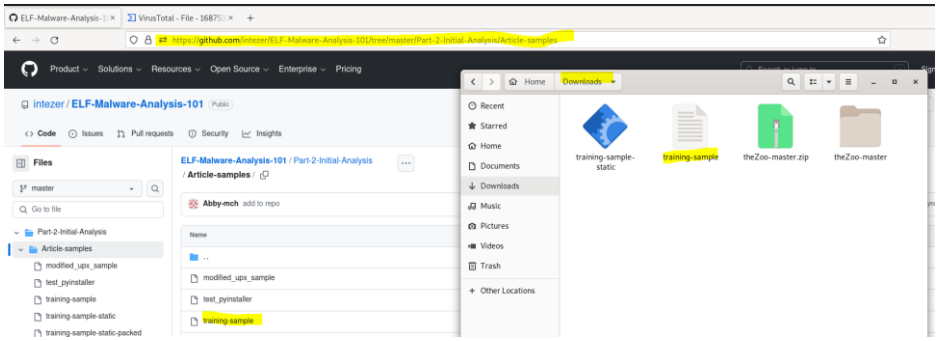
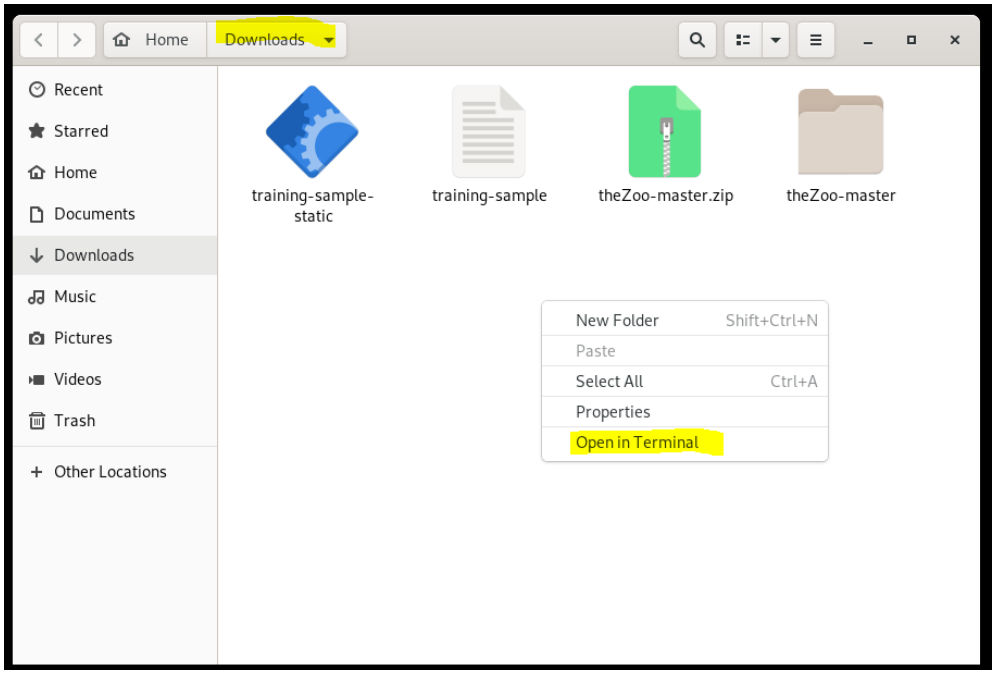
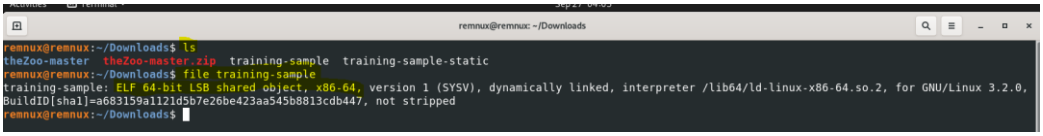
Static analysis using REMnux

- **Machine**: Launch the REMnux machine
- **Tools**: file, die(Detect it Easy), md5sum, readelf
- **Ransomware** – sample from (not a malware) https://github.com/intezer/ELF-Malware-Analysis-101/tree/master/Part-2-Initial-Analysis
- Launch the browser and use the above link to download the sample files – training-sample



