000088E9
    protected override void PauseDetector()
    {
        this.isRunning = false;
    }

    // Token: 0x06000201 RID: 513 RVA: 0x0000A4F2 File Offset: 0x000088F2
    protected override void ResumeDetector()
    {
        if (this.detectionAction == null && !this.detectionEventHasListener)
        {
            return;
        }
        this.isRunning = true;
    }

    // Token: 0x06000202 RID: 514 RVA: 0x0000A512 File Offset: 0x00008912
    protected override void StopDetectionInternal()
```

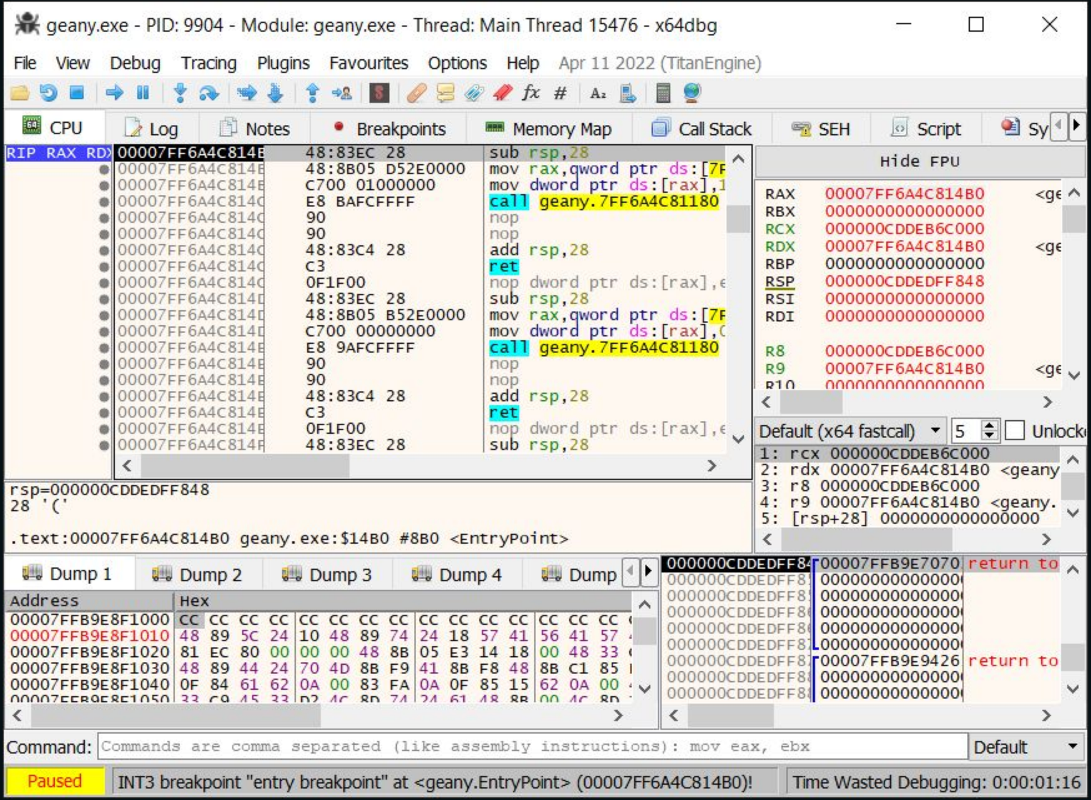
Dynamic Analysis

The techniques for analyzing the sample while it runs

NOTE: HAS TO BE DONE IN AN ISOLATED ENVIRONMENT (VM)

- Intercepting and analyzing network traffic generated by the application
- Analyzing the API calls made by the application
- Debugging the sample
- Intercepting the decrypted data/code at runtime
- Performing instrumentation (injecting code into the application)
- Faking network traffic by setting up a fake dns / server (dnschef, inetsim)

A debugging session in x64dbg


geany.exe - PID: 9904 - Module: geany.exe - Thread: Main Thread 15476 - x64dbg
File View Debug Tracing Plugins Favourites Options Help Apr 11 2022 (TitanEngine)
CPU
Log
Notes
Breakpoints
Memory Map
Call Stack
SEH
Script
Sy
Hide FPU
Default (x64 fastcall)
5
Unlock
Dump 1
Dump 2
Dump 3
Dump 4
Dump
Command:
Default

```
RIP RAX RDX 00007FF6A4C814E   48:83EC 28          sub rsp,28
            00007FF6A4C814E   48:8B05 D52E0000    mov rax,qword ptr ds:[7F
            00007FF6A4C814E   C700 01000000       mov dword ptr ds:[rax],1
            00007FF6A4C814C   E8 BAFCFFFF         call geany.7FF6A4C81180
            00007FF6A4C814C   90                  nop
            00007FF6A4C814C   90                  nop
            00007FF6A4C814C   48:83C4 28          add rsp,28
            00007FF6A4C814C   C3                  ret
            00007FF6A4C814C   0F1F00              nop dword ptr ds:[rax],e
            00007FF6A4C814D   48:83EC 28          sub rsp,28
            00007FF6A4C814D   48:8B05 B52E0000    mov rax,qword ptr ds:[7F
            00007FF6A4C814D   C700 00000000       mov dword ptr ds:[rax],0
            00007FF6A4C814E   E8 9AFCFFFF         call geany.7FF6A4C81180
            00007FF6A4C814E   90                  nop
            00007FF6A4C814E   90                  nop
            00007FF6A4C814E   48:83C4 28          add rsp,28
            00007FF6A4C814E   C3                  ret
            00007FF6A4C814E   0F1F00              nop dword ptr ds:[rax],e
            00007FF6A4C814F   48:83EC 28          sub rsp,28
```

