PRACTICAL 1

AIM: Static analysis of malware samples using disassemblers such as IDA Free.

Tools Required:

1. IDA Pro (Free Version)
2. A sample malware binary (e.g., DarkComet)
3. A safe analysis environment (Virtual Machine)

Procedure:

Step 1: Setting Up a Safe Environment.

1. Use a virtual machine with no network access to analyze malware safely.
2. Install IDA Free Version on the analysis system.
3. Obtain a known malware sample for analysis.(e.g., DarkComet)

Step 2: Loading the Malware in IDA Free Version.

1. Open IDA Free Version and load the malware executable.
2. Let the disassembler analyze the binary and generate assembly code.

Step 3: Identifying Key Components.

1. Look for the main function or entry point of the malware.
2. Identify API calls related to file access, network comm's, or registry changes.
3. Check for suspicious strings, obfuscation techniques, and encryption routines.

Step 4: Understanding the Malware's Behavior.

1. Analyze function calls and code flow to determine the malware's intent.
2. Identify hardcoded IP addresses, domains, or commands that indicate communication with a C2 server.
3. Check for anti-debugging or anti-analysis techniques used by the malware.

Step 5: Documenting Findings.

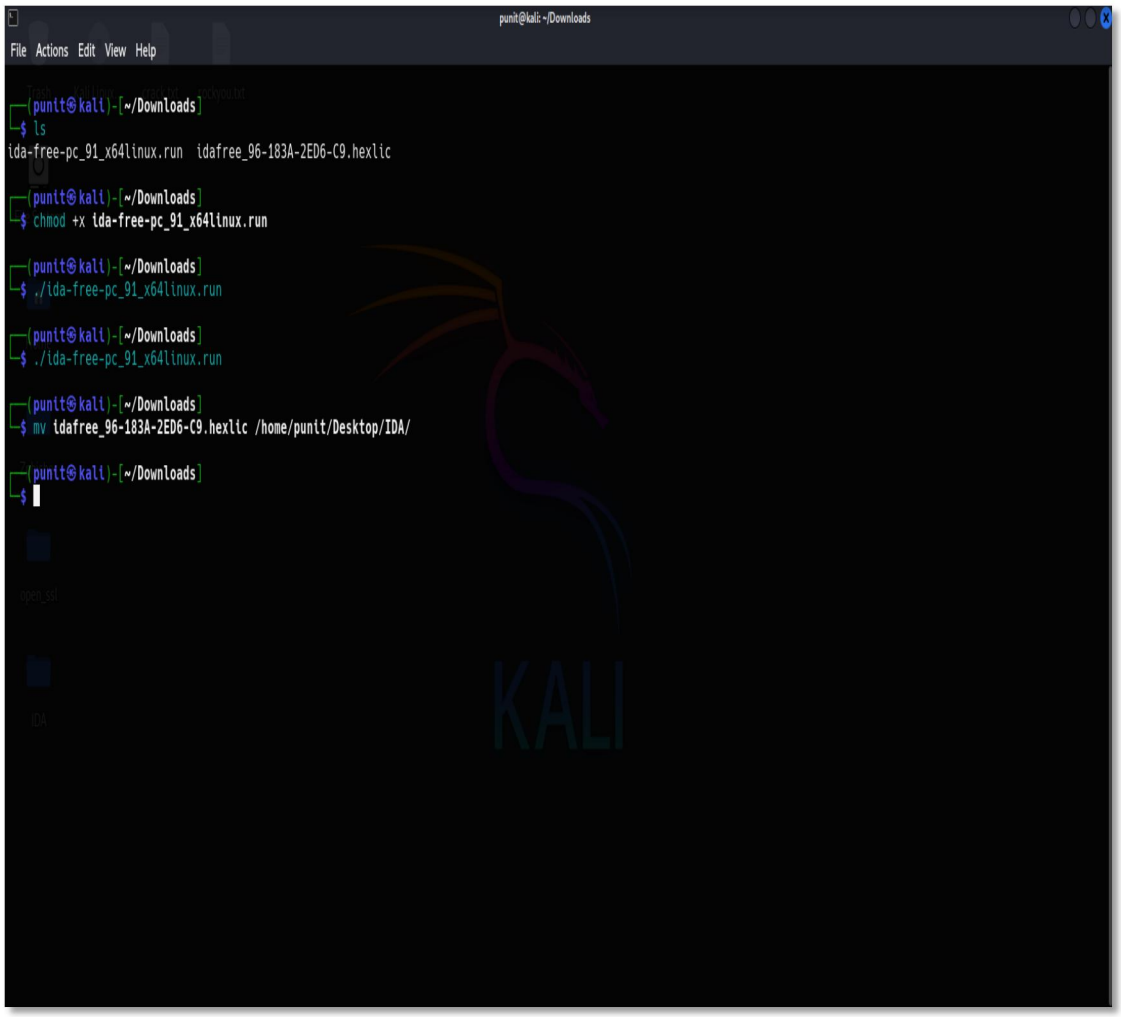
1. Summarize the malware's capabilities (e.g., keylogging, file encryption, data exfiltration).
2. Report the indicators of compromise (IOCs) such as file hashes, API calls, and network connections.
3. Suggest mitigation techniques to prevent infection and spread.

