بسم الله الرحمن الرحيم

Syieun university for technology

University since application

**IS** department level **4**

# The Title of Research

## (MALWARE STATIC ANALYSIS)

## GROUP (C)

Students' preparation

Nader Ameen AL Hwaimail

SUPERVISE By

A/Shadi Bajri

The Session 2023-2024

| The Files | Types |
|---|---|
| C1 | Not Malware |
| C2 | Malware (Trojan) |
| C3 | Not Malwar |
| C4 | Not Malwar |

*Info*

There is a similarity between files C3 and C4

```
~/.../C/C $ ssdeep * > fuzzy.txt
~/.../C/C $ ssdeep -m fuzzy.txt * -s
/storage/emulated/0/Project_1/C/C/C1 matches fuzzy.txt:/storage/emulated/0/Project_1/C/C/C1 (100)
/storage/emulated/0/Project_1/C/C/C2 matches fuzzy.txt:/storage/emulated/0/Project_1/C/C/C2 (100)
/storage/emulated/0/Project_1/C/C/C3 matches fuzzy.txt:/storage/emulated/0/Project_1/C/C/C3 (100)
/storage/emulated/0/Project_1/C/C/C3 matches fuzzy.txt:/storage/emulated/0/Project_1/C/C/C4 (90)
/storage/emulated/0/Project_1/C/C/C4 matches fuzzy.txt:/storage/emulated/0/Project_1/C/C/C3 (90)
/storage/emulated/0/Project_1/C/C/C4 matches fuzzy.txt:/storage/emulated/0/Project_1/C/C/C4 (100)
/storage/emulated/0/Project_1/C/C/-s: No such file or directory
```

*FuzzyHash*


*C4*


*C3*

## Trojan:

The Trojan virus first appeared in the early 1970s and was named after the famous Trojan Horse from Greek mythology. The target entity of a Trojan virus is typically a computer system or network, with the goal of gaining unauthorized access or causing damage to the system.

Trojan viruses can cause a variety of damages, including stealing sensitive information such as passwords and financial data, deleting files, disrupting system performance, and allowing remote access to the infected system for cybercriminals.

## The reference

https://www.fortinet.com/
https://www.webroot.com/

# Chapter I

# C1

## 1- Files' Fingerprints (MD5, SHA1):

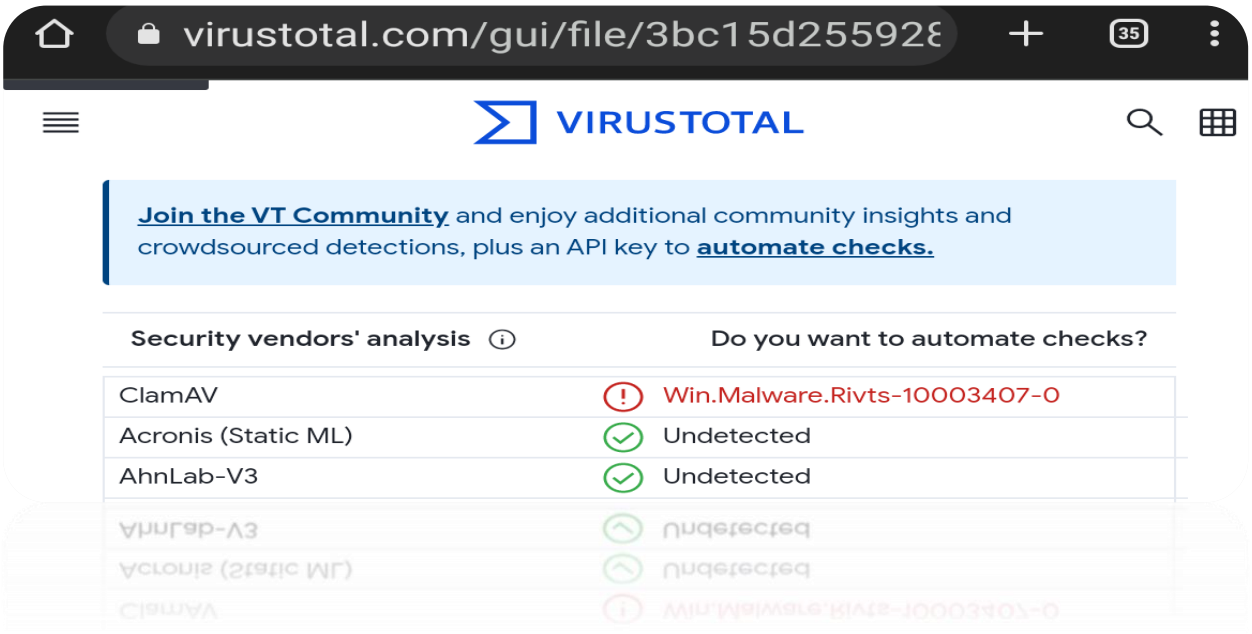
**md5** ---> fcc8437ee3696f3caf5ac9e59ed8772e

**sha1** ---> 8333718b08e91907204b962d0d2f3db7445e65ff

| | |
|---|---|
| MD5: | fcc8437ee3696f3caf5ac9e59ed8772e |
| SHA1: | 8333718b08e91907204b962d0d2f3db7445e65ff |

*Fingerprint*

## 2- The first 3 result in virustotal:

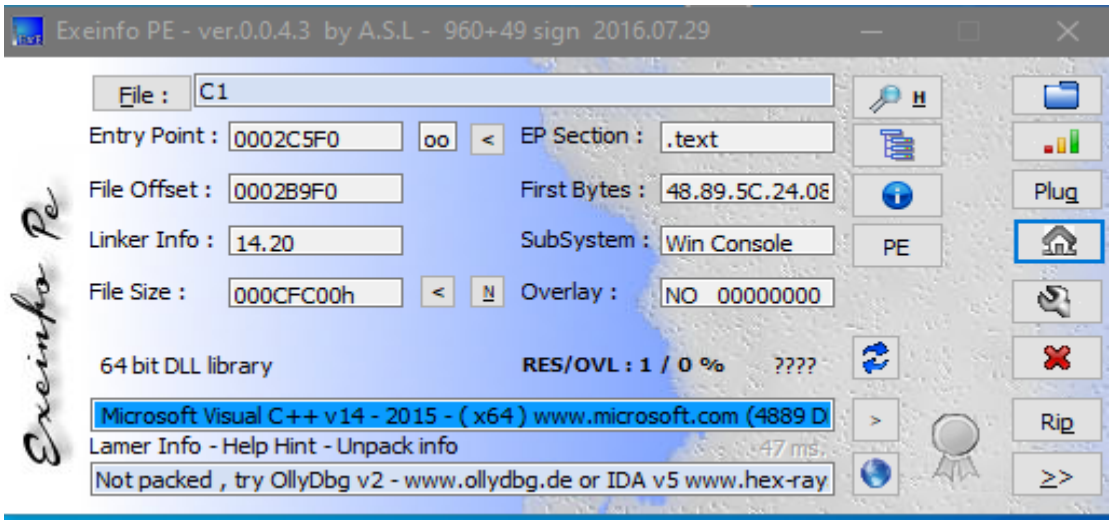


## 3- Text strings containing flags to mark the file as malicious or harmless:

## 4- was file has been packed or compressed (Packed)?

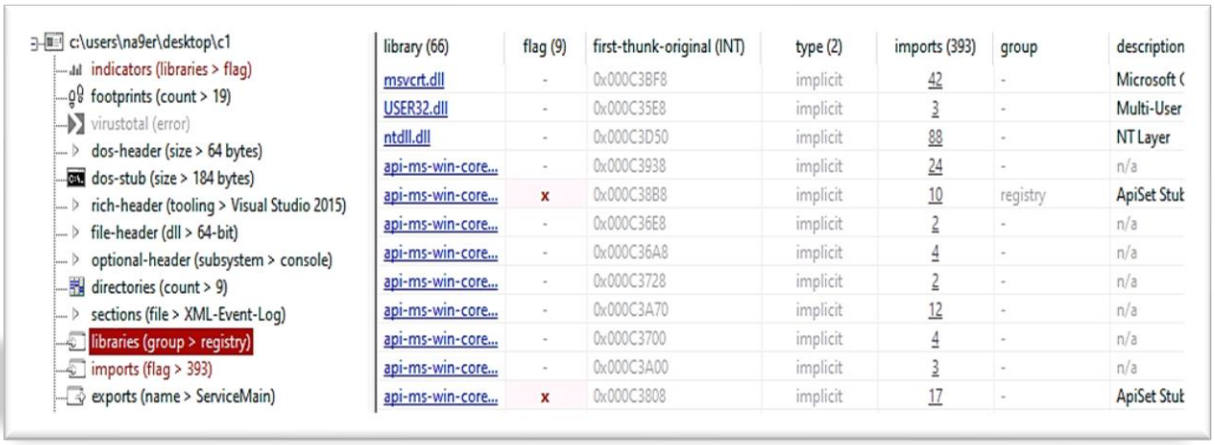
- BY Exeinfo PE the file C1 not packed

Not packed , try OllyDbg v2 - www.ollydbg.de or IDA v5 www.hex-rays.com or x64 debug v0024 www.x64dbg.com




## 5- Used libraries:

- There are 66 libraries in File C1



| libraries | Short Description |
|---|---|
| msvcrt.dll | Microsoft C++ Runtime Library |
| USER32.dll | Multi-User Windows USER API Client Library |
| dwmapi.dll | Microsoft Desktop Window Manager API |

**Additional details:**
**msvcrt.dll**

The msvcrt.dll file is a part of the "Microsoft Visual Studio 6.0" and is crucial for most applications to work properly. It also contains program code that enables applications written in "Microsoft Visual C++" to run properly. The program code is basically string comparison tools and mathematic operations such as trigonometric operations. This is a very valuable file to programmers.

