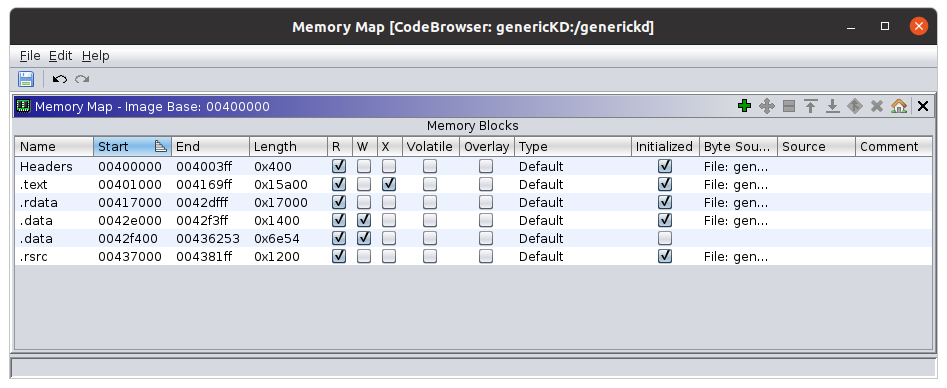
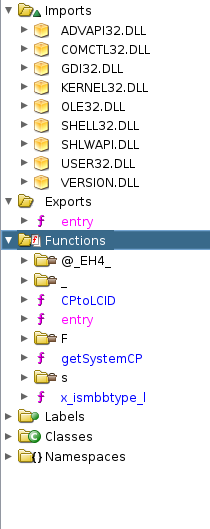
***1.Trojan.GenericKD.3652107:***

**SECTIONS and OFFSET:**

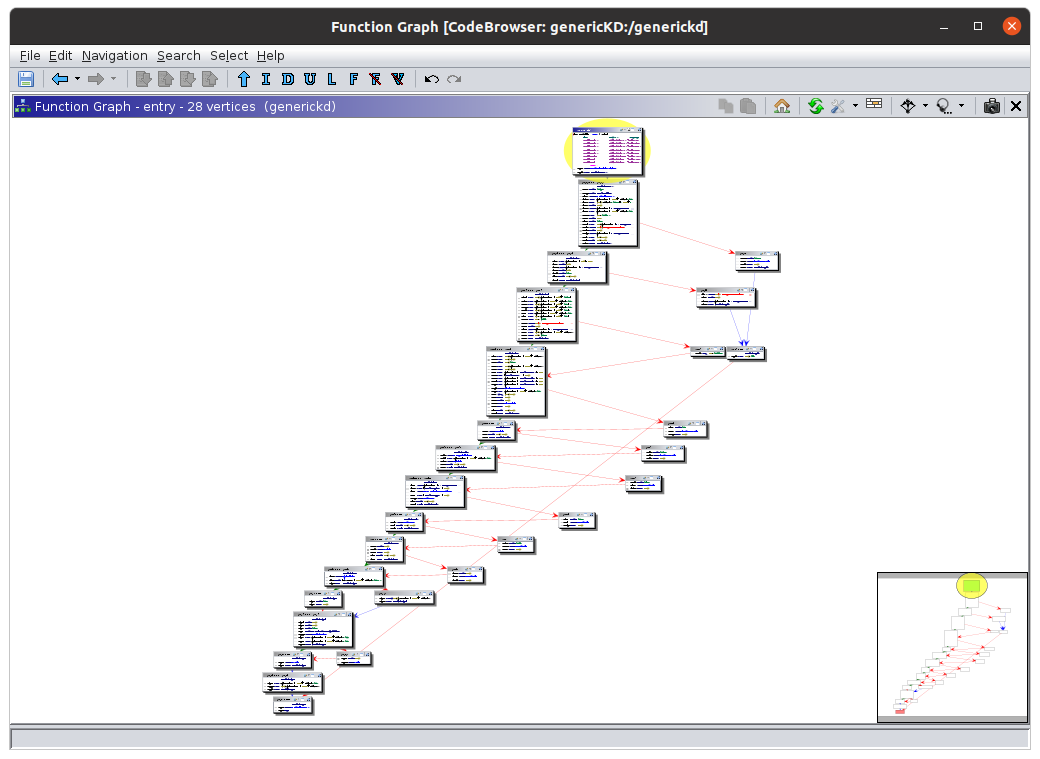


**Different functions, imports and exports of each Malware:**

****

These are the imports,exports and the functions used by the malware according to Ghidra.