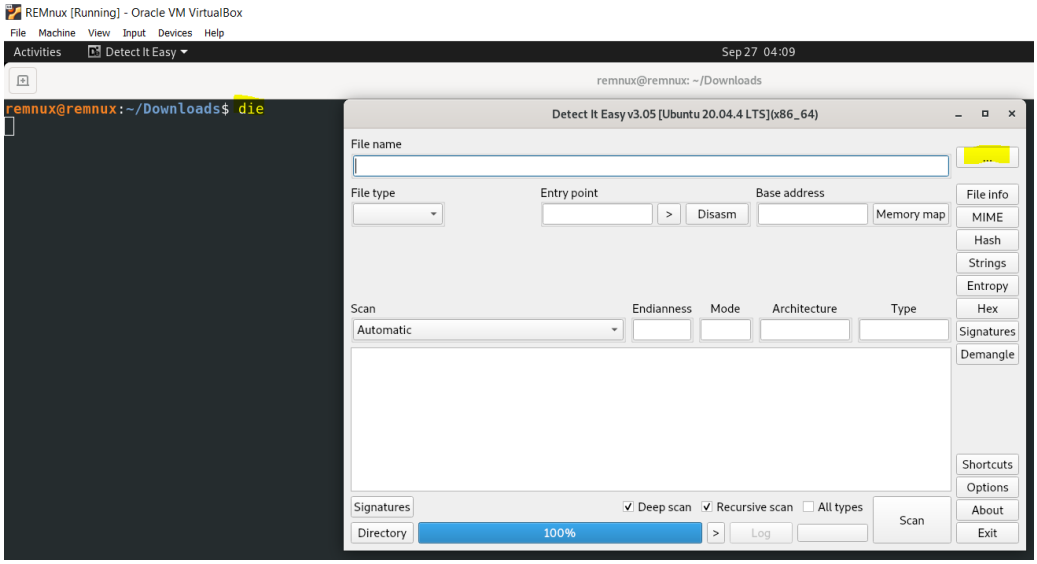
- Open the terminal and use the tool - file to identify the type of the file