Practicla Demo

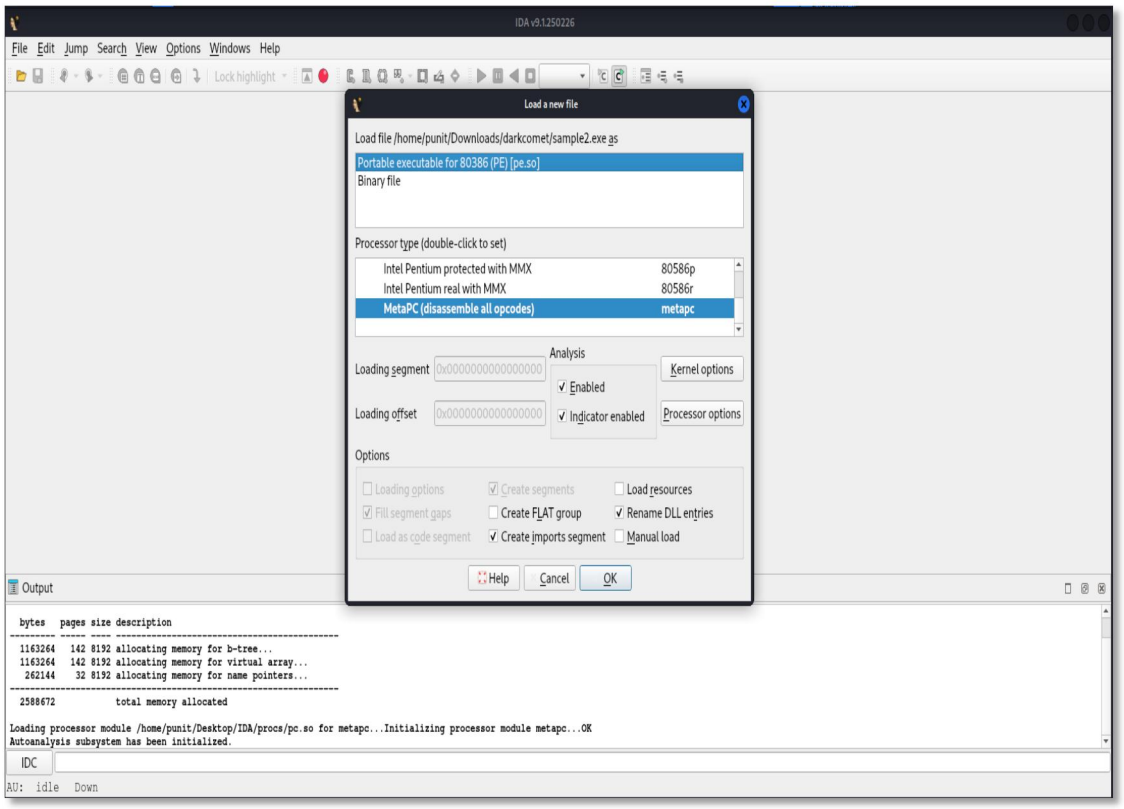
Step 1: Download and Install IDA Free Version Into Kali Linux and It's License File and Put That File Into That IDA Installed Folder To Detect and Run IDA Free Version.

Command To Run The IDA Software:

./ida



Step 2: After Running The IDA Tool Choose New File and Select The Malware Sample For Static Analysis. (**Darkcomet -> Sample2.exe**)