```
rsp=000000CDDEDFF848
28 '('

.text:00007FF6A4C814B0 geany.exe:$14B0 #8B0 <EntryPoint>
```

```
RAX  00007FF6A4C814B0    <ge
RBX  0000000000000000
RCX  000000CDDEB6C000
RDX  00007FF6A4C814B0    <ge
RBP  0000000000000000
RSP  000000CDDEDFF848
RSI  0000000000000000
RDI  0000000000000000

R8   000000CDDEB6C000
R9   00007FF6A4C814B0    <ge
R10  0000000000000000
```

```
1: rcx 000000CDDEB6C000
2: rdx 00007FF6A4C814B0 <geany
3: r8 000000CDDEB6C000
4: r9 00007FF6A4C814B0 <geany.
5: [rsp+28] 0000000000000000
```

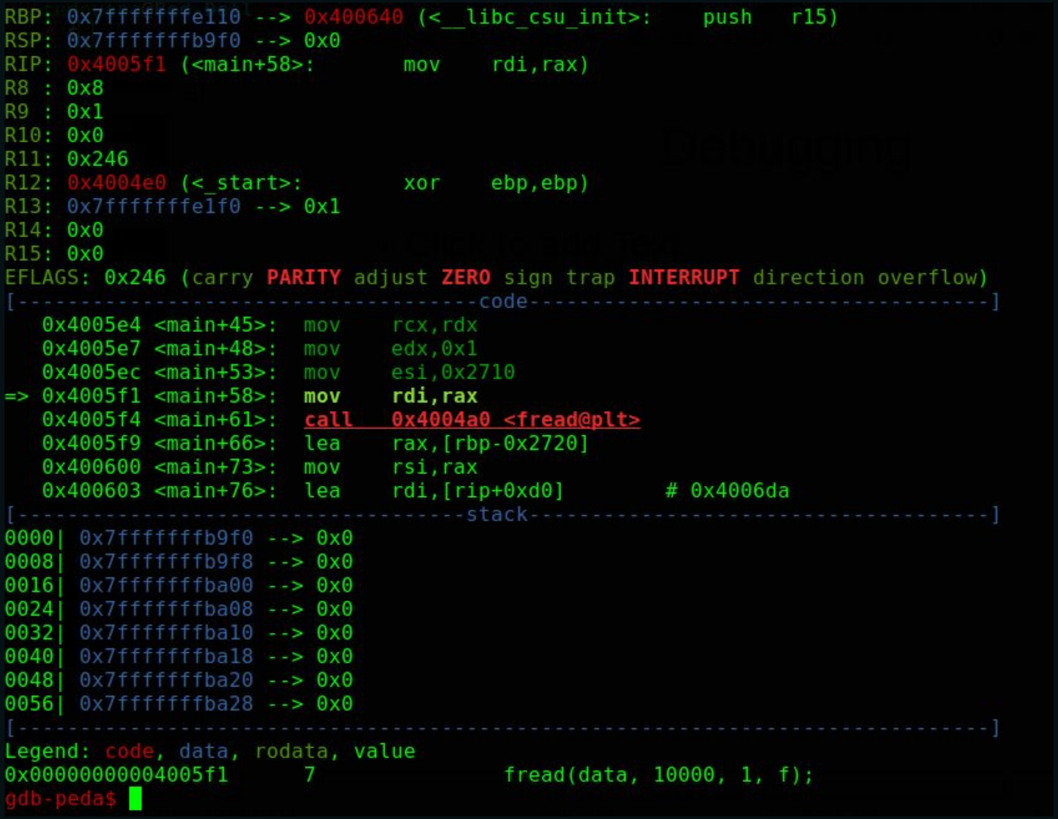
```
Address           Hex
00007FFB9E8F1000  CC CC CC CC  CC CC CC CC  CC CC CC CC  CC CC CC
00007FFB9E8F1010  48 89 5C 24  10 48 89 74  24 18 57 41  56 41 57
00007FFB9E8F1020  81 EC 80 00  00 00 48 8B  05 E3 14 18  00 48 33
00007FFB9E8F1030  48 89 44 24  70 4D 8B F9  41 8B F8 48  8B C1 85
00007FFB9E8F1040  0F 84 61 62  0A 00 83 FA  0A 0F 85 15  62 0A 00
00007FFB9E8F1050  33 C9 45 33  D2 4C 8D 74  24 61 48 8B  00 4C 8D
```