Some games or applications may need the file in the game/application installation folder. Copying it from Windows systemfolder to the install-folder of the game/application should fix that problem. https://www.dll-files.com/

## USER32.dll

The "USER32.dll" library is a dynamic link library (DLL) file that is a part of the Windows operating system. It contains a set of functions and components that provide user interface services for Windows applications.

The "USER32.dll" library is responsible for managing and controlling windows, menus, dialog boxes, buttons, scrollbars, and other graphical elements of the user interface. It provides functions for creating and manipulating windows, handling user input, managing keyboard and mouse events, and performing various UI-related tasks.As for the source of the "USER32.dll" library, it is developed and provided by Microsoft as an integral component of the Windows operating system. It is not available for separate download or distribution. The library is typically installed along with the Windows operating system and is located in the "System32" folder on a Windows system. **More detail** https://strontic.github.io/xcyclopedia/library/user32.dll

**dwmapi.dll**

The dwmapi.dll file is a crucial component of the Windows operating system, responsible for managing visual effects and GUI elements. However, it can encounter errors that may disrupt the functioning of applications and processes. By following the methods mentioned above, such as restarting the computer, running virus scans, updating Windows and drivers, performing SFC scans, and reinstalling or repairing applications, users can effectively resolve dwmapi.dll errors and restore the smooth operation of their Windows system.

https://malwaretips.com/blogs/dwmapi-dll

## 6- Import Function:

- The are 393 imports functions in file C1

| Import Fun | Short Description |
|---|---|
| NtOpenEvent | Opens an existing event object |
| DeleteTimerQueueTimer | Deletes a timer from queue |
| RegisterServiceCtrlHandlerExW | Registers handler for controlling Windows services. |

**Additional details From: → https://learn.microsoft.com/**

**NtOpenEvent**

ZwOpenEvent can open either notification or synchronization events. Events are used to coordinate execution. File system drivers can use events to enable a caller to wait for completion of the requested operation until the given event is set to the Signaled state.

```
C++                                                    Copy

NTSYSCALLAPI NTSTATUS ZwOpenEvent(
  [out] PHANDLE            EventHandle,
  [in]  ACCESS_MASK        DesiredAccess,
  [in]  POBJECT_ATTRIBUTES ObjectAttributes
);
```

*Syntax*

**DeleteTimerQueueTimer**

function is a Windows API function that is used to delete a timer from a timer queue.

Timer queues are used to schedule timer callbacks, allowing applications to execute certain code at specific intervals or after a certain amount of time has elapsed. The **DeleteTimerQueueTimer** function allows you to cancel or remove a previously created timer from the timer queue, preventing the associated callback function from being executed.

By calling **DeleteTimerQueueTimer**, you provide the handle to the timer queue and the timer to be deleted. Upon successful deletion, the resources associated with the timer are released, and the timer is removed from the queue.

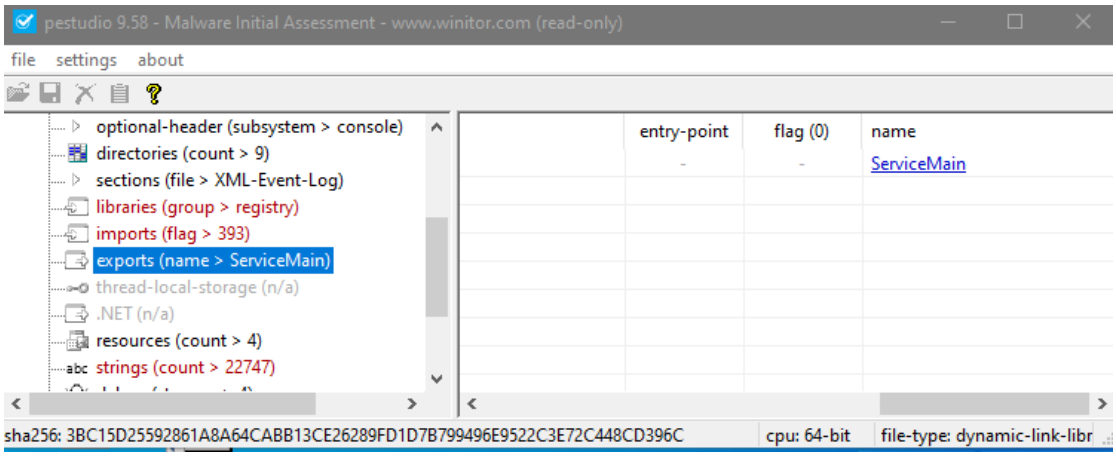
C++ Copy

```
BOOL DeleteTimerQueueTimer(
  [in, optional] HANDLE TimerQueue,
  [in]           HANDLE Timer,
  [in, optional] HANDLE CompletionEvent
);
```

*Syntax*

## 7- Export Function:

There is one only export function in file C1




| Export Fun | Short Description |
|---|---|
| ServiceMain | Windows service entry point |

A **ServiceMain** function in a service DLL can be coded as for one in a service program. Refer to Microsoft's documentation. However, some work that might be done in a service program can be avoided in a service DLL because it is already done by SVCHOST. First, there is per-process initialization which is not only convenient to be done by SVCHOST but is perhaps better attempted only once per process (as with initializing COM security). Second, though only since version 5.1, SVCHOST provides service DLLs with access to shared code and data. To learn the addresses, a service DLL should export a function named SvchostPushServiceGlobals, which SVCHOST calls before each call to any ServiceMain function in the DLL.

https://www.geoffchappell.com/ .

**Example**

This example uses the following global definitions.

C++ Copy

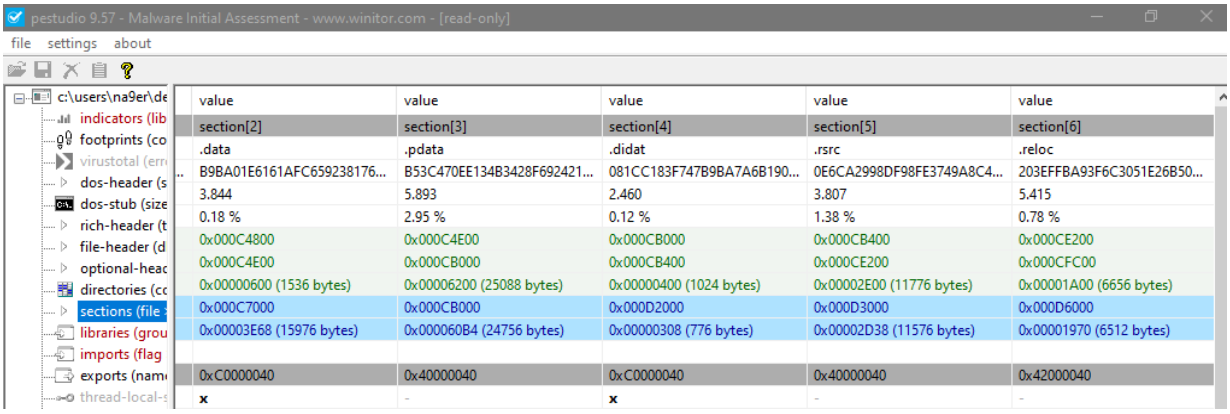
```cpp
#define SVCNAME TEXT("SvcName")

SERVICE_STATUS          gSvcStatus;
SERVICE_STATUS_HANDLE   gSvcStatusHandle;
HANDLE                  ghSvcStopEvent = NULL;
```