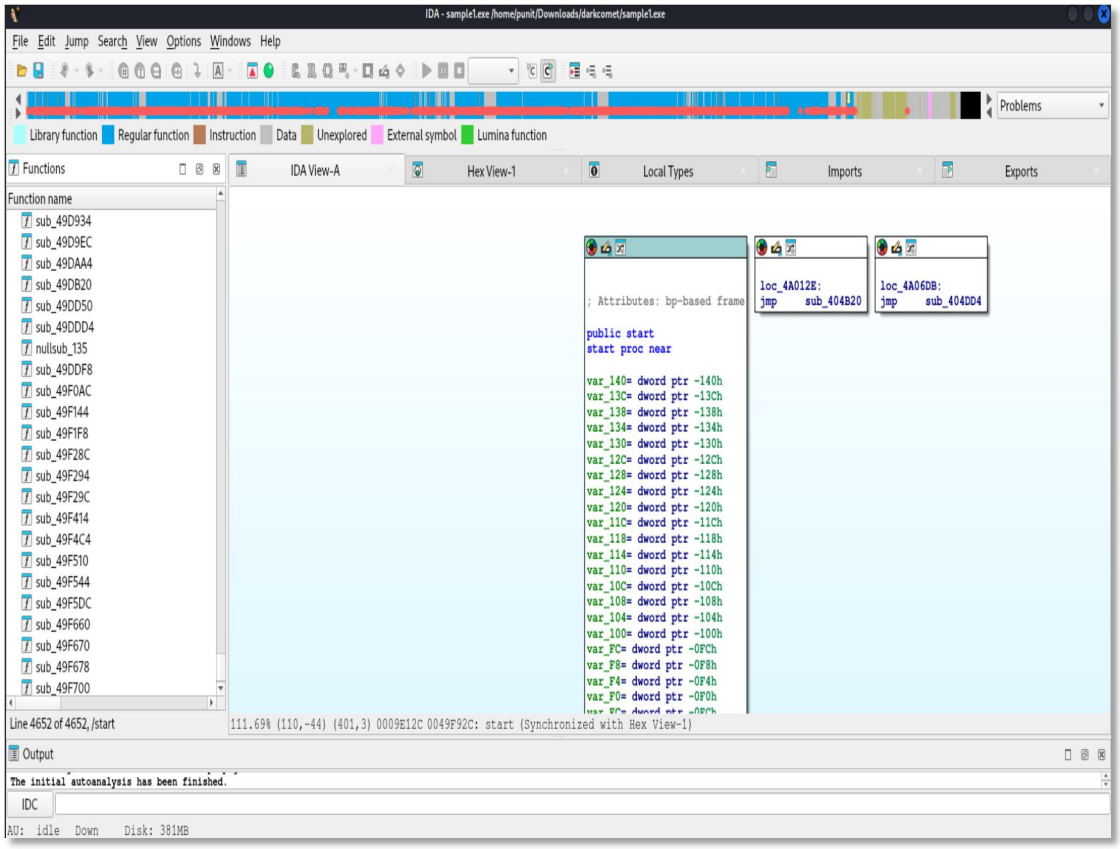
OutPut:

```
Directory /tmp/ida does not exist, creating...
Possible file format: Portable executable for 80386 (PE) (/home/punit/Desktop/IDA/loaders/pe.so)

  bytes   pages size description
--------- ----- ---- --------------------------------------------
  1163264   142 8192 allocating memory for b-tree...
  1163264   142 8192 allocating memory for virtual array...
   262144    32 8192 allocating memory for name pointers...
-------------------------------------------------------------------
  2588672           total memory allocated

Loading processor module /home/punit/Desktop/IDA/procs/pc.so for metapc...Initializing processor module metapc...OK
Autoanalysis subsystem has been initialized.
Loading file '/home/punit/Downloads/darkcomet/sample2.exe' into database...
Detected file format: Portable executable for 80386 (PE)
Assuming __cdecl calling convention by default
  0. Creating a new segment  (00401000-00481000) ... ... OK
  1. Creating a new segment  (00481000-004C7000) ... ... OK
Reading imports directory...
Type library 'mssdk_win7' loaded. Applying types...
Types applied to 0 names.
Plan  FLIRT signature: SEH for vc7-14
Marking typical code sequences...
Flushing buffers, please wait...ok
File '/home/punit/Downloads/darkcomet/sample2.exe' has been successfully loaded into the database.
Flushing buffers, please wait...ok
Hex-Rays Cloud Decompiler plugin has been loaded (v9.1.0.250226)
  The decompilation hotkey is F5.
  Please check the Edit/Plugins menu for more information.
Using FLIRT signature: SEH for vc7-14
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
```

Step 3: After Scan The Malware Sample Into IDA Free Version You Get The Function's and The IDA View For Analysis Into Assembly Language and Hexa-View and You Can See The Graph and Connected Node's That Show's The Working of The Executable File and The Code Structure and Functionality of The Executable File.



Conclusion:

Disassemblers like IDA Free Version play a vital role in malware analysis, offering deep insights into the code structure and functionality. While static analysis is effective in identifying malicious behavior, combining it with other techniques, including dynamic analysis, provides a more complete picture of the threat landscape.