```
000000CDDEDFF84  00007FFB9E7070  return to
000000CDDEDFF8   00000000000000
000000CDDEDFF8   00000000000000
000000CDDEDFF8   00000000000000
000000CDDEDFF8   00000000000000
000000CDDEDFF8   00000000000000
000000CDDEDFF8   00007FFB9E9426  return to
000000CDDEDFF8   00000000000000
000000CDDEDFF8   00000000000000
```

Commands are comma separated (like assembly instructions): mov eax, ebx

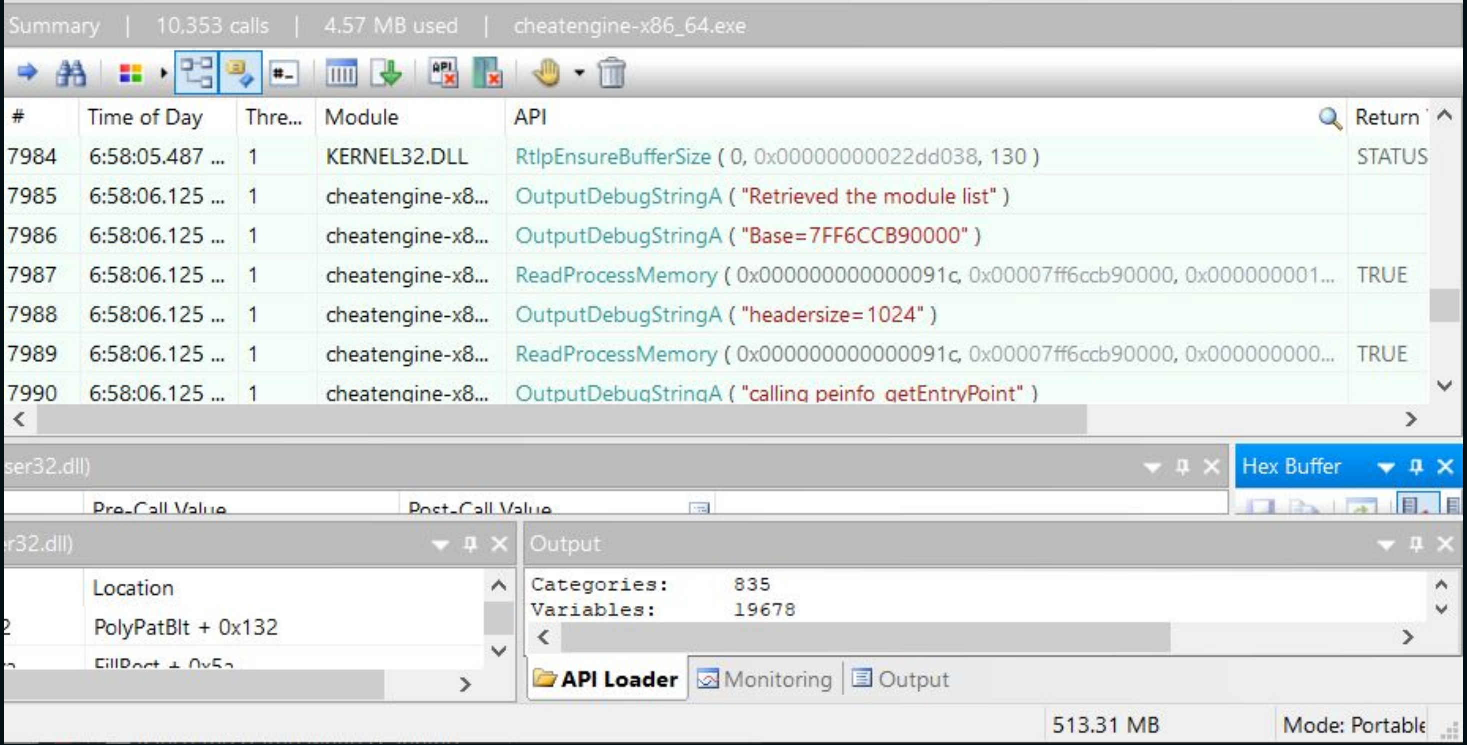
Paused | INT3 breakpoint "entry breakpoint" at <geany.EntryPoint> (00007FF6A4C814B0)! | Time Wasted Debugging: 0:00:01:16

A debugging session in gdb