## 8- Section:

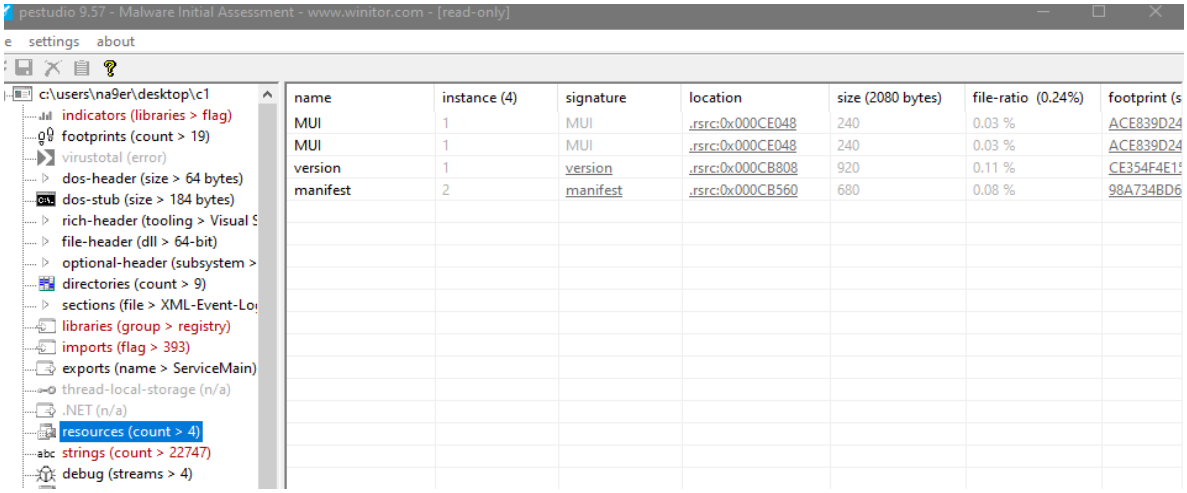
There are 7 section in file C1

| **sec** | Section [0] | Section [1] | Section [2] | Section [3] | Section [4] | Section [4] | Section [6] |
|---|---|---|---|---|---|---|---|
| **Name** | **.text** | **.rdata** | **.data** | **.pdata** | **.didat** | **.rsrc** | **.reloc** |
| **Raw-size in bytes** | 537088 bytes | 266752 bytes | 1536 bytes | 25088 bytes | 1024 bytes | 11776 bytes | 6656 bytes |
| **Virtual size in bytes** | 536610 bytes | 266342 bytes | 15976 bytes | 24756 bytes | 776 bytes | 11576 bytes | 6512 bytes |
| **Entry point .txt** | 0x0002 C5F0 | --- | ---- | --- | ---- | --- | ---- |
| **Raw add begin** | 0x00000 400 | 0x00083 600 | 0x000C 4800 | 0x000C 4E00 | 0x000C B000 | 0x000CB 400 | 0x000C E200 |
| **Raw add end** | 0x00083 600 | 0x000C 4800 | 0x000C 4E00 | 0x000C B000 | 0x000C B400 | 0x000CE 200 | 0x000CF C00 |

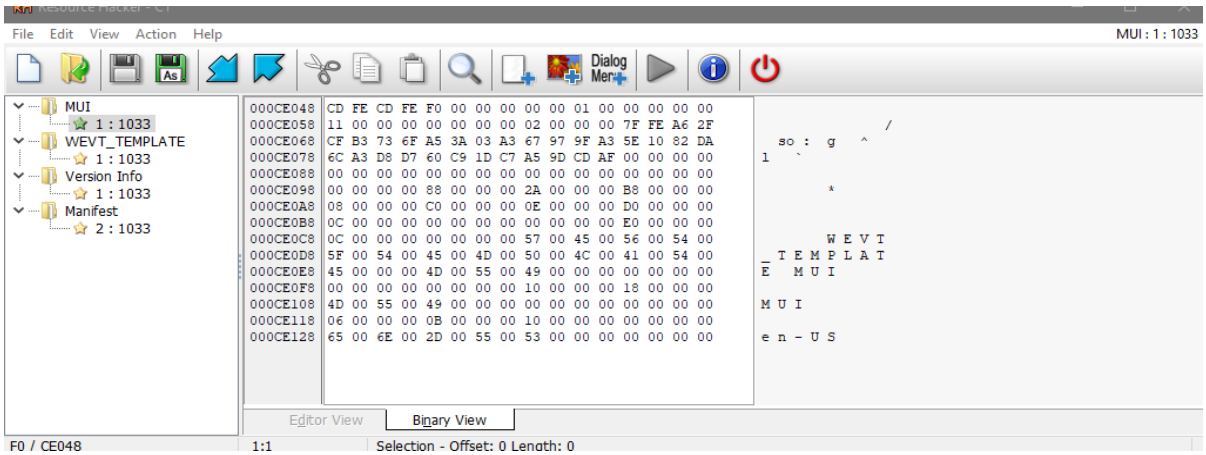


## 9- Count of resource:

There are 4 resources in file C1




*BY pestudio tool*



# Chapter II

# C2

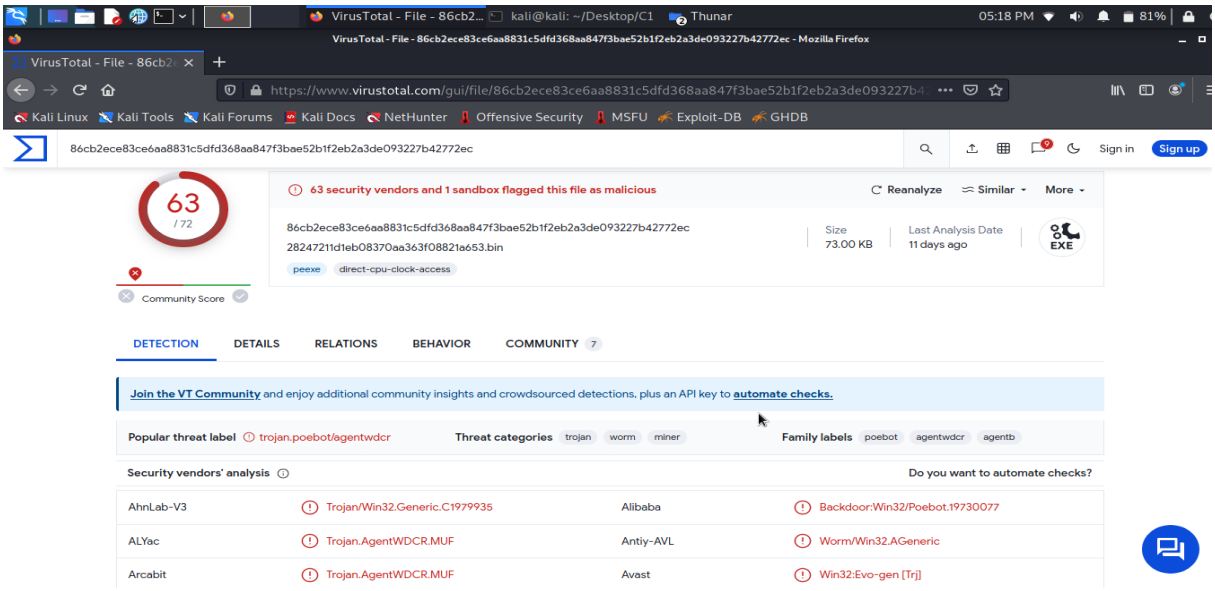
## 1- Files' Fingerprints (MD5, SHA1):

**md5** ---> 28247211d1eb08370aa363f08821a653

**sha1** ---> 9d16705ff7bd06d238b389f9320e1c646639c2f7

| | |
|---|---|
| MD5: | 28247211d1eb08370aa363f08821a653 |
| SHA1: | 9d16705ff7bd06d238b389f9320e1c646639c2f7 |

