Malware Analysis & Reverse Engineering

Assignment 1:

Question: Use the strings command to extract readable text from a given malware sample. Identify any URLs, IP addresses, or commands in the output.

To extract readable text, URLs, IP addresses, and commands safely in Kali Linux for both analysis and documentation, follow these steps using built-in tools and commands:

Procedure on Kali Linux

- Place the malware sample in an isolated or non-executable directory within your analysis virtual machine
- Run the strings command to extract readable text:

text

```
strings /malware_samples/malware_sample.exe/ > sample_strings.txt
```

  - This saves all extracted strings to sample_strings.txt for documentation and further review.

- document URLs in the output:

text

```
grep -Eo '\bhttps?://[a-zA-Z0-9./?=_-]*' sample_strings.txt > urls.txt
```

  - This extracts detected URLs and saves them to urls.txt.

- For IP addresses:

text

```
grep -Eo '([0-9]{1,3}\.){3}[0-9]{1,3}' sample_strings.txt > ips.txt
```

  - This extracts IPv4 addresses into ips.txt for further examination.

- To search for common command keywords (like "cmd", "powershell"):

text

```
grep -i -E 'cmd|powershell|shell|wget|curl|ftp' sample_strings.txt > commands.txt
```

  - This saves command-related strings to commands.txt.

Safety and Documentation Tips