```
RBP: 0x7fffffffe110 --> 0x400640 (<__libc_csu_init>:    push   r15)
RSP: 0x7fffffffb9f0 --> 0x0
RIP: 0x4005f1 (<main+58>:       mov    rdi,rax)
R8 : 0x8
R9 : 0x1
R10: 0x0
R11: 0x246
R12: 0x4004e0 (<_start>:        xor    ebp,ebp)
R13: 0x7fffffffe1f0 --> 0x1
R14: 0x0
R15: 0x0
EFLAGS: 0x246 (carry PARITY adjust ZERO sign trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x4005e4 <main+45>:  mov    rcx,rdx
   0x4005e7 <main+48>:  mov    edx,0x1
   0x4005ec <main+53>:  mov    esi,0x2710
=> 0x4005f1 <main+58>:  mov    rdi,rax
   0x4005f4 <main+61>:  call   0x4004a0 <fread@plt>
   0x4005f9 <main+66>:  lea    rax,[rbp-0x2720]
   0x400600 <main+73>:  mov    rsi,rax
   0x400603 <main+76>:  lea    rdi,[rip+0xd0]        # 0x4006da
[------------------------------------stack-------------------------------------]
0000| 0x7fffffffb9f0 --> 0x0
0008| 0x7fffffffb9f8 --> 0x0
0016| 0x7fffffffba00 --> 0x0
0024| 0x7fffffffba08 --> 0x0
0032| 0x7fffffffba10 --> 0x0
0040| 0x7fffffffba18 --> 0x0
0048| 0x7fffffffba20 --> 0x0
0056| 0x7fffffffba28 --> 0x0
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value
0x00000000004005f1      7               fread(data, 10000, 1, f);
gdb-peda$ 
```

API montior on Windows



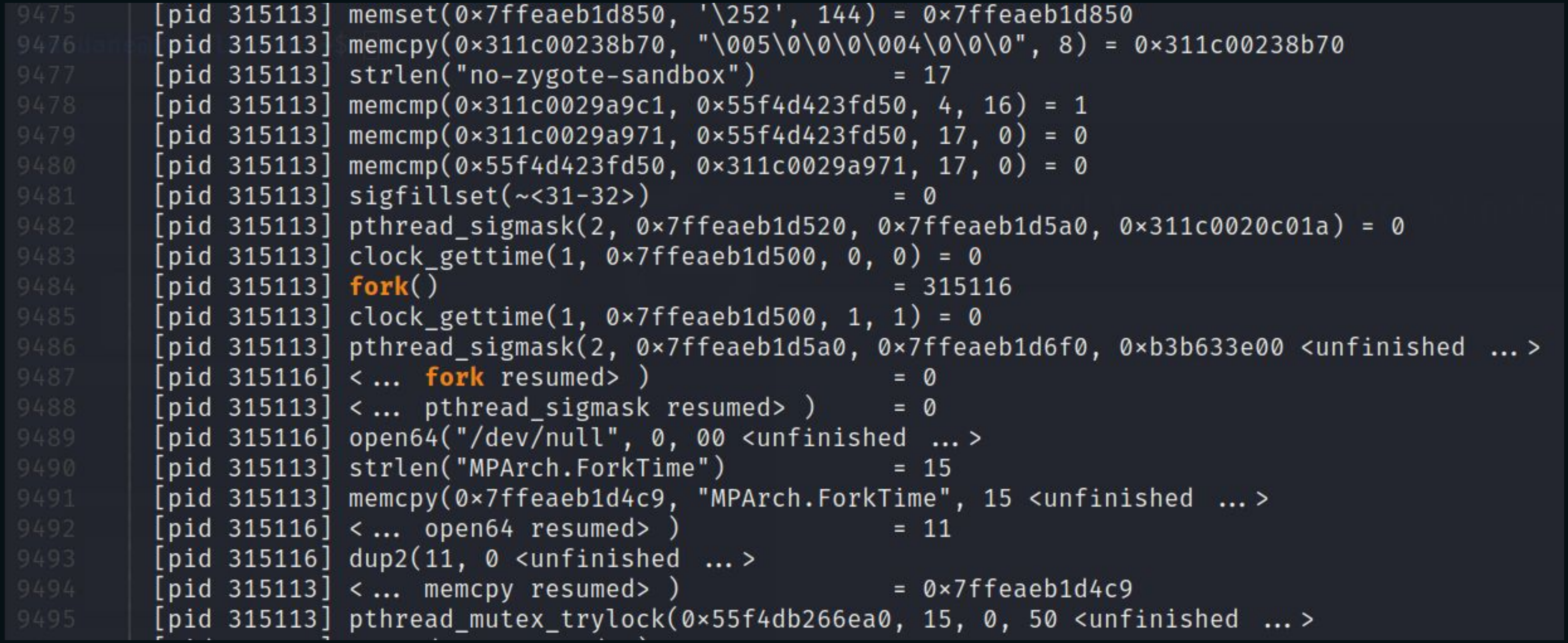
Summary | 10,353 calls | 4.57 MB used | cheatengine-x86_64.exe

#	Time of Day	Thre...	Module	API	Return
7984	6:58:05.487 ...	1	KERNEL32.DLL	RtlpEnsureBufferSize (0, 0x00000000022dd038, 130)	STATUS
7985	6:58:06.125 ...	1	cheatengine-x8...	OutputDebugStringA ("Retrieved the module list")	
7986	6:58:06.125 ...	1	cheatengine-x8...	OutputDebugStringA ("Base=7FF6CCB90000")	
7987	6:58:06.125 ...	1	cheatengine-x8...	ReadProcessMemory (0x000000000000091c, 0x00007ff6ccb90000, 0x000000001...	TRUE
7988	6:58:06.125 ...	1	cheatengine-x8...	OutputDebugStringA ("headersize=1024")	
7989	6:58:06.125 ...	1	cheatengine-x8...	ReadProcessMemory (0x000000000000091c, 0x00007ff6ccb90000, 0x000000000...	TRUE
7990	6:58:06.125 ...	1	cheatengine-x8...	OutputDebugStringA ("calling peinfo_getEntryPoint")	

	Location
2	PolyPatBlt + 0x132
	FillRect + 0x5a