## 2- The first 3 result in virustotal:

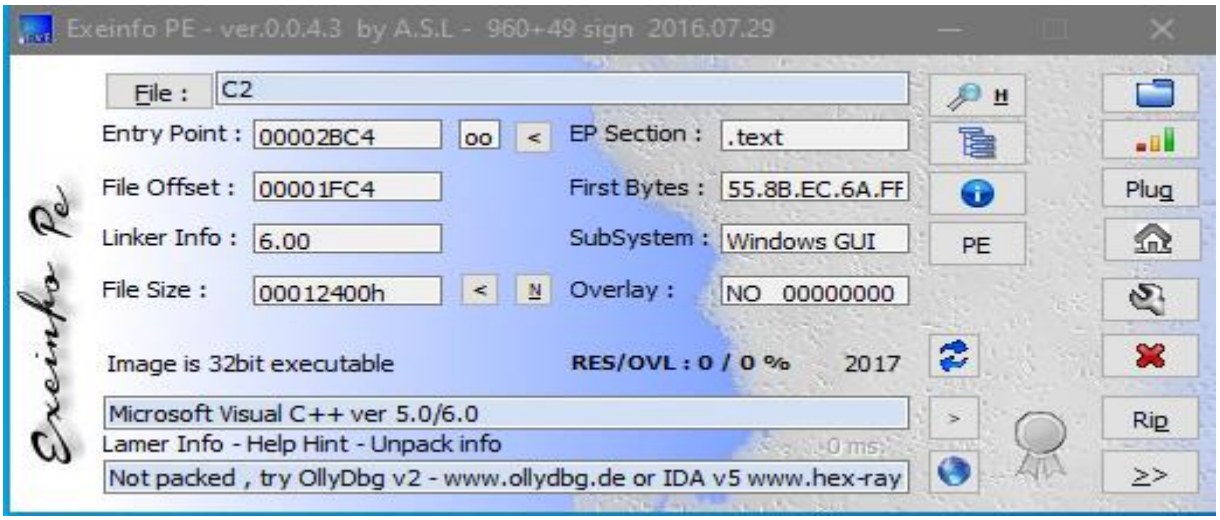


## 3- Text strings containing flags to mark the file as malicious or harmless:

The presence of error messages related to the Microsoft Visual C++ Runtime Library, such as "Runtime Error!" and various error codes (e.g., R6002, R6008, R6009, etc.), suggests that the

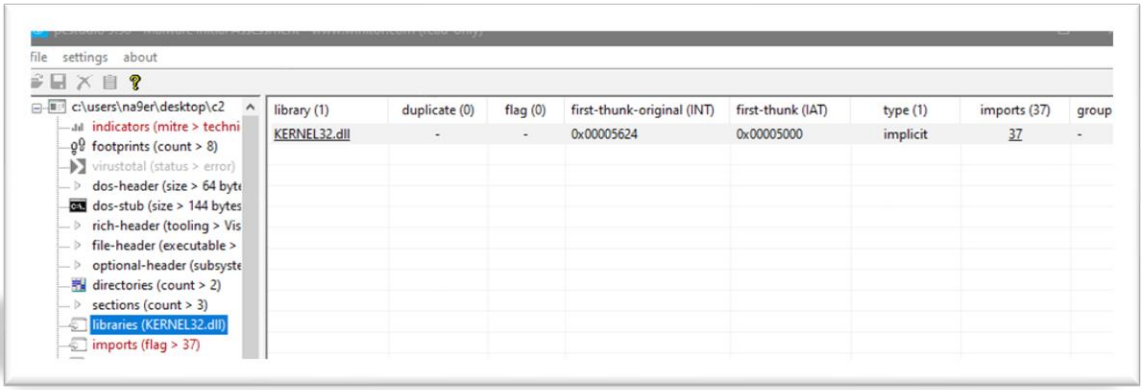
## 4- was file has been packed or compressed (Packed)?

- BY Exeinfo PE the file C2 not packed




## 5- Used libraries:

- There is ONE only libraries in File C2




| libraries | Short Description |
|---|---|
| KERNEL32.dll | Windows NT BASE API Client |

## KERNEL32.dll:

Kernel32.dll is the 32-bit dynamic link library found in the Windows operating system kernel. It handles memory management, input/output operations, and interrupts. When Windows boots up, kernel32.dll is loaded into a protected memory space so other applications do not take that space over.

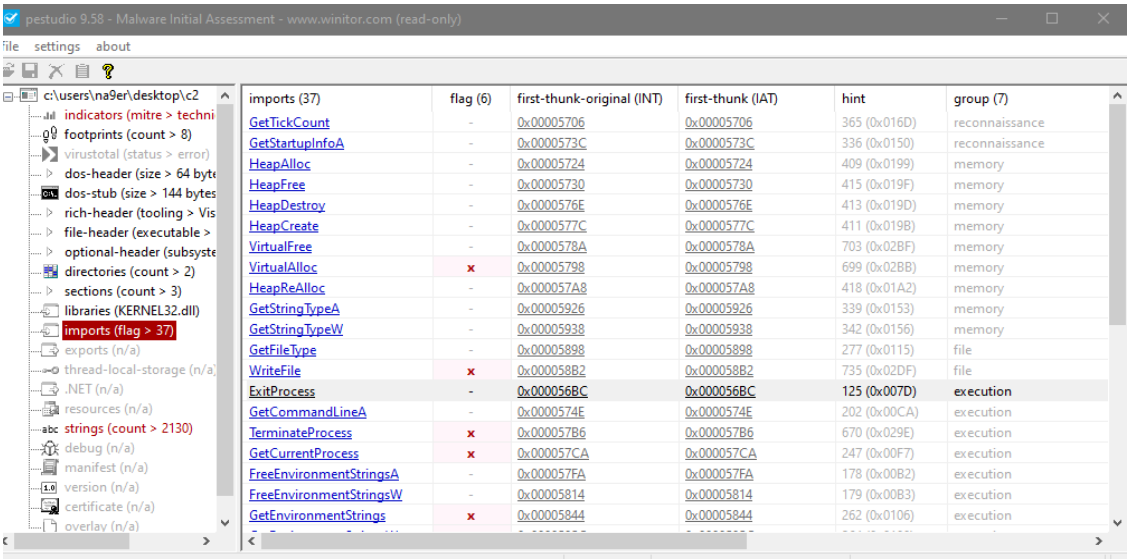
On occasion, though, users may encounter the “invalid page fault” error.

This error occurs when a program or application tries to access kernel32.dll’s protected memory space. Sometimes the error is caused by one particular program or application, and other times it is provoked by multiple files and applications.

**The reference** https://www.webopedia.com/

## 6- Import Function:

- There are 37 imports functions in file C2




| Import Fun | Short Description |
|---|---|
| VirtualAlloc | Allocate virtual memory |
| WriteFile | Write to file |
| ExitProcess | Terminate current process |

**Additional details from: → https://learn.microsoft.com/**

**VirtualAlloc**

The **VirtualAlloc** function is used to allocate memory in the virtual address space of a process. It reserves or commits a region of pages, allowing the process to use that memory for various purposes, such as storing data or executing code.

```cpp
LPVOID VirtualAlloc(
  [in, optional] LPVOID lpAddress,
  [in]           SIZE_T dwSize,
  [in]           DWORD  flAllocationType,
  [in]           DWORD  flProtect
);
```

*Syntax*

## WriteFile

The **WriteFile** function is used to write data to a file or input/output (I/O) device. It is commonly used to write bytes, characters, or other data to a file, pipe, console, or serial communication resource.

```cpp
BOOL WriteFile(
  [in]                HANDLE       hFile,
  [in]                LPCVOID      lpBuffer,
  [in]                DWORD        nNumberOfBytesToWrite,
  [out, optional]     LPDWORD      lpNumberOfBytesWritten,
  [in, out, optional] LPOVERLAPPED lpOverlapped
);
```

*Syntax*

## ExitProcess

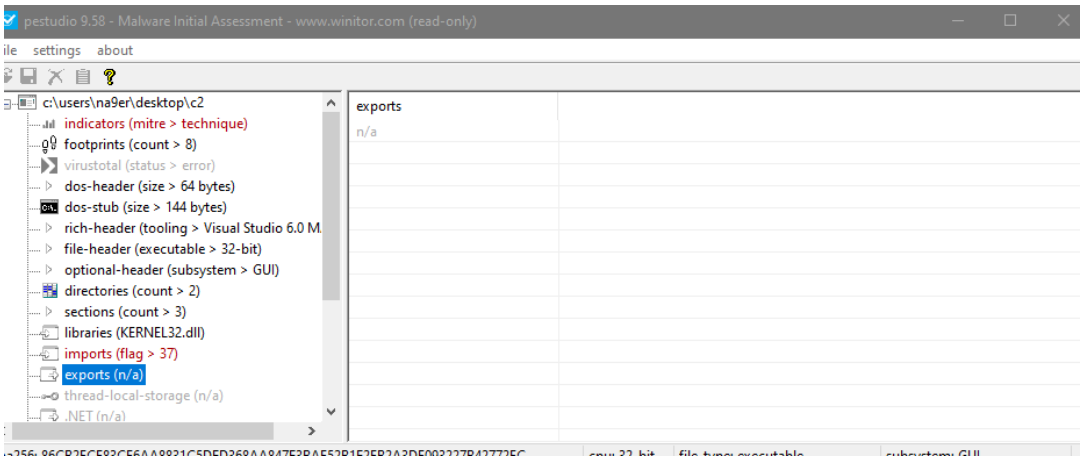
The **ExitProcess** function is used to terminate the calling process and exit it gracefully. It performs the necessary cleanup and terminates the process, including closing open handles and freeing resources.

```cpp
void ExitProcess(
  [in] UINT uExitCode
);
```