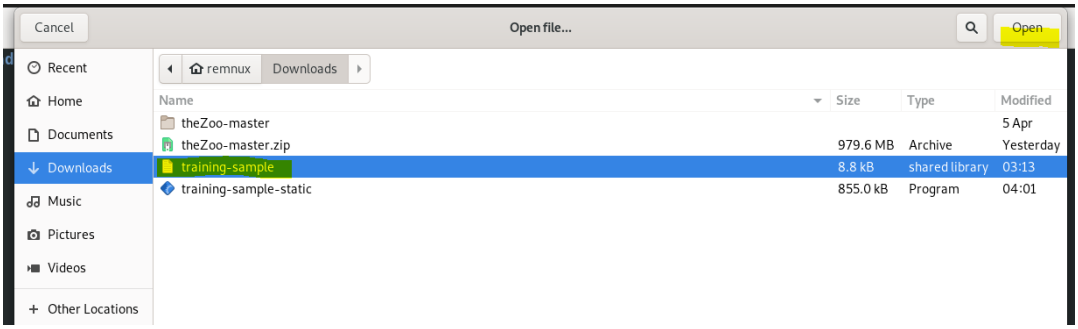


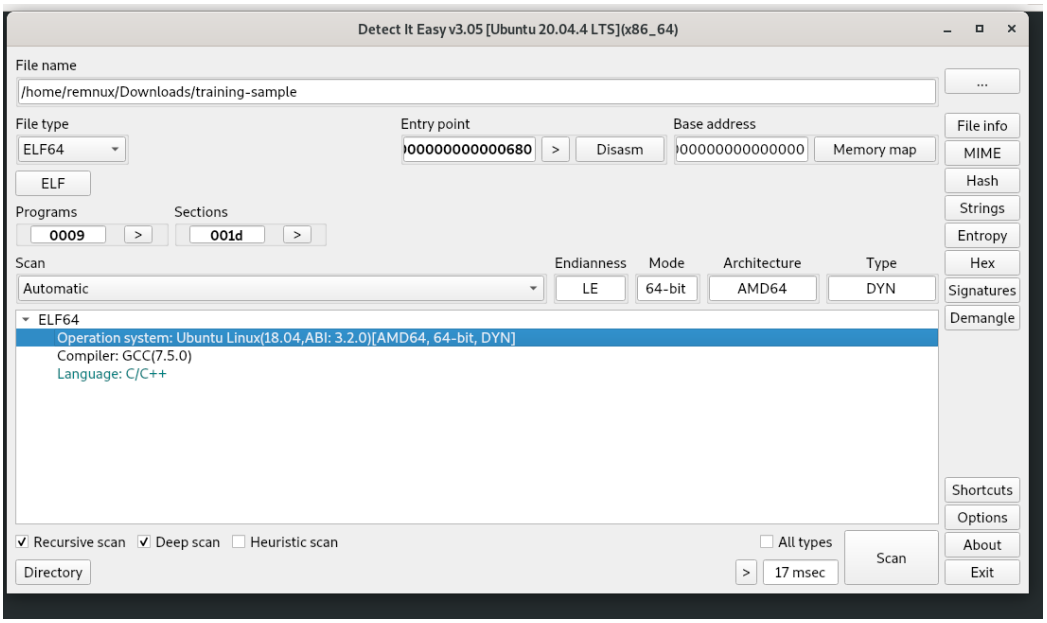


- **Observations**: ELF – a binary file that can run on Linux

- Use the tool die(Detect it Easy) to know more about this file

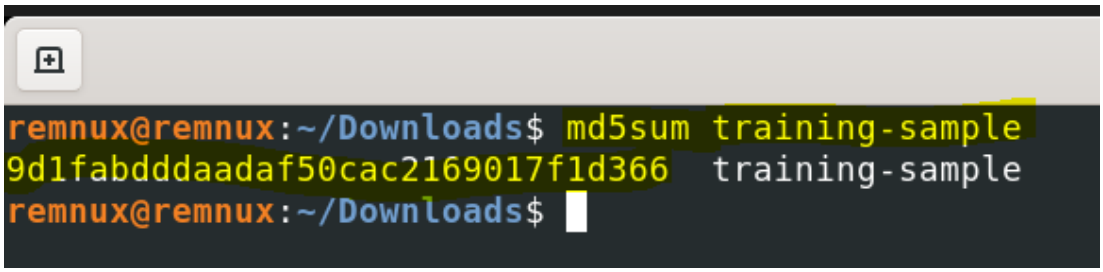


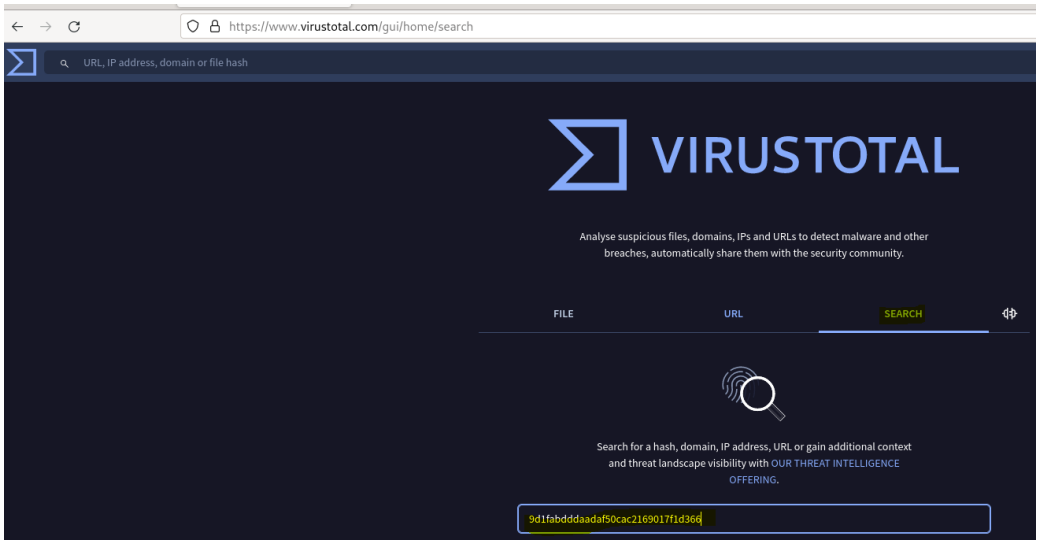