```
Categories:     835
Variables:      19678
```

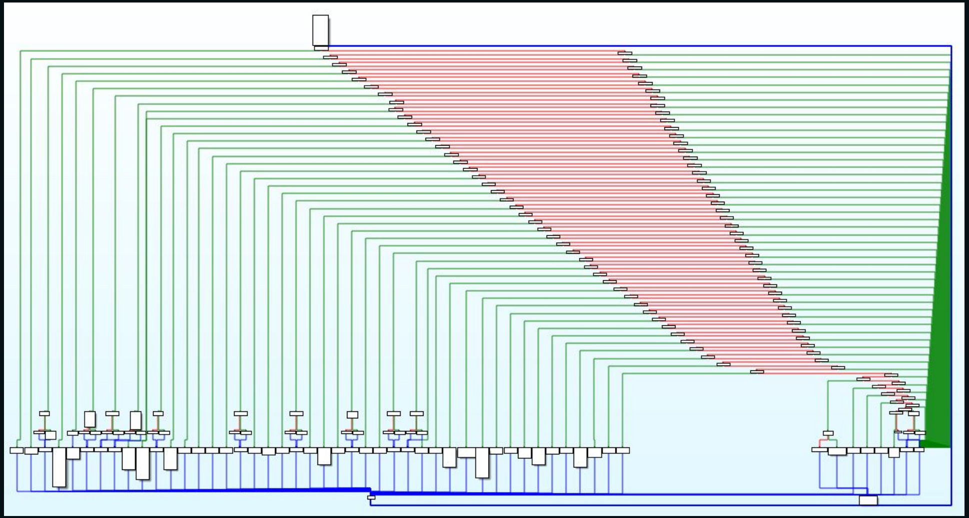
ltrace log on Linux