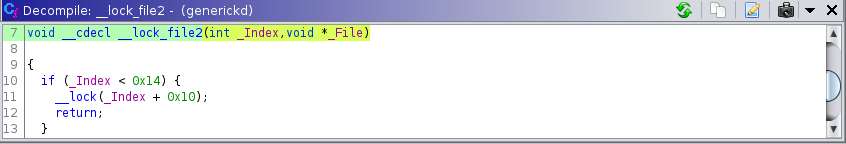
**Flow of functions:**



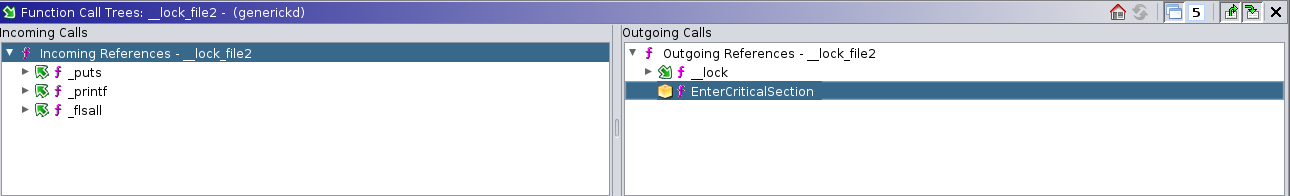
This is the flow of entry point functions according to Ghidra there are many functions that are being called that is why the names of each function is not visible in the screenshot of the Function tree graph.

Suspicious Functions:

1.



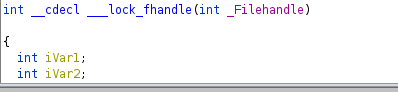
This function uses EnterCriticalSection which gives access to the malware inside a critical section of the **OS.** this may be used an advantage for the malware to hide and persist on the Machine.



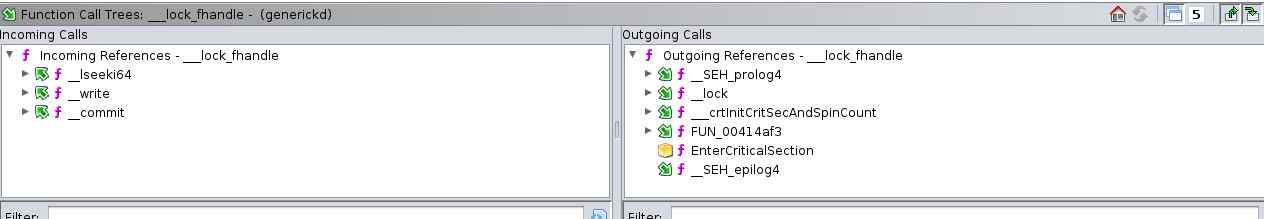
These are the incoming and the outgoing function calls.

There are two more functions that use EnterCriticalSection

2.



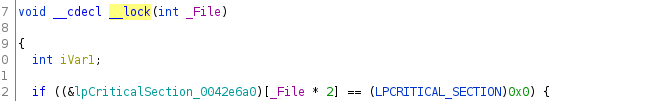
It takes one integer type parameter.



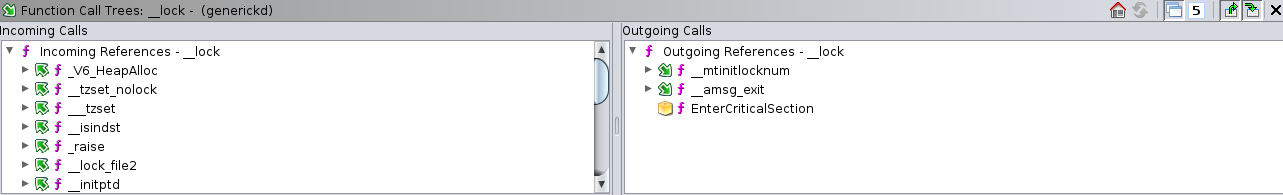
Incoming and outgoing function calls

The other one is;

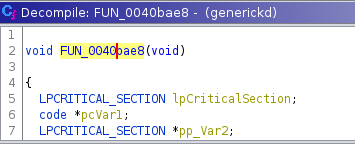
3.