*Syntax*

## 7- Export Function:

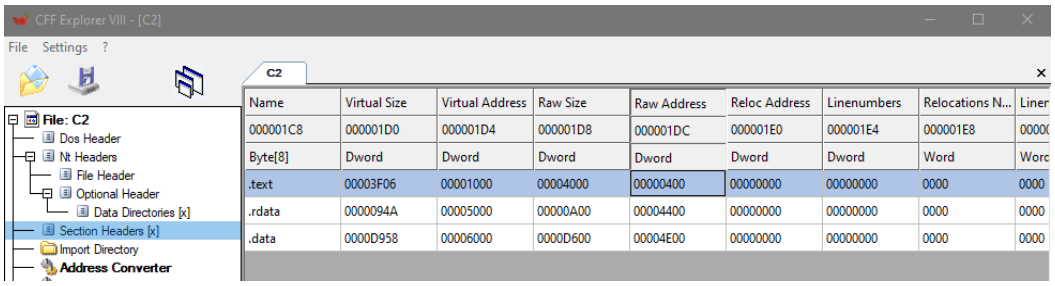
- There is NO exports function in file C2

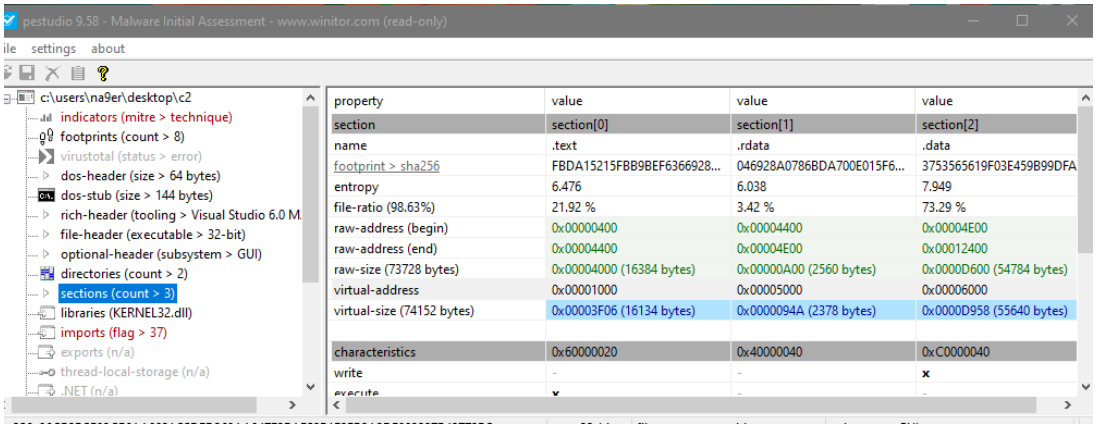


## 8- Section: → entropy 7,747

- There are Three section in file C2

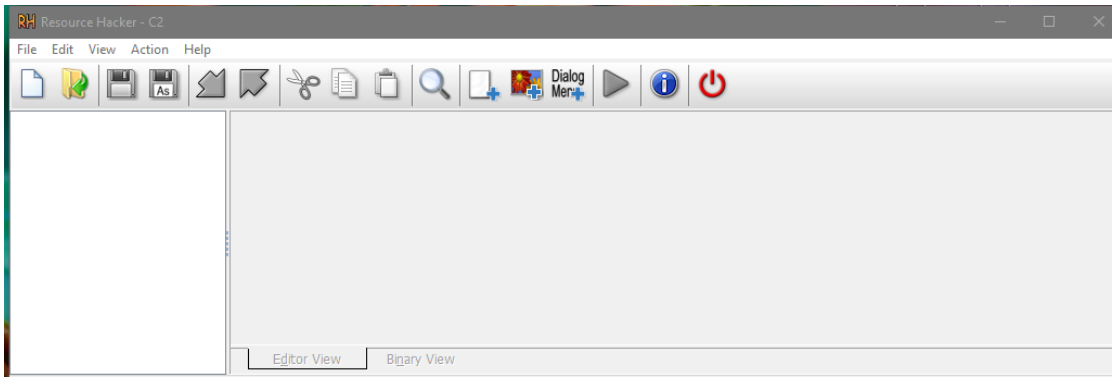
| sec | Section [0] | Section [1] | Section [2] |
|---|---|---|---|
| **Name** | **.text** | **.rdata** | **.data** |
| **Raw-size in bytes** | 16384 bytes | 2560 bytes | 54784 bytes |
| **Virtual size in bytes** | 16134 bytes | 2378 bytes | 55640 bytes |
| **Entry point .txt** | 0x00002BC4 | --- | ---- |
| **Raw add begin** | 0x00000400 | 0x00004400 | 0x00004E00 |
| **Raw add end** | 0x00004400 | 0x00004E00 | 0x00012400 |



## 9- Count of resource:

- There are zero resources in file C2



# Chapter III

# C3

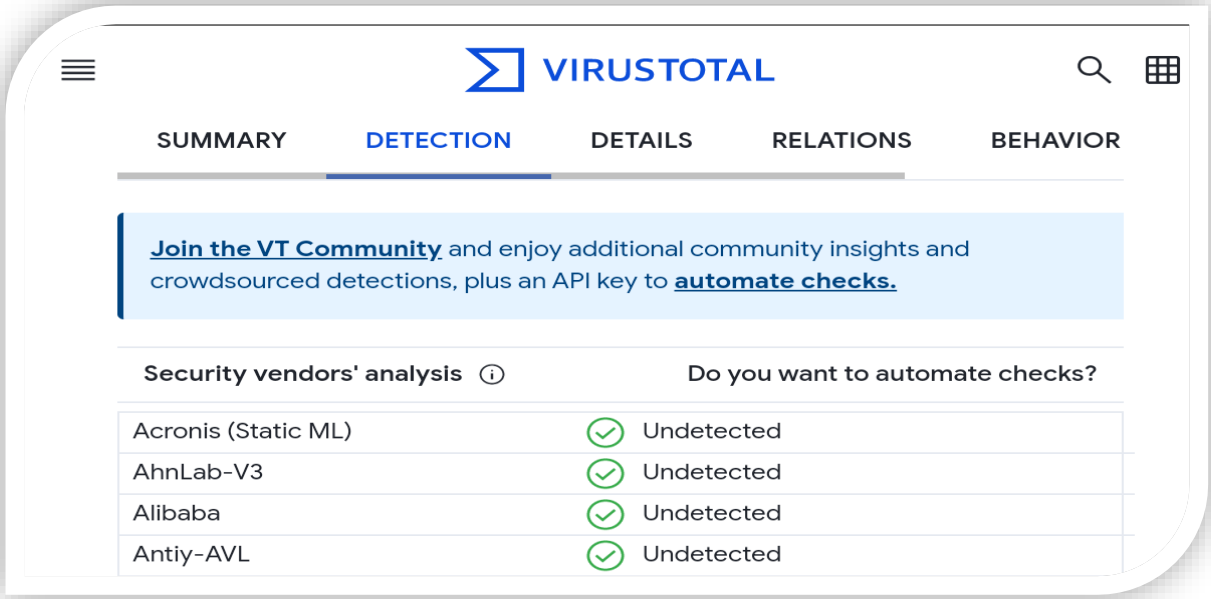
## 1- Files' Fingerprints (MD5, SHA1):

**md5** ---> 4d05488fec7d3fb63248ca9652b48cbb

**sha1** ---> 5877f71c48ae3f5f36f77e78ec86e4c7a7f1fba7

| | |
|---|---|
| MD5: | 4d05488fec7d3fb63248ca9652b48cbb |
| SHA1: | 5877f71c48ae3f5f36f77e78ec86e4c7a7f1fba7 |