```
[pid 315113] memset(0x7ffeaeb1d850, '\252', 144) = 0x7ffeaeb1d850
[pid 315113] memcpy(0x311c00238b70, "\005\0\0\0\004\0\0\0", 8) = 0x311c00238b70
[pid 315113] strlen("no-zygote-sandbox")         = 17
[pid 315113] memcmp(0x311c0029a9c1, 0x55f4d423fd50, 4, 16) = 1
[pid 315113] memcmp(0x311c0029a971, 0x55f4d423fd50, 17, 0) = 0
[pid 315113] memcmp(0x55f4d423fd50, 0x311c0029a971, 17, 0) = 0
[pid 315113] sigfillset(~<31-32>)                = 0
[pid 315113] pthread_sigmask(2, 0x7ffeaeb1d520, 0x7ffeaeb1d5a0, 0x311c0020c01a) = 0
[pid 315113] clock_gettime(1, 0x7ffeaeb1d500, 0, 0) = 0
[pid 315113] fork()                              = 315116
[pid 315113] clock_gettime(1, 0x7ffeaeb1d500, 1, 1) = 0
[pid 315113] pthread_sigmask(2, 0x7ffeaeb1d5a0, 0x7ffeaeb1d6f0, 0xb3b633e00 <unfinished ...>
[pid 315116] <... fork resumed> )                = 0
[pid 315113] <... pthread_sigmask resumed> )     = 0
[pid 315116] open64("/dev/null", 0, 00 <unfinished ...>
[pid 315113] strlen("MPArch.ForkTime")           = 15
[pid 315113] memcpy(0x7ffeaeb1d4c9, "MPArch.ForkTime", 15 <unfinished ...>
[pid 315116] <... open64 resumed> )              = 11
[pid 315116] dup2(11, 0 <unfinished ...>
[pid 315113] <... memcpy resumed> )              = 0x7ffeaeb1d4c9
[pid 315113] pthread_mutex_trylock(0x55f4db266ea0, 15, 0, 50 <unfinished ...>
```

Mitigations and obstacles

Static analysis	Dynamic analysis
obfuscation	anti-debugging
encryption	ssl pinning
API hashing	VM detection
…	…



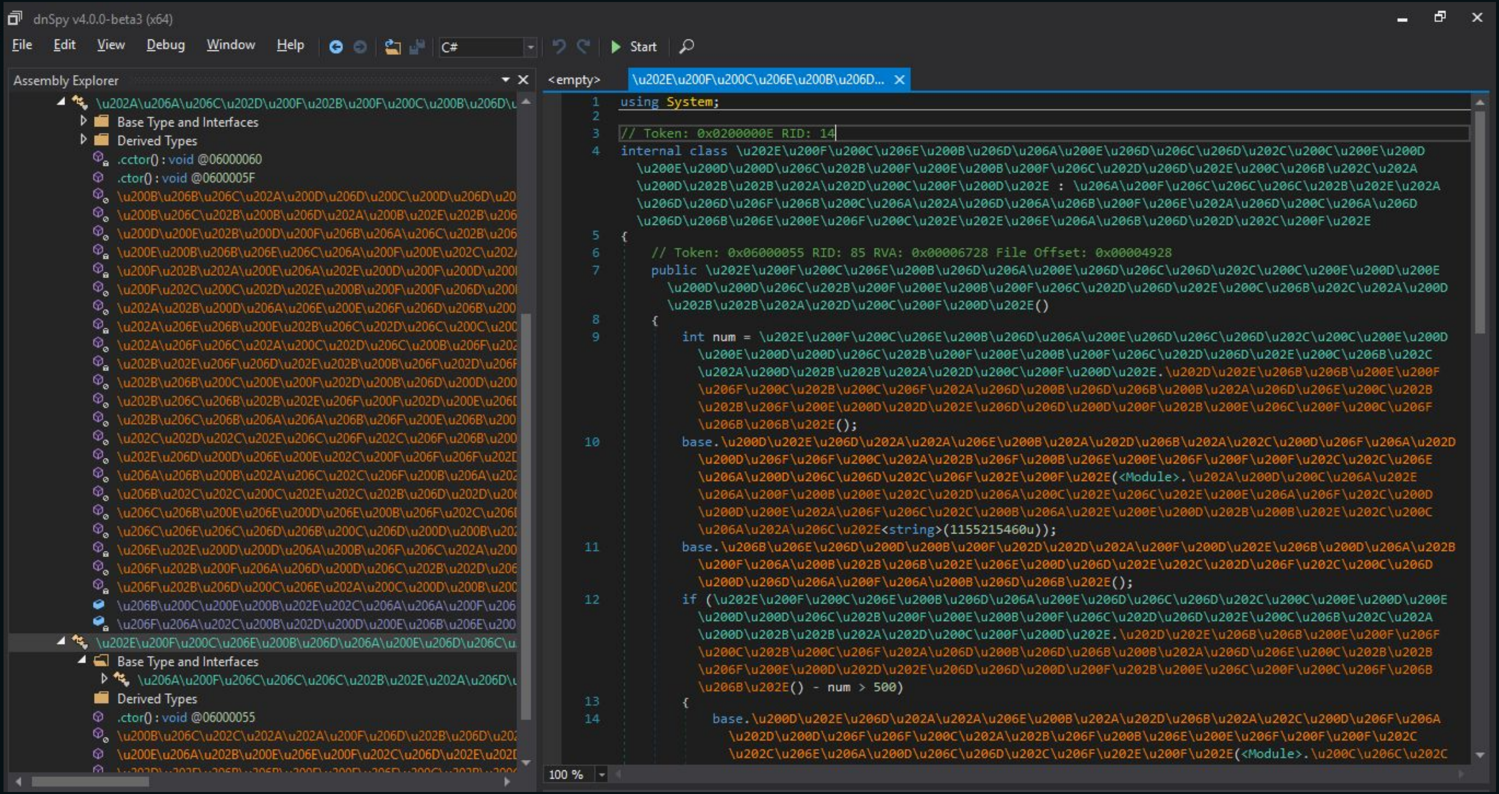
Obfuscated control flow as seen on IDA pro

Obfuscation

Techniques that aim to make it more difficult to read the code

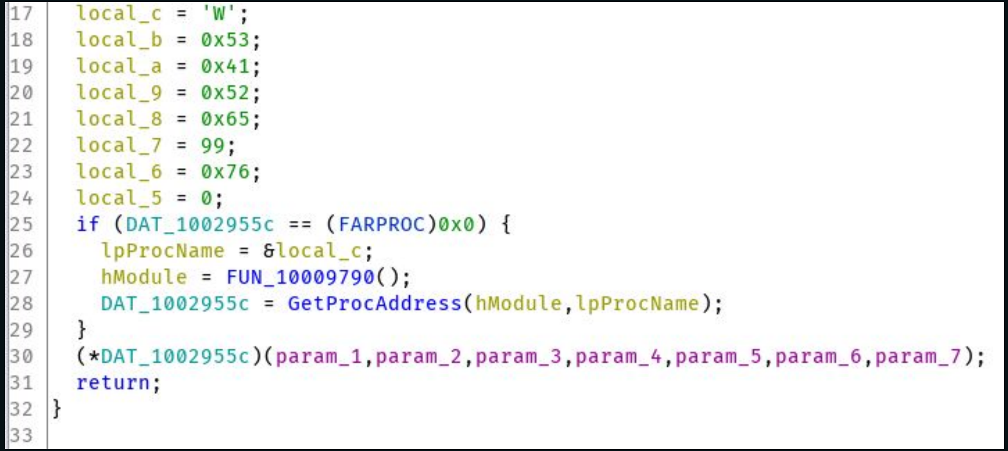
⇒ just requires more effort and understanding from the reverser

Obfuscated C# program loaded in dnSpy


dnSpy v4.0.0-beta3 (x64)
File
Edit
View
Debug
Window
Help
C#
Start
Assembly Explorer
\u202A\u206A\u206C\u202D\u200F\u202B\u200F\u200C\u200B\u206D\u