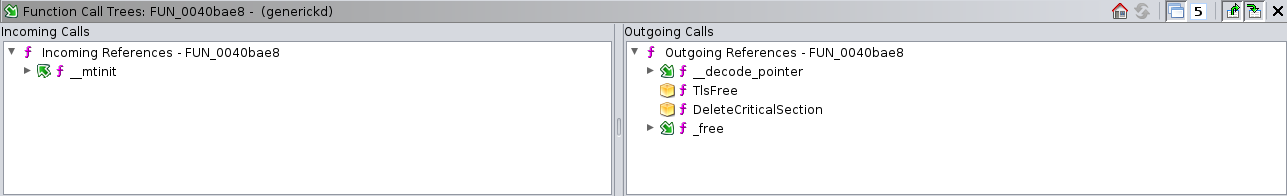
Int type parameter



4.

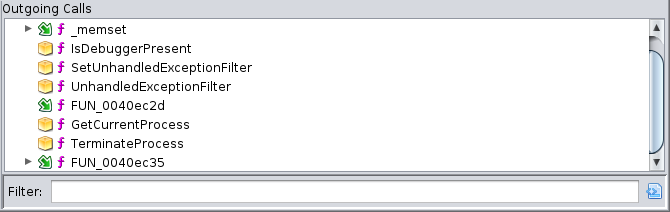


This function does not take any parameter, uses a library function DeleteCriticalSection.



5.

**\_\_invoke\_watson(wchar\_t \*param\_1,wchar\_t \*param\_2,wchar\_t \*param\_3,uint param\_4,uintptr\_t param\_5)**



Firstly it is checking if the debugger is present and than it calls another function if the debugger is not present to terminate a process

**hProcess = GetCurrentProcess();**

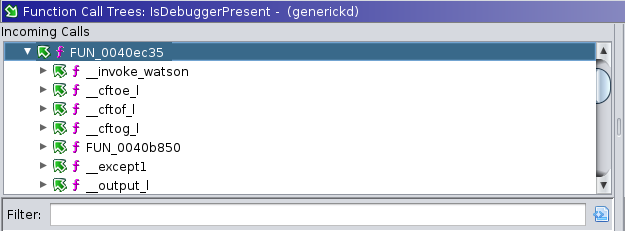
**TerminateProcess(hProcess,uExitCode);**

There is another function that is checking for a debugger and is setting parameters for Invoke\_Watson function

6.

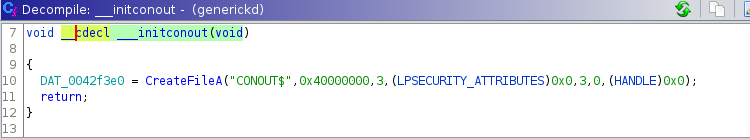
**void \_\_fastcall FUN\_0040ec35(int param\_1,undefined4 param\_2,undefined param\_3)**

This is being called from Invoke\_watson

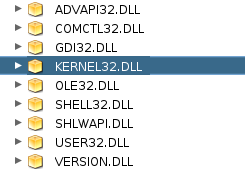


**\_\_cdecl \_\_\_initconout(void)**

This function is creating a file with security attributes

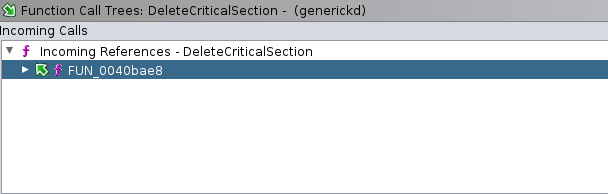


**Dynamic link libraries:**

****

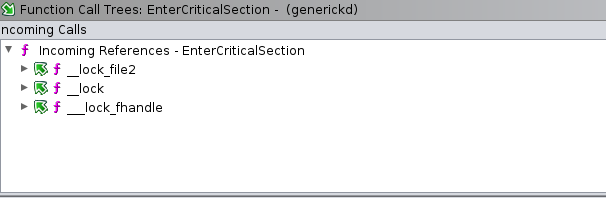
These are the Dlls included in the malware.

**ComCTL32** is used for cnc but it is not used by any function in the Malware.

**DeleteCriticalSection by Kernal32.DLL:**

There is a function in **Kernal32.dll** Named **DeleteCriticalSection**  which is used to Release all resources used by an unowned critical section object. It is being used in our Malware inside a function FUN\_004bae8 which makes the function and the dll both suspicious.

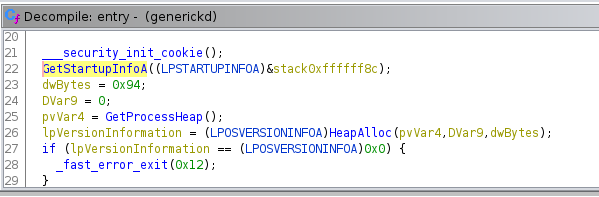
**EnterCriticalSection by Kernal32.dll:**

****

There is a function in **Kernal32.dll** Named **EnterCriticalSection**  which is used to Wait for ownership of the specified critical section object. The function returns when the calling thread is granted ownership. It is being used in our Malware inside a function Three different functions.

