**Cyber Security – Assessment # 8**

**Q1**. Static Malware Analysis is the process of analyzing a malicious file (malware) without executing it. This approach focuses on studying the file's structure, behavior, and characteristics in a safe, controlled environment to identify its purpose, capabilities, and potential impact. Static analysis helps researchers and security professionals detect malware even before it runs on an infected system. Using appropriate tools, find out following: -

* File Type
* Embedded Strings

**Answer**

Static malware analysis is a fundamental approach to identifying threats without executing potentially harmful code. To determine the **file type** and **embedded strings**, here are some tools and techniques you can use:

**1. Identifying File Type**

* **File Command (Linux/macOS)**: Run file <filename> to examine the file’s format.
* **PEStudio (Windows)**: Provides insights into executable files, including headers and structures.
* **ExifTool**: Useful for checking metadata and file format information.
* **Binwalk**: Helps in identifying embedded files and extracting firmware data.

**2. Extracting Embedded Strings**

* **Strings Command (Linux/Windows)**: Run strings <filename> to extract readable text from binary files.
* **Ghidra/IDA Pro**: Advanced reverse engineering tools that help analyze the binary for strings and code structures.
* **Floss**: A tool specifically designed to extract obfuscated strings from malware.
* **Hex Editors (HxD, Hex Workshop)**: Allows manual inspection of data within a file.

These signatures are located at the beginning of a file (file header) and can be checked using tools like Hex Editors, Binwalk, or TrID.

|  |  |  |
| --- | --- | --- |
| **File Type** | **Signature (Hexadecimal)** | **Description** |
| **Executable (PE)** | 4D 5A (MZ) | Windows Portable Executable (EXE, DLL, SYS) files. |
| **PDF** | 25 50 44 46 (%PDF) | Adobe Portable Document Format files. |
| **ZIP Archive** | 50 4B 03 04 (PK..) | Compressed archive format used for packaging files. |
| **RAR Archive** | 52 61 72 21 1A 07 00 (Rar!) | Proprietary archive format. |
| **JPEG Image** | FF D8 FF E0 | Common image format used for pictures. |
| **PNG Image** | 89 50 4E 47 (‰PNG) | Lossless image format often used online. |
| **MS Word Doc (Old)** | D0 CF 11 E0 A1 B1 1A E1 | Microsoft Office document (DOC, XLS, PPT - pre-2007). |
| **MS Word Doc (New)** | 50 4B 03 04 (PK..) | Microsoft Office document (DOCX, XLSX, PPTX - since 2007, ZIP-based). |
| **ELF Executable** | 7F 45 4C 46 (ELF) | Linux/Unix Executable and Linkable Format. |
| **ISO Disk Image** | 43 44 30 30 31 (CD001) | Optical disc image format. |

**1. Using HxD (Hex Editor)**

HxD is a powerful hex editor that allows you to view and search for readable strings inside files.

**Steps:**

1. Open **HxD**.
2. Load the **target file (e.g., ZIP, EXE, DLL, etc.)**.
3. Use **Ctrl + F** and set **Search Type** to "Text-string."
4. Enter keywords or search for human-readable text within the binary.
5. You can also view encoded strings and manually analyze them.

**2. Using CFF Explorer**

CFF Explorer is a tool designed for executable files, allowing deeper analysis including strings extraction.

**Steps:**

1. Open **CFF Explorer** and load the file.
2. Navigate to **"String Viewer"** under the file structure section.
3. Browse through detected strings, including those in executables or DLLs.
4. Copy important strings for further investigation.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **File Type** | **Embedded Strings** |
| **1** | Sample-Lab-3-1-6 | OBJ | .drectve  .debug$S  B.data  .text$mn  P` /DEFAULTLIB:"LIBCMT" /DEFAULTLIB:"OLDNAMES"  C:\cygwin\home\poona\Sample-Lab-3-1-5.obj  Microsoft (R) Optimizing Compiler  Hello Malwares  @comp.id}y  @feat.00  .drectve  .debug$S  .data  $SG1327  .text$mn  \_dummy  \_printf |
| **2** | Sample-Lab-3-1-9 | Microsoft Word | [Content\_Types].xml  l"%3  ^i7+  %p)O  5}nH"  t4Q+  |T\y  \_rels/.rels  jH[{  l0/%  word/document.xml  Smr)  eOiI  FK5.@  Rsla  wZ$G  <=TG  YfXAI  WRn]  HO+Q |
| **3** | Sample-Lab-3-1-8 | HTML | <html>  <h1>Hello Malwares</h1>  </html> |
| **4** | Sample-Lab-3-1-10 | Microsoft PowerPoint | [Content\_Types].xml  N+PU  cHT<JQ  {dS5r  25=d  \_rels/.rels  hD.Cy  (H[s  [:b4  S7t}N!  y}4)  ppt/presentation.xml  D@#N  n5J(Q  y\*;B  3lOo  5jJ&  pD6'  ppt/slides/slide1.xml  |6M\* |
| **5** | Sample-Lab-3-1-11 | Microsoft Excel | [Content\_Types].xml  ji){^  P!XS)bR  |\|z  h\_-[  DO97\*  \_rels/.rels  BKwAH  GJy(v  USh9i  r:"y\_dl  xl/workbook.xml  eU:OLi.E  \*rZJ  NPaL  b%5@\_  LV5@  !;!X  SRm&97,O  ]NWz |