## 2- The first 3 result in virustotal:



## 3- Text strings containing flags to mark the file as malicious or harmless:

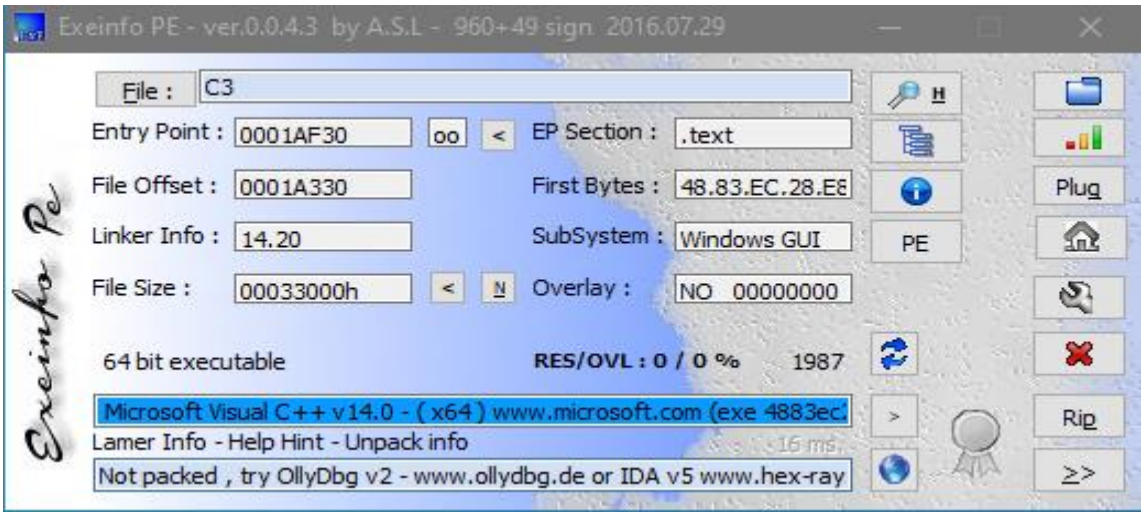
| هل هو ضار | عمله | String |
|---|---|---|
| غير ضار | EnterCriticalSection تستخدم لدخول قسم الحماية المتزامنة (critical section) في برنامج يعمل على نظام التشغيل Windows. | EnterCriticalSection |
| غير ضار | LeaveCriticalSection تستخدم للخروج من قسم الحماية المتزامنة (critical section) في برنامج يعمل على نظام التشغيل Windows. | LeaveCriticalSection |
| غير ضار | CreateSemaphoreEx تستخدم لإنشاء كائن سمافور (semaphore object) في برنامج يعمل على نظام التشغيل Windows. | CreateSemaphoreEx |
| غير ضار | InitializeCriticalSection تستخدم لتهيئة قسم الحماية المتزامنة (critical section) في برنامج يعمل على نظام التشغيل Windows. | InitializeCriticalSection |
| غير ضار | SetEvent تستخدم لتعيين حالة الحدث إلى "مفتوح" وإشعار العمليات المنتظرة على حدوث الحدث وهي غير ضارة. | SetEvent |
| غير ضار | DeleteCriticalSection تستخدم لحذف قسم الحماية المتزامنة (critical section) بعد الانتهاء من استخدامه في برنامج يعمل على نظام التشغيل Windows. | DeleteCriticalSection |

- c:\users\sa\desktop\c3
  - indicators (libraries > flag)
  - footprints (count > 17)
  - virustotal (error)
  - dos-header (size > 64 bytes)
  - dos-stub (size > 192 bytes)
  - rich-header (tooling > Visual Studio 2015)
  - file-header (executable > 64-bit)
  - optional-header (subsystem > GUI)
  - directories (count > 10)
  - sections (count > 7)
  - libraries (group > registry)
  - imports (flag > 180)
  - exports (duplicate > 33)
  - thread-local-storage (n/a)
  - .NET (n/a)
  - resources (count > 3)
  - strings (count > 4874)
  - debug (streams > 3)
  - manifest (level > asInvoker)
  - version (FileDescription > WMI Performance I
  - certificate (n/a)
  - overlay (n/a)

| label (472) | group (13) | technique (16) | value |
|---|---|---|---|
| import | synchronization | - | CreateSemaphoreEx |
| import | synchronization | - | TryEnterCriticalSection |
| import | synchronization | - | LeaveCriticalSection |
| import | synchronization | - | WaitForSingleObject |
| import | synchronization | - | ReleaseSemaphore |
| import | synchronization | - | InitializeCriticalSection |
| import | synchronization | - | ResetEvent |
| import | synchronization | - | SetEvent |
| import | synchronization | - | EnterCriticalSection |
| import | synchronization | - | ReleaseMutex |
| import | synchronization | - | DeleteCriticalSection |
| import | synchronization | - | CreateMutex |
| import | synchronization | - | CreateEvent |
| import | synchronization | - | WaitForMultipleObjectsEx |
| import | synchronization | - | ReleaseSRWLockExclusive |
| import | synchronization | - | AcquireSRWLockExclusive |
| import | synchronization | - | WakeAllConditionVariable |
| import | synchronization | - | SleepConditionVariableSR |
| import | synchronization | - | OpenEvent |
| import | services | T1569 \| System Services | CloseServiceHandle |
| import | services | T1569 \| System Services | OpenSCManager |
| import | services | T1543 \| Create or Modify System Proc... | OpenService |
| import | services | T1569 \| System Services | QueryServiceConfig |
| import | services | T1569 \| System Services | QueryServiceStatus |

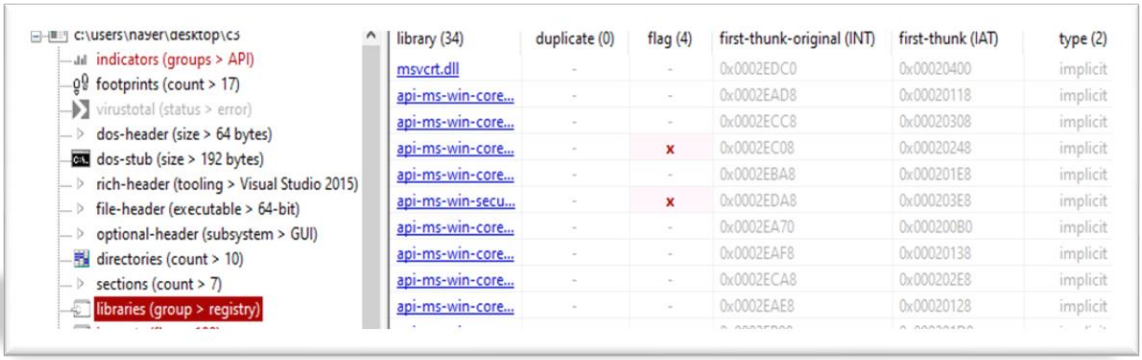
## 4- was file has been packed or compressed (Packed)?

- BY Exeinfo PE the file C3 not packed




## 5- Used libraries:

- There is 34 libraries in File C3



| libraries | Short Description |
|---|---|
| msvcrt.dll | Microsoft C++ Runtime Library |
| ntdll.dll | NT Layer |
| **api-ms-win-security-base-l1-1-0.dll** | ApiSet Stub Library |

## Additional details:

## ntdll.dll

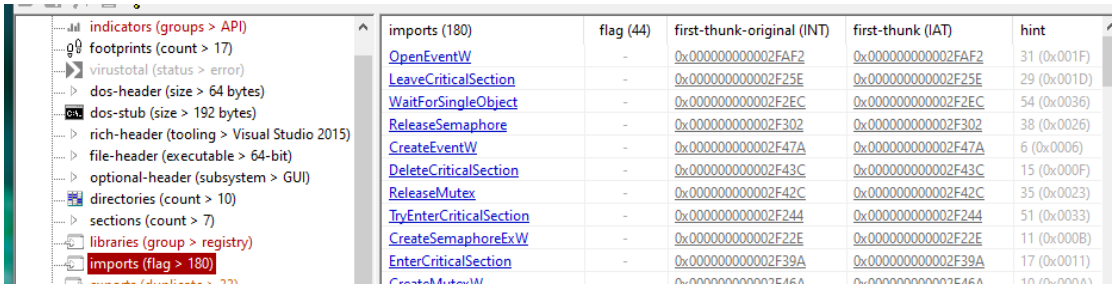
The **ntdll.dll** is a crucial component of the Windows operating system. It plays a vital role in the execution of various system processes and provides essential functions for the smooth operation of your computer. However, encountering errors related to ntdll.dll can be frustrating and disruptive to your workflow. In this article, we will explore what ntdll.dll is, common error messages associated with it, and effective methods to fix these errors. https://malwaretips.com/

## api-ms-win-security-base-l1-1-0.dll

api-ms-win-security-base-l1-1-0.dll is a dynamic link library (DLL) file that is part of the Microsoft Windows operating system. It contains essential functions related to security and base-level operations within the Windows security subsystem. The "api-ms-win-security-base-l1-1-0.dll" file provides a set of application programming interfaces (APIs) that allow software developers to access and utilize various security features and functionalities provided by the Windows operating system. These functions may include authentication, authorization, access control, and other security-related tasks. The DLL file is crucial for the proper functioning of security-related applications and services on a Windows system.