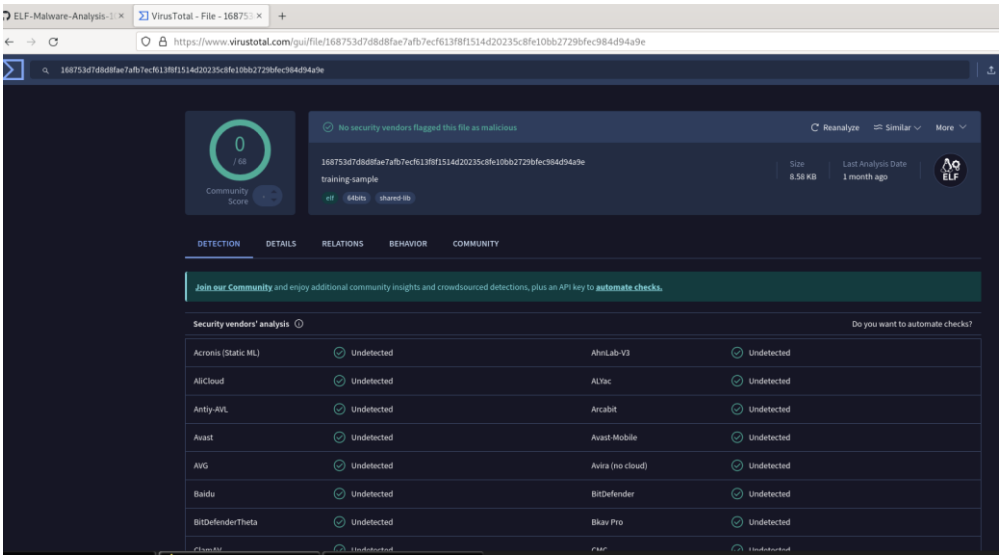


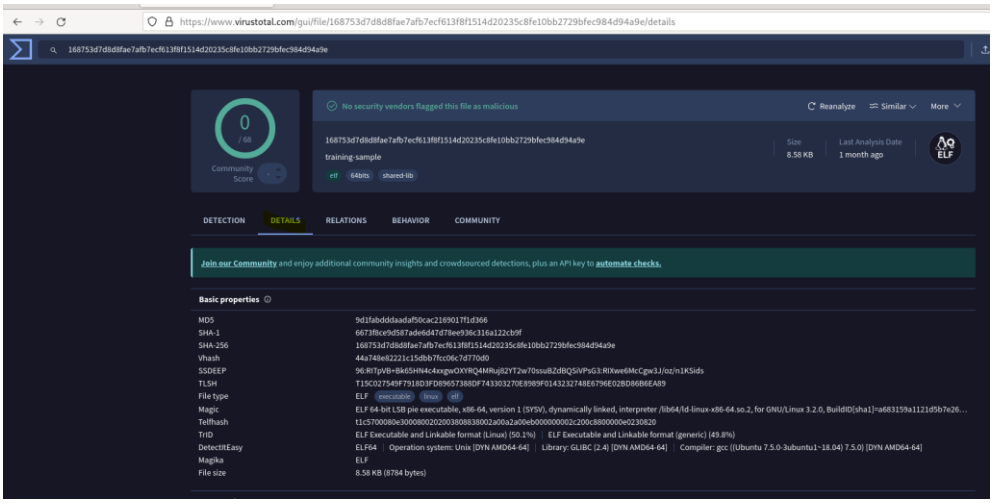
- **Observations**: type of OS, name of the compiler and name of the programming language used can be detected

- Check if the malware file has already been analyzed. Find the hash of the malware sample and check in the virustotal web site



