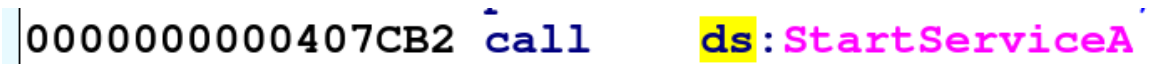
- Find real world use-cases or examples of logical addresses being used.

One example can be found when using IDA Pro to analyze the Wannacry sample (24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c).



In the address 0x407CB2 the function StartServiceA is called. In this case, IDA Pro uses the logical address (segment_selector:effective_address) in its disassembly.

In the Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 2 (2A, 2B, 2C & 2D): Instruction Set Reference, A-Z, for the "call" instruction (page 3-122) we can find:

9A *cp*	CALL *ptr16:32*	D	Invalid	Valid	Call far, absolute, address given in operand.

There's one option for the call that follows the ptr16:32 syntax. It's for "far" calls. What does it mean?

Far Call — A call to a procedure located in a different segment than the current code segment, sometimes referred to as an inter-segment call.

If we read the section about "Far Calls in Protected Mode" we find:

*"The target operand specifies an **absolute far address** either directly with a pointer (ptr16:16 ptr16:32) or indirectly with a memory location (m16:16 or m16:32)."*

We can see how they reference this type of addresses "absolute far address".

In the Intel® 64 and IA-32 Architectures Software Developer's Manual, Combined Volumes if we search for "logical address" we can find in page Vol. 1 3-9 how a logical address is also defined with the "far pointer" naming convention:

When using 32-bit addressing, a logical address (or far pointer) consists of a 16-bit segment selector and a 32-bit offset; when using 16-bit addressing, an address consists of a 16-bit segment selector and a 16-bit offset.

Far address, far pointer, logical address are all the same. And we saw how IDA Pro used it in that example.

However, the lesson to be learned is that most of the times, assembly instructions assume the segment selector part of the logical address so it does not need to specifically be specified. That's why seeing logical address is less common.

For example, a *jmp* instruction assumes that the execution flow will continue in the same CS (Code Segment). The same happens with the *push* instruction and the SS (Stack Segment).

On the other hand, Windows uses the FS segment to point to the PEB data structure within a process. The logical address FS:30h points to the PEB structure. By using this pointer, many malware samples load libraries and resolve imports.