## 6- Import Function:

- There are 180 imports functions in file C3




| Import Fun | Short Description |
|---|---|
| WaitForSingleObject | Wait for object completion |
| ReleaseSemaphore | Release semaphore resource. |
| SetEvent | Set event signal |

### Additional details:

### WaitForSingleObject

The WaitForSingleObject function is used to wait until the specified object is in the signaled state or until the specified timeout interval elapses. It is commonly used to synchronize threads or to wait for a specific event or resource to become available.

```cpp
DWORD WaitForSingleObject(
  [in] HANDLE hHandle,
  [in] DWORD  dwMilliseconds
);
```

## ReleaseSemaphore

The **ReleaseSemaphore** function is used to release a semaphore object, allowing other threads or processes waiting on the semaphore to proceed. It increments the semaphore's count by a specified amount.

```cpp
BOOL ReleaseSemaphore(
  [in]            HANDLE hSemaphore,
  [in]            LONG   lReleaseCount,
  [out, optional] LPLONG lpPreviousCount
);
```

## SetEvent

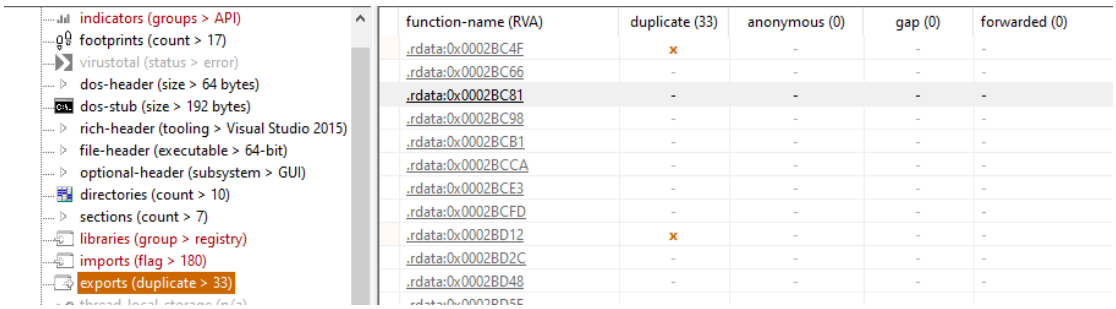
The **SetEvent** function is used to set the state of the specified event object to signaled, allowing threads or processes waiting on the event to proceed.

```cpp
BOOL SetEvent(
  [in] HANDLE hEvent
);
```

## 7- Export Function:

- There are 33 export function in file C3

| Export Fun | Short Description |
|---|---|
| CHPtrArray | Dynamic pointer array |
| **CHString** | String object manipulation |
| **CHPtrArray** | Dynamic pointer array |



## Additional details:

**CHPtrArray** is a class that represents a dynamic array of pointers. It provides functionality for managing and manipulating an array of pointers to objects or data.

```cpp
void * GetAt(
  int nIndex
);
```

**CHString** is a class that represents a string object. It provides methods for working with strings, including operations such as concatenation, comparison, and manipulation.

## 8- Section:

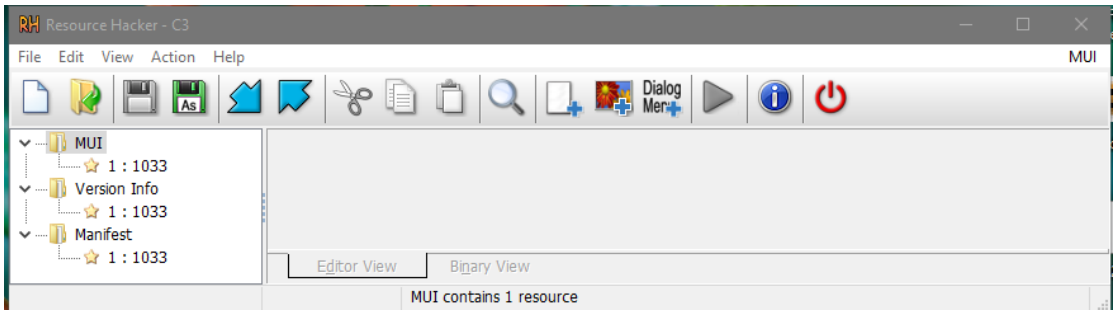
- There are 7 section in file C3

| **sec** | Section [0] | Section [1] | Section [2] | Section [3] | Section [4] | Section [4] | Section [6] |
|---|---|---|---|---|---|---|---|
| **Name** | **.text** | **.rdata** | **.data** | **.pdata** | **.didat** | **.rsrc** | **.reloc** |
| **Raw-size in bytes** | 122368 bytes | 69120 bytes | 4096 bytes | 8192 bytes | 512 bytes | 2560 bytes | 1024 bytes |
| **Virtual size in bytes** | 121919 bytes | 69090 bytes | 6112 bytes | 7836 bytes | **296** bytes | 2080 bytes | **1004** bytes |
| **Entry point .txt** | 0x0001 AF30 | --- | ---- | --- | ---- | --- | ---- |
| **Raw add begin** | 0x00000 400 | 0x0001 E200 | 0x0002 F000 | 0x00030 000 | 0x00032 000 | 0x00032 200 | 0x00032 C00 |
| **Raw add end** | 0x0001 E200 | 0x0002 F000 | 0x00030 000 | 0x00032 000 | 0x00032 200 | 0x00032 C00 | 0x00033 000 |

| property | value | value | value |
|---|---|---|---|
| section | section[0] | section[1] | section[2] |
| name | .text | .rdata | .data |
| footprint > sha256 | 0B32D20B17EFFAE389479374... | 9528E6A31FB8C9A38AB04D... | 397DF079F0B7C22246F86B2.. |
| entropy | 6.085 | 4.539 | 4.168 |
| file-ratio (99.51%) | 58.58 % | 33.09 % | 1.96 % |
| raw-address (begin) | 0x00000400 | 0x0001E200 | 0x0002F000 |
| raw-address (end) | 0x0001E200 | 0x0002F000 | 0x00030000 |
| raw-size (207872 bytes) | 0x0001DE00 (122368 bytes) | 0x00010E00 (69120 bytes) | 0x00001000 (4096 bytes) |
| virtual-address | 0x00001000 | 0x0001F000 | 0x00030000 |
| virtual-size (208337 bytes) | 0x0001DC3F (121919 bytes) | 0x00010DE2 (69090 bytes) | 0x000017E0 (6112 bytes) |

## 9- Count of resource:

- There are three resources in file C3




| name | instance (3) | signature | location | size (1830 bytes) | file-ratio (0. |
|---|---|---|---|---|---|
| MUI | 1 | MUI | .rsrc:0x00032958 | 200 | 0.10 % |
| version | 1 | version | .rsrc:0x000325A0 | 948 | 0.45 % |
| manifest | 1 | manifest | .rsrc:0x000322F0 | 682 | 0.33 % |

# Chapter IV

# C4

## 1- Files' Fingerprints (MD5, SHA1):

**md5** ---> 047c5dba99838a5d056ed4ade6fa28dd

**sha1** ---> 54daa3d6701db2239965d743afffa4e29ec1dfde

| | |
|---|---|
| MD5: | 047c5dba99838a5d056ed4ade6fa28dd |
| SHA1: | 54daa3d6701db2239965d743afffa4e29ec1dfde |

## 2- The first 3 result in virustotal:

VIRUSTOTAL

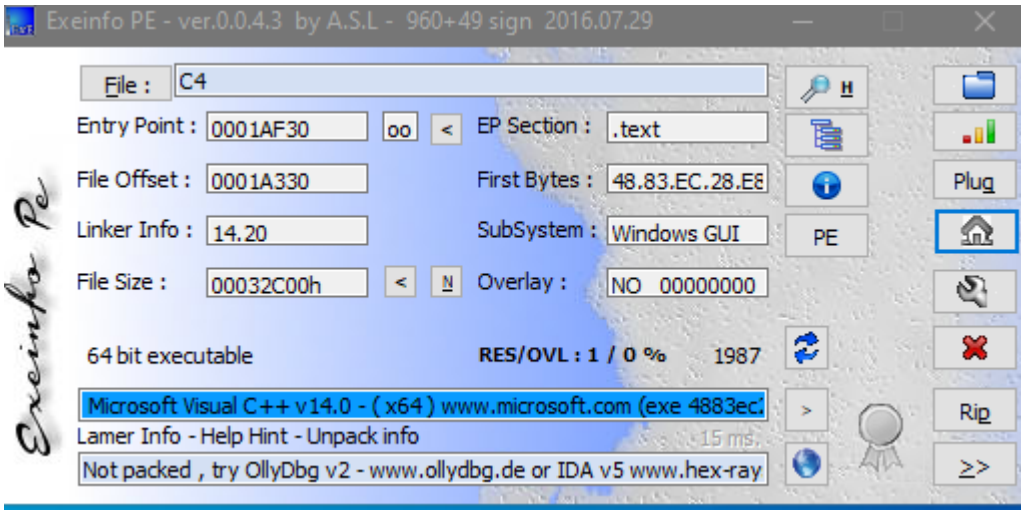
Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

| Security vendors' analysis | Do you want to automate checks? |
|---|---|
| SecureAge | Malicious |
| Acronis (Static ML) | Undetected |
| AhnLab-V3 | Undetected |
| Alibaba | Undetected |
| ALYac | Undetected |
| Antiy-AVL | Undetected |

## 3- Text strings containing flags to mark the file as malicious or harmless:

## 4- was file has been packed or compressed (Packed)?

- BY Exeinfo PE the file C4 not packed




## 5- Used libraries:

- There is 34 libraries in File C4

| libraries | Short Description |
|---|---|
| oleaut32.dll | oleaut32 library |
| api-ms-win-core-com-l1-1-0.dll | ApiSet Stub Library |
| **api-ms-win-core-memory-l1-1-0.dll** | ApiSet Stub Library |

## Additional details:

### oleaut32.dll

oleaut32.dll is a dynamic link library (DLL) file that is an integral part of the Microsoft Windows operating system. It stands for "Object Linking and Embedding Automation Extensions 32-bit." This DLL file contains functions and interfaces that enable software developers to create and manipulate objects, automate tasks, and facilitate inter-process communication within the Windows environment.

oleaut32.dll provides a set of Application Programming Interfaces (APIs) known as the Automation interfaces. These interfaces allow applications to interact with other applications and components, enabling features like scripting, data exchange, and automation.