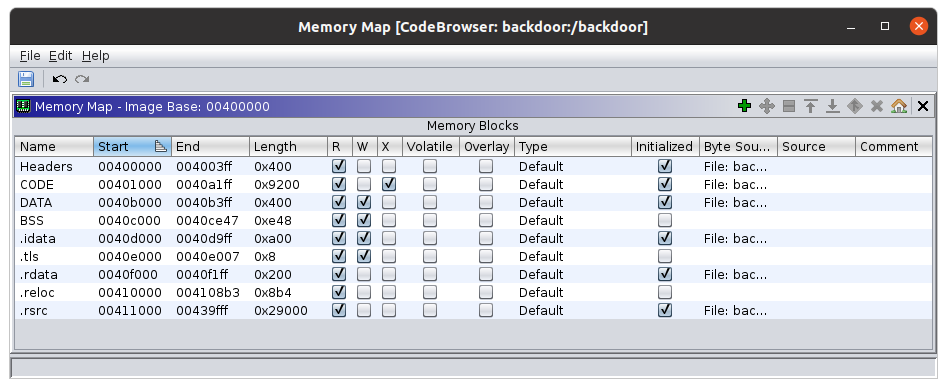
There is a function used named **STARTUPINFOA** which is used toSpecify the window station, desktop, standard handles, and appearance of the main window for a process at creation time.

This may help the process to know the other processes running on start up and it may be able to protect or hide it self.

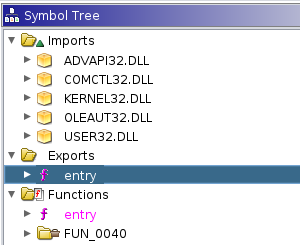


***2)Backdoor.TXLK-8101:***

**SECTIONS and OFFSET:**

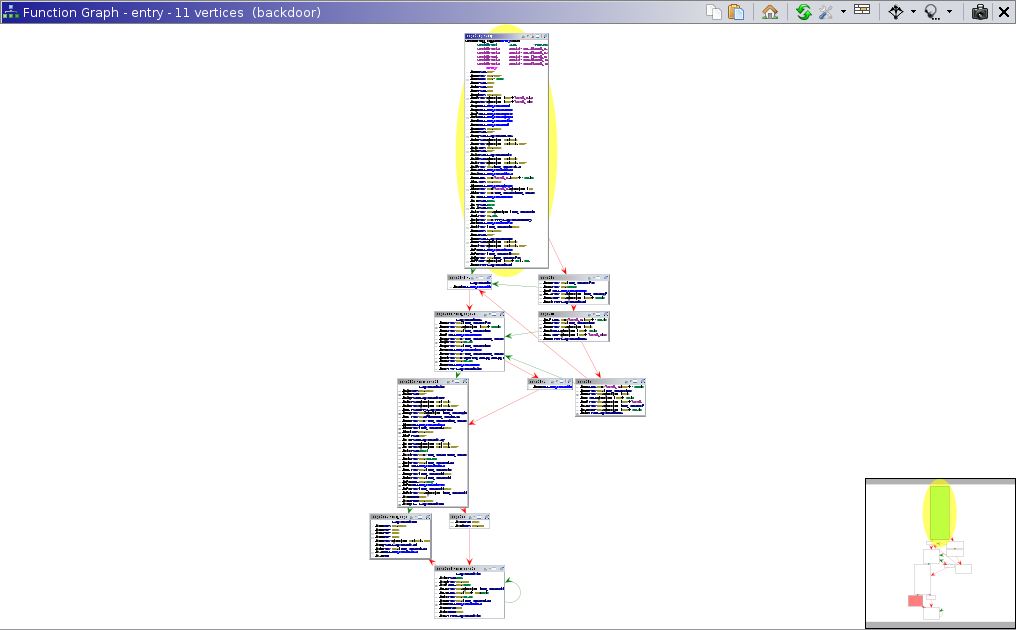


**Different functions, imports and exports of each Malware:**



These are the imports,exports and the functions used by the malware according to Ghidra.