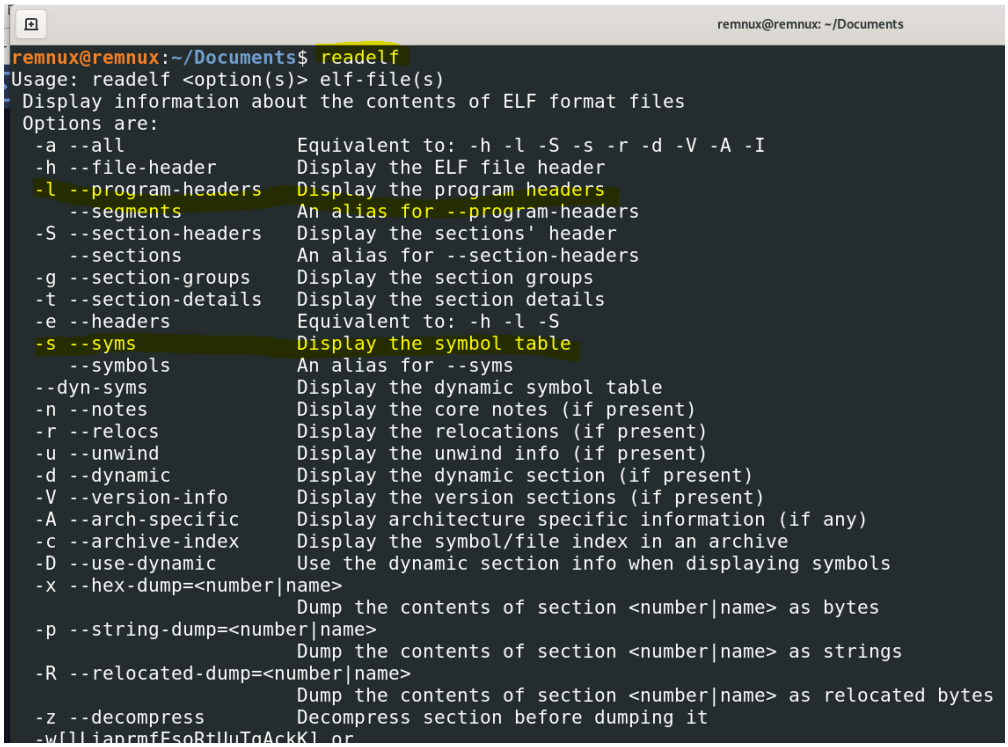


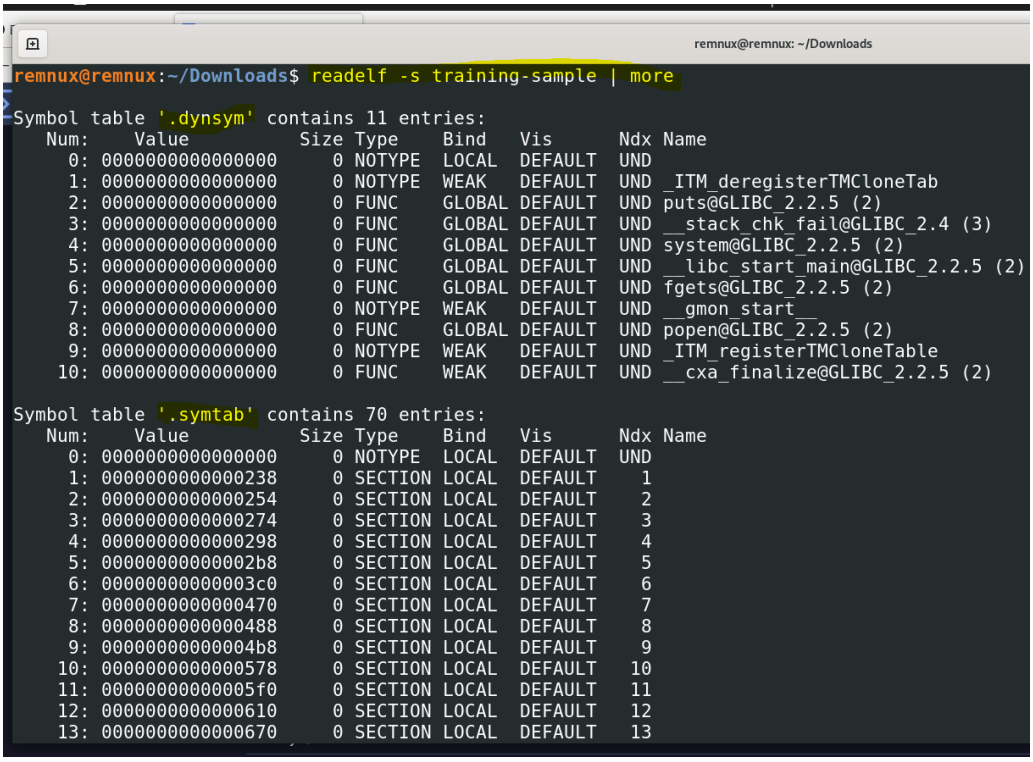




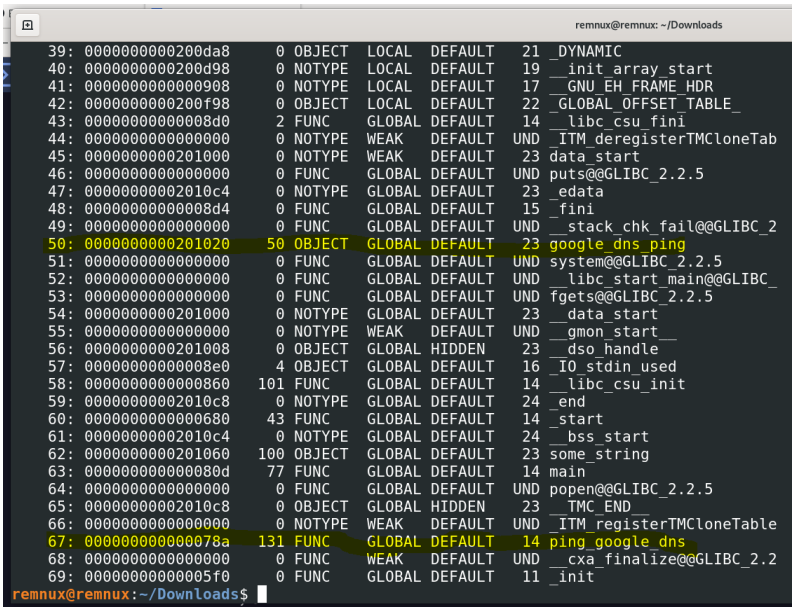
- **Observation**: not a malware (since the file is written for training purpose). Click the details tab and look for useful information