### api-ms-win-core-com-l1-1-0.dll

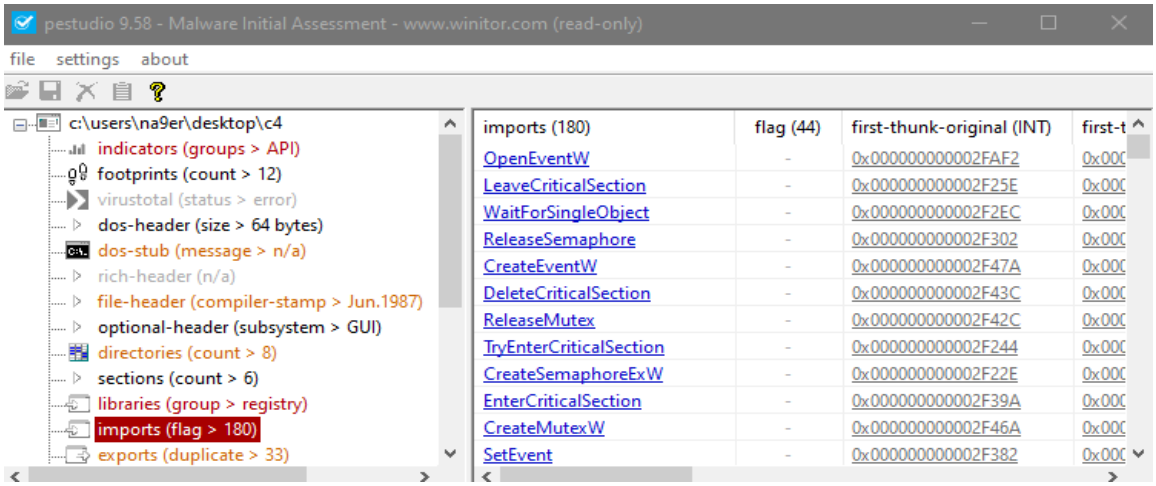
is a dynamic link library (DLL) file that is part of the Microsoft Windows operating system. It contains essential functions

related to Component Object Model (COM) and COM-based operations within the Windows core subsystem.COM is a binary-interface standard used in Windows for software components to communicate and interact with each other. It enables interprocess communication, object creation, and method invocation between software components, regardless of the programming language they are written in.

## 6- Import Function:

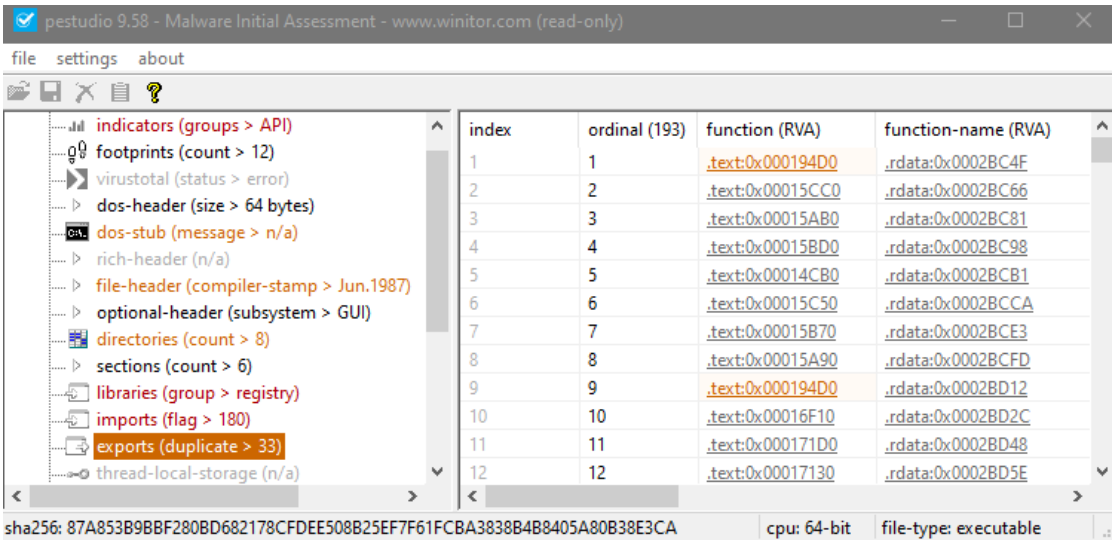
- There are 180 imports functions in file C4
  WaitForSingleObject
  ReleaseSemaphore
  SetEvent

  The file C4 the Same as file C3 in import functions




## 7- Export Function:

- There are 33 export function in file C4




| Export Fun | Short Description |
|---|---|
| **CHPtrArray::CHPtrArray(void)** | Initialize CHPtrArray instance. |
| **CRegistrySearch::CRegistrySearch** | Copy CRegistrySearch instance. |
| **CHString::~CHString(void)** | Delete CHString instance |

## Additional details:

## CHPtrArray

This function is a constructor for the **CHPtrArray** class. It creates a new instance of the class and initializes it.

### CRegistrySearch

This function is a copy constructor for the **CRegistrySearch** class. It creates a new instance of the class by making a copy of an existing instance.

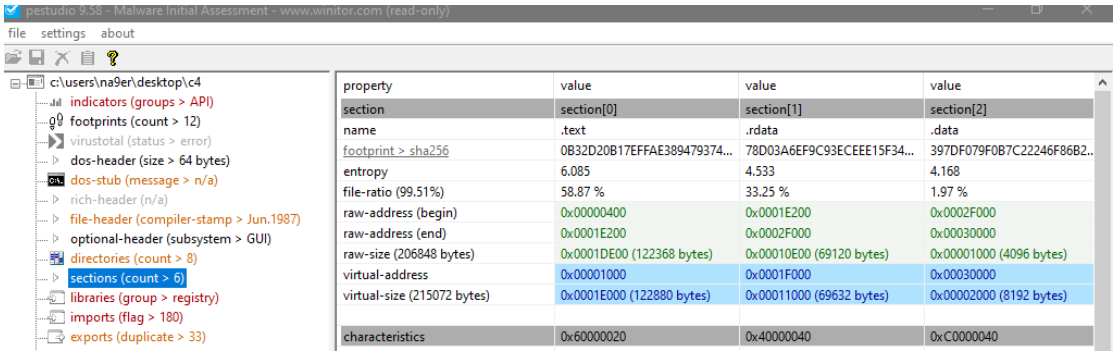
### CHString

This function is a destructor for the **CHString** class. It is responsible for cleaning up and releasing any resources held by an instance of the class when it is destroyed.

## 8- Section:

- There are 6 section in file C4

| **sec** | Section [0] | Section [1] | Section [2] | Section [3] | Section [4] | Section [5] |
|---|---|---|---|---|---|---|
| **Name** | **.text** | **.rdata** | **.data** | **.pdata** | **.didat** | **.rsrc** |
| **Raw-size in bytes** | 122368 bytes | 69120 bytes | 4096 bytes | 8192 bytes | 512 bytes | 2560 bytes |
| **Virtual size in bytes** | 122880 bytes | 69632 bytes | 8192 bytes | 8192 bytes | **4096** bytes | 2080 bytes |
| **Entry point .txt** | 0x0001 AF30 | --- | ---- | --- | ---- | --- |
| **Raw add begin** | 0x00000 400 | 0x0001 E200 | 0x0002 F000 | 0x00030 000 | 0x00032 000 | 0x00032 200 |
| **Raw add end** | 0x0001 E200 | 0x0002 F000 | 0x00032 000 | 0x00032 200 | 0x00032 200 | 0x00032 C00 |




## 9- Count of resource:

- There are three resources in file C4