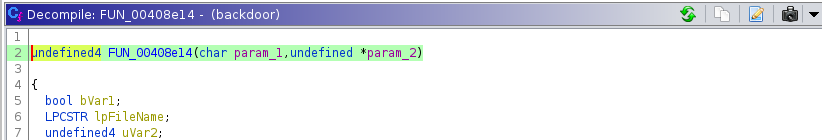
**Flow of functions:**

****

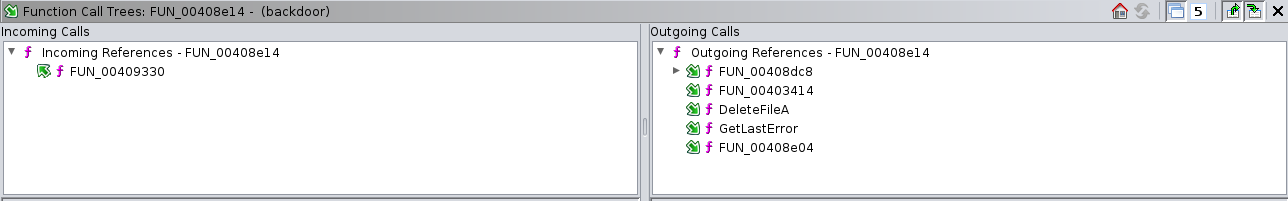
This is the flow of entry point functions according to Ghidra there are many functions that are being called that is why the names of each function is not visible in the screenshot of the Function tree graph.

Suspicious Functions:

1.



This function is undefined and has two parameters char type and the other undefined I found this function to be suspicious because it is using a function named DeleteFileA which is used to delete a file, It may delete an Important file.



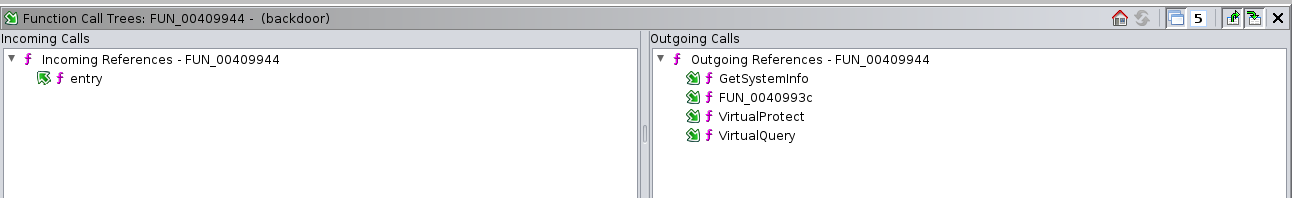
Incoming and outgoing function calls.

**2.**



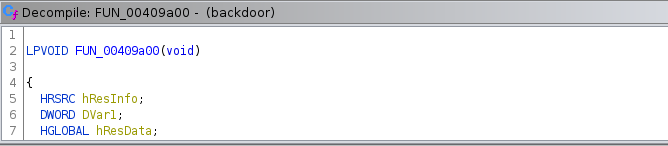
There is only one parameter of void type.

This function is suspicious because it uses a call from kernal32.dll to getsysteminfo, after getting that info it may, send manipulate to its own advantage or use it against the system in any way.



Incoming and outgoing function calls.

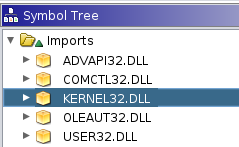
**3.**

****

The function is accessing the the rescr, by a predefined function known as LoadRescorce

If the malware accesses the resource it is usually to drop or move a file onto the disk which may harm the **OS.**

**Dynamic link libraries:**



These are the DLLs used by the malware The **COMCTL32.dll** is used to communicate through a network, so is suspicious as it may be used by the attacker to control the malware functionalities.

Kernel32.dll helps the software to run commands in privileged mode so as we have seen in the many suspicious functions are used of the **KERNAL32.Dll**  so this may be also suspicious.

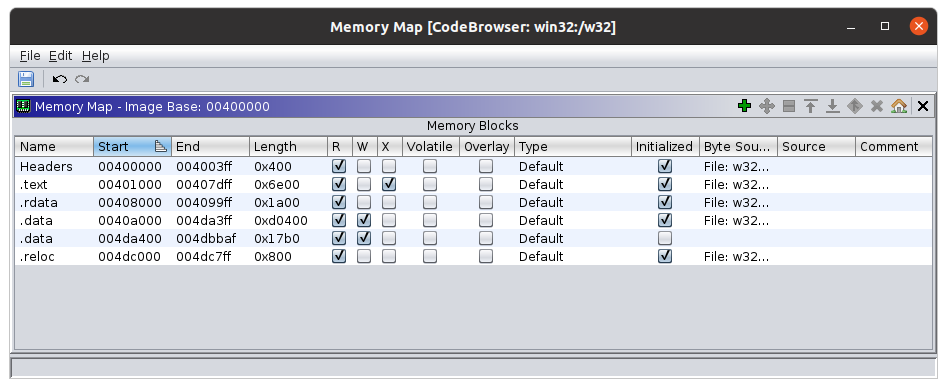
For example DeleteCriticalSection is used

There is a function in **Kernal32.dll** Named **DeleteCriticalSection**  which is used to Release all resources used by an unowned critical section object. It is being used in our Malware inside a function FUN\_004019dc(void) which makes the function and the dll both suspicious.