Base Type and Interfaces
Derived Types
.cctor() : void @06000060
.ctor() : void @0600005F
\u200B\u206B\u206C\u202A\u200D\u206D\u200C\u200D\u206D\u20
\u200B\u206C\u202B\u200B\u206D\u202A\u200B\u202E\u202B\u206
\u200D\u200E\u202B\u200D\u200F\u206B\u206A\u206C\u202B\u206
\u200E\u200B\u206B\u206E\u206C\u206A\u200F\u200E\u202C\u202
\u200F\u202B\u202A\u200E\u206A\u202E\u200D\u200F\u200D\u200
\u200F\u202C\u200C\u202D\u202E\u200B\u200F\u200F\u206D\u200
\u202A\u202B\u200D\u206A\u206E\u200E\u206F\u206D\u206B\u200
\u202A\u206E\u206B\u200E\u202B\u206C\u202D\u206C\u200C\u200
\u202A\u206F\u206C\u202A\u200C\u202D\u206C\u200B\u206F\u202
\u202B\u202E\u206F\u206D\u202E\u202B\u200B\u206F\u202D\u206
\u202B\u206B\u200C\u200E\u200F\u202D\u200B\u206D\u200D\u200
\u202B\u206C\u206B\u202B\u202E\u206F\u200F\u202D\u200E\u206
\u202B\u206C\u206B\u206A\u206A\u206B\u206F\u200E\u206B\u200
\u202C\u202D\u202C\u202E\u206C\u206F\u202C\u206F\u206B\u200
\u202E\u206D\u200D\u206E\u200E\u202C\u200F\u206F\u206F\u202
\u206A\u206B\u200B\u202A\u206C\u202C\u206F\u200B\u206A\u202
\u206B\u202C\u202C\u200C\u202E\u202C\u202B\u206D\u202D\u206
\u206C\u206B\u200E\u206E\u200D\u206E\u200B\u206F\u202C\u206
\u206C\u206E\u206C\u206D\u206B\u200C\u206D\u200D\u200B\u202
\u206E\u202E\u200D\u200D\u206A\u200B\u206F\u206C\u202A\u200
\u206F\u202B\u200F\u206A\u206D\u200D\u206C\u202B\u202D\u206
\u206F\u202B\u206D\u200C\u206E\u202A\u200C\u200D\u200B\u200
\u206B\u200C\u200E\u200B\u202E\u202C\u206A\u206A\u200F\u206
\u206F\u206A\u202C\u200B\u202D\u200D\u200E\u206B\u206E\u200
\u202E\u200F\u200C\u206E\u200B\u206D\u206A\u200E\u206D\u206C\u
Base Type and Interfaces
\u206A\u200F\u206C\u206C\u206C\u202B\u202E\u202A\u206D\u
Derived Types
.ctor() : void @06000055
\u200B\u206C\u202C\u202A\u202A\u200F\u206D\u202B\u206D\u202
\u200E\u206A\u202B\u200E\u206E\u200F\u202C\u206D\u202E\u202
<empty>
\u202E\u200F\u200C\u206E\u200B\u206D...
100 %