- To analyse the ELF binary, use the tool – readelf

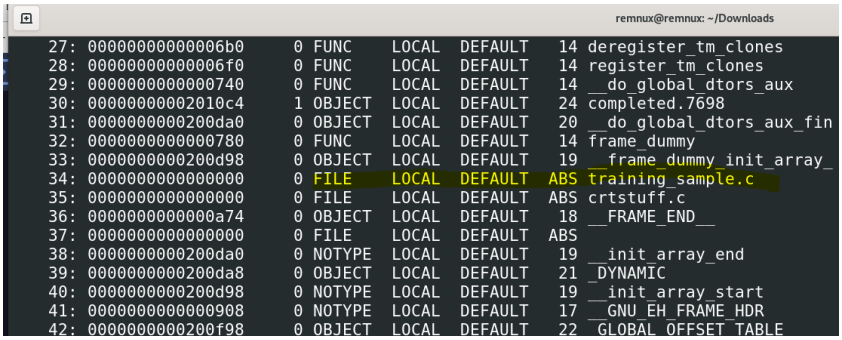




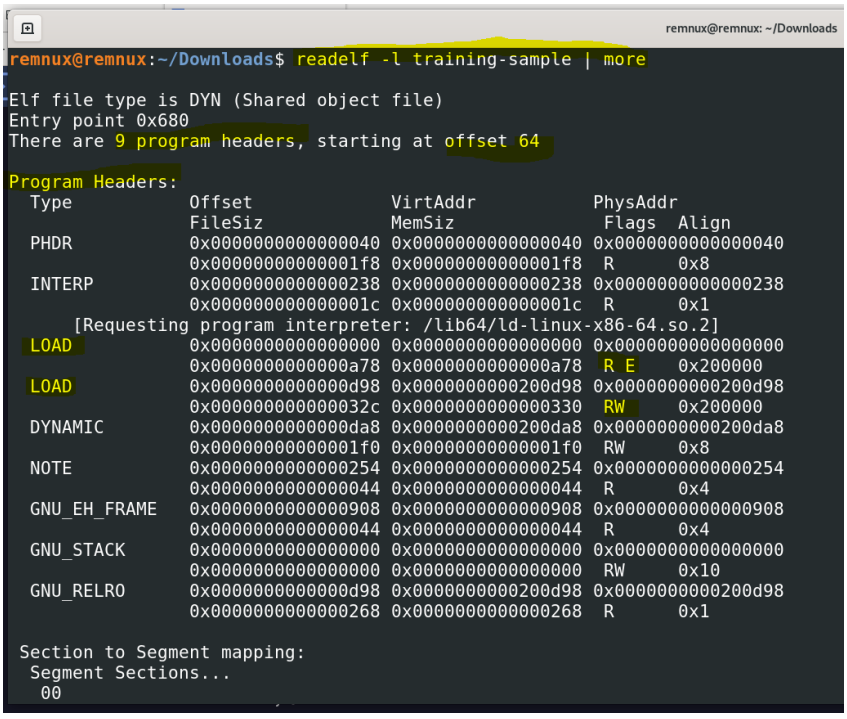
- Symbols helps to uncover the name of the functions and source codes used in a malware.
- .dynsym – contains symbols that are dynamically linked
- .symtab – contains symbols that are actual functions. Looking into the name we may be able to identify the function's task at this point



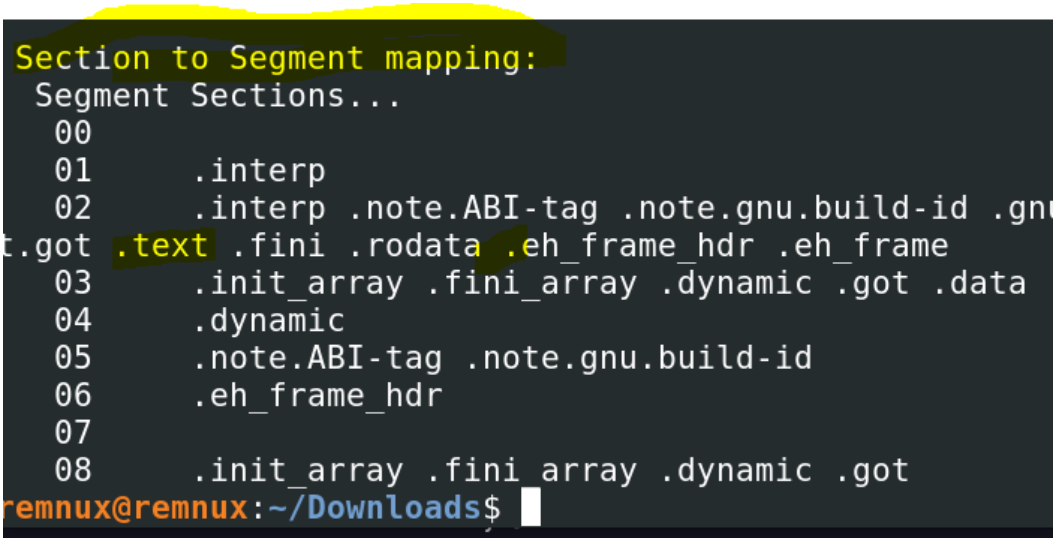
- File – name of the source code file



- Analyse the program headers and sections



- **Observations**: look for LOAD – actual location where the actual executable code is loaded. The code with RE permission will execute the malware. The code with RW stores file's global variable data and dynamic linking library is stored.
- Analyse the section to segment mapping



- **Observation:** .text section contains the executable code.