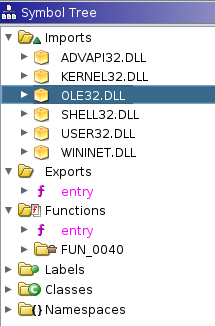
***3)W32.SecretKAN.Trojan:***

**Sections and Offset:**

****

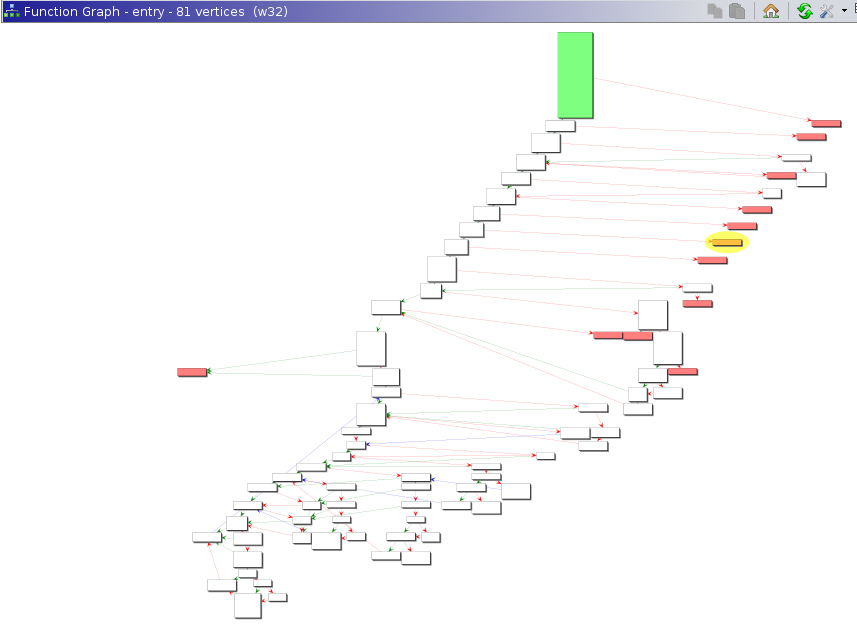
The above screenshot shows the sections of the malware and their offset with the permissions also shown.

**Different functions, imports and exports of each Malware:**

****

These are the imports, exports and the functions used by the malware.

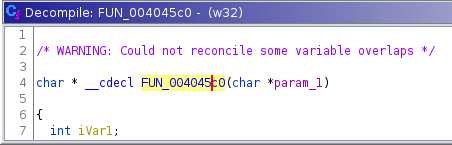
**Flow of functions:**



This is the flow of entry point functions according to Ghidra there are many functions that are being called that is why the names of each function is not visible in the screenshot of the Function tree graph.

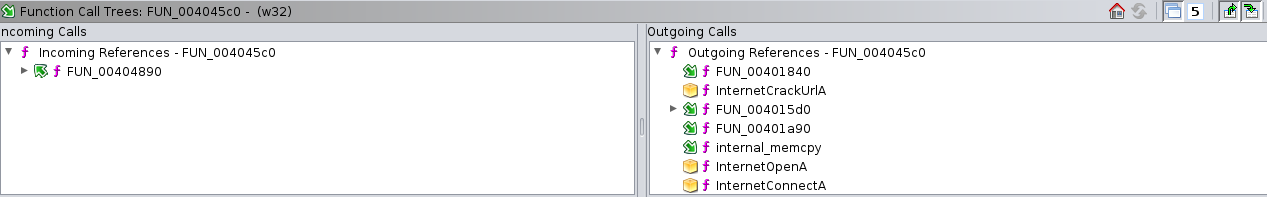
Suspicious Functions:

1.



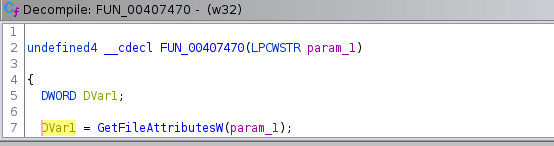
This function is is taking one character from parameter.