```
using System;

// Token: 0x0200000E RID: 14
internal class \u202E\u200F\u200C\u206E\u200B\u206D\u206A\u200E\u206D\u206C\u206D\u202C\u200C\u200E\u200D
  \u200E\u200D\u200D\u206C\u202B\u200F\u200E\u200B\u200F\u206C\u202D\u206D\u202E\u200C\u206B\u202C\u202A
  \u200D\u202B\u202B\u202A\u202D\u200C\u200F\u200D\u202E : \u206A\u200F\u206C\u206C\u206C\u202B\u202E\u202A
  \u206D\u206D\u206F\u206B\u200C\u206A\u202A\u206D\u206A\u206B\u200F\u206E\u202A\u206D\u200C\u206A\u206D
  \u206D\u206B\u206E\u200E\u206F\u200C\u202E\u202E\u206E\u206A\u206B\u206D\u202D\u202C\u200F\u202E
{
    // Token: 0x06000055 RID: 85 RVA: 0x00006728 File Offset: 0x00004928
    public \u202E\u200F\u200C\u206E\u200B\u206D\u206A\u200E\u206D\u206C\u206D\u202C\u200C\u200E\u200D\u200E
      \u200D\u200D\u206C\u202B\u200F\u200E\u200B\u200F\u206C\u202D\u206D\u202E\u200C\u206B\u202C\u202A\u200D
      \u202B\u202B\u202A\u202D\u200C\u200F\u200D\u202E()
    {
        int num = \u202E\u200F\u200C\u206E\u200B\u206D\u206A\u200E\u206D\u206C\u206D\u202C\u200C\u200E\u200D
          \u200E\u200D\u200D\u206C\u202B\u200F\u200E\u200B\u200F\u206C\u202D\u206D\u202E\u200C\u206B\u202C
          \u202A\u200D\u202B\u202B\u202A\u202D\u200C\u200F\u200D\u202E.\u202D\u202E\u206B\u206B\u200E\u200F
          \u206F\u200C\u202B\u200C\u206F\u202A\u206D\u200B\u206D\u206B\u200B\u202A\u206D\u206E\u200C\u202B
          \u202B\u206F\u200E\u200D\u202D\u202E\u206D\u206D\u200D\u200F\u202B\u200E\u206C\u200F\u200C\u206F
          \u206B\u206B\u202E();
        base.\u200D\u202E\u206D\u202A\u202A\u206E\u200B\u202A\u202D\u206B\u202A\u202C\u200D\u206F\u206A\u202D
          \u200D\u206F\u206F\u200C\u202A\u202B\u206F\u200B\u206E\u200E\u206F\u200F\u200F\u202C\u202C\u206E
          \u206A\u200D\u206C\u206D\u202C\u206F\u202E\u200F\u202E(<Module>.\u202A\u200D\u200C\u206A\u202E
          \u206A\u200F\u200B\u200E\u202C\u202D\u206A\u200C\u202E\u206C\u202E\u200E\u206A\u206F\u202C\u200D
          \u200D\u200E\u202A\u206F\u206C\u202C\u200B\u206A\u202E\u200E\u200D\u202B\u200B\u202E\u202C\u200C
          \u206A\u202A\u206C\u202E<string>(1155215460u));
        base.\u206B\u206E\u206D\u200D\u200B\u200F\u202D\u202D\u202A\u200F\u200D\u202E\u206B\u200D\u206A\u202B
          \u200F\u206A\u200B\u202B\u206B\u202E\u206E\u200D\u206D\u202E\u202C\u202D\u206F\u202C\u200C\u206D
          \u200D\u206D\u206A\u200F\u206A\u200B\u206D\u206B\u202E();
        if (\u202E\u200F\u200C\u206E\u200B\u206D\u206A\u200E\u206D\u206C\u206D\u202C\u200C\u200E\u200D\u200E
          \u200D\u200D\u206C\u202B\u200F\u200E\u200B\u200F\u206C\u202D\u206D\u202E\u200C\u206B\u202C\u202A
          \u200D\u202B\u202B\u202A\u202D\u200C\u200F\u200D\u202E.\u202D\u202E\u206B\u206B\u200E\u200F\u206F
          \u200C\u202B\u200C\u206F\u202A\u206D\u200B\u206D\u206B\u200B\u202A\u206D\u206E\u200C\u202B\u202B
          \u206F\u200E\u200D\u202D\u202E\u206D\u206D\u200D\u200F\u202B\u200E\u206C\u200F\u200C\u206F\u206B
          \u206B\u202E() - num > 500)
        {
            base.\u200D\u202E\u206D\u202A\u202A\u206E\u200B\u202A\u202D\u206B\u202A\u202C\u200D\u206F\u206A
              \u202D\u200D\u206F\u206F\u200C\u202A\u202B\u206F\u200B\u206E\u200E\u206F\u200F\u200F\u202C
              \u202C\u206E\u206A\u200D\u206C\u206D\u202C\u206F\u202E\u200F\u202E(<Module>.\u200C\u206C\u202C
```



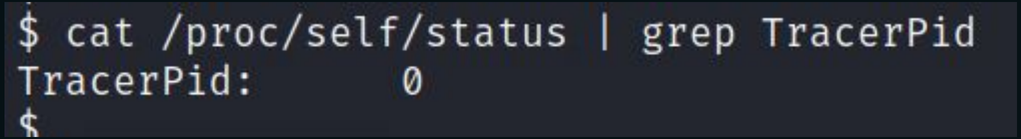
```
  local_c = 'W';
  local_b = 0x53;
  local_a = 0x41;
  local_9 = 0x52;
  local_8 = 0x65;
  local_7 = 99;
  local_6 = 0x76;
  local_5 = 0;
  if (DAT_1002955c == (FARPROC)0x0) {
    lpProcName = &local_c;
    hModule = FUN_10009790();
    DAT_1002955c = GetProcAddress(hModule,lpProcName);
  }
  (*DAT_1002955c)(param_1,param_2,param_3,param_4,param_5,param_6,param_7);
  return;
}
```

Example dynamic API lokup from PlugX malware

API Hashing

Instead of calling API functions directly, the sample uses a function that looks up API functions by hash

From static analysis, it’s hard to figure out which function is being called.



```
$ cat /proc/self/status | grep TracerPid
TracerPid:      0
$
```

TracerPid on Linux, if a debugger is attached, it will be non-zero

Debugger detection

- Checking the process-list
- Trying to attach as a debugger
- Trying to trigger and handle exceptions
- Calculating timing to detect breakpoints
- Checking for integrity to detect software breakpoints

VM detection

- Checking the process-list / services
- Checking the MAC address
- CPUID instruction
- Timing measurement (RDTSC instruction)

Conclusion

- Mitigations don’t “prevent” reverse engineering.
- They just make it require more time and effort.

Their goal: Make sure that:

The effort and resources required ⩾ Outcome from reverse engineering.

Practice time

Any questions?