And is using a function named HttpOpenRequestA from WININET.dll which is used to create an HTTP request handle. And may be further used to transfer data to the attacker or may be s=used by the attacker to give certain commands to the malware.



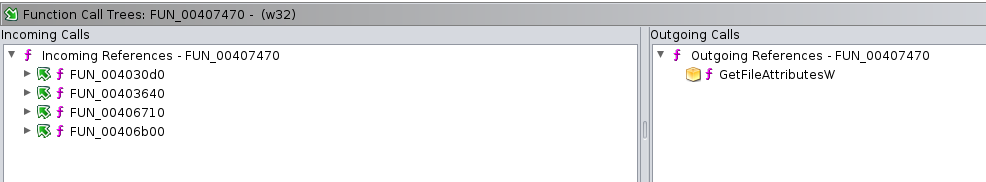
The incoming and out going function calls from the function.

**2.**

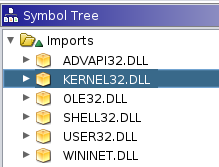


This function is using GetFileAttributes and takes 1 undefined argument,

**This** function is used to get attributes of a file by knowing the attributes it can change the file functionalities and can cause security risks.



**Dynamic link libraries:**

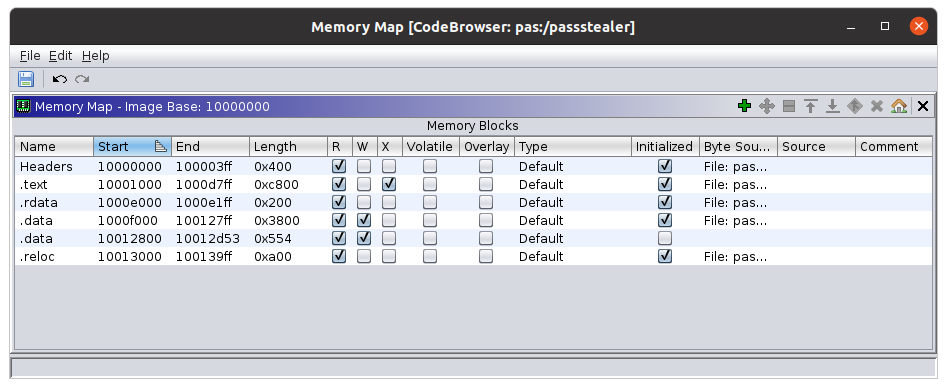


These are the DLLs used by the malware The **WININET.dll** is used to communicate through a HTML network, so is suspicious as it may be used by the attacker to control the malware functionalities.

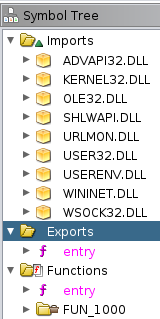
Kernel32.dll helps the software to run commands in privileged mode so as we have seen in the many suspicious functions are used of the **KERNAL32.Dll**  so this may be also suspicious.

***3)Password-Stealer:***

**Sections and offset:**

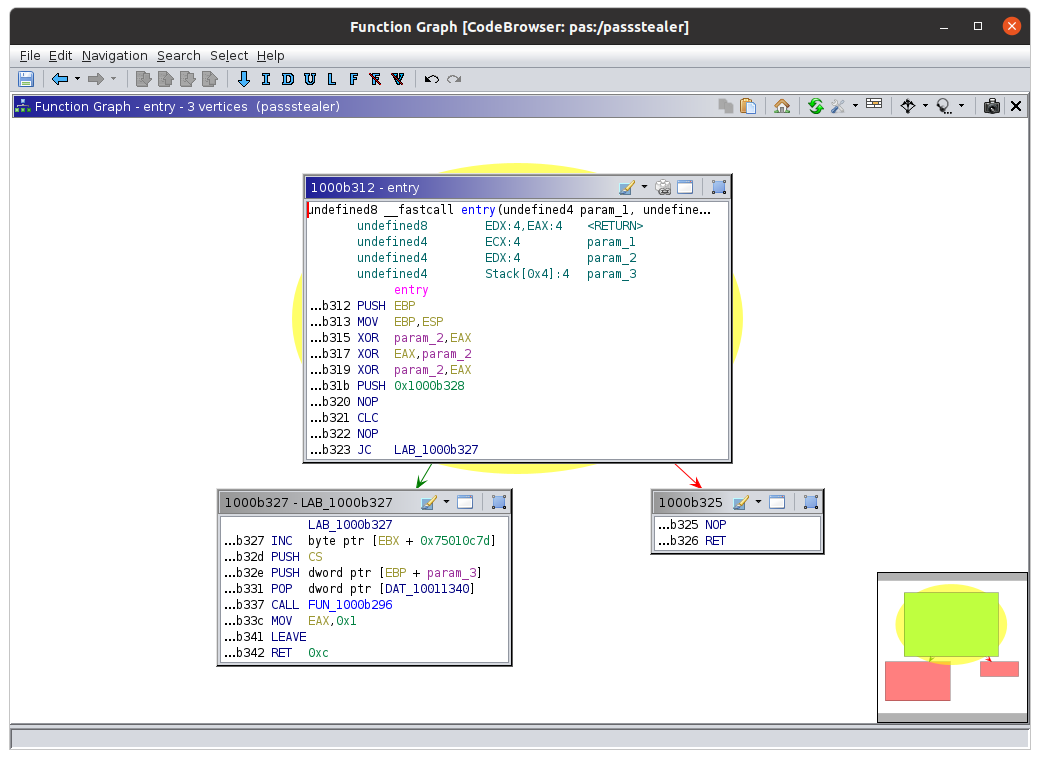


**Different functions, imports and exports of each Malware:**

****

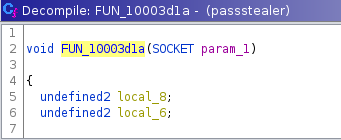
These are the imports, exports and the functions used by the malware.

**Flow of functions:**

****

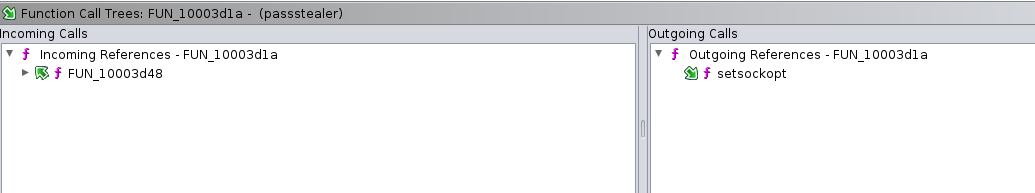
**Suspicious Functions:**

**1.**

****

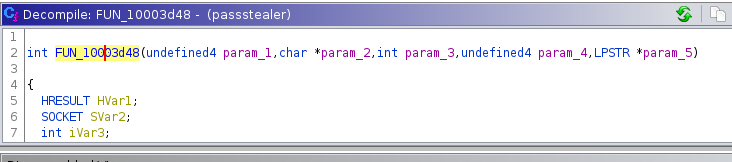
This is taking a undefined parameter

This is using setsockopt function which may be used to change the port and may the machine more vulnerable.



These are the function incoming and outgoing calls.

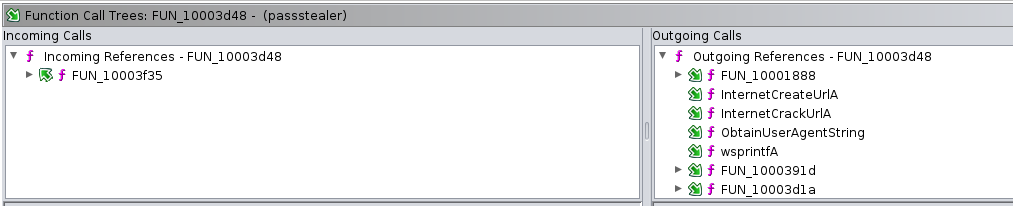
**2.**

****

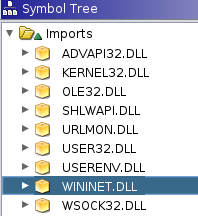
It is taking 5 input parameters

It is creating passes to communicate with the attacker by the help of function InternetCreateUrl function.

It may help the malware to communicate.



**Dynamic link libraries:**

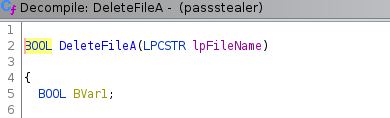
****

There are 9 dlls used in the malware

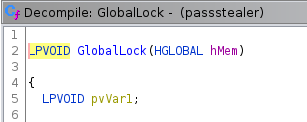
The suspicious dlls are WININET, KERNEL32 and WSOCK32 these have functions built into them which give the malware privileged accesses to make changes to the **OS.**

**As from kernal32 dll**

**DeleteFileA is used which is used to delete files it may be used to hide the original malware after creating a copy of the malware.**

****

**One more example could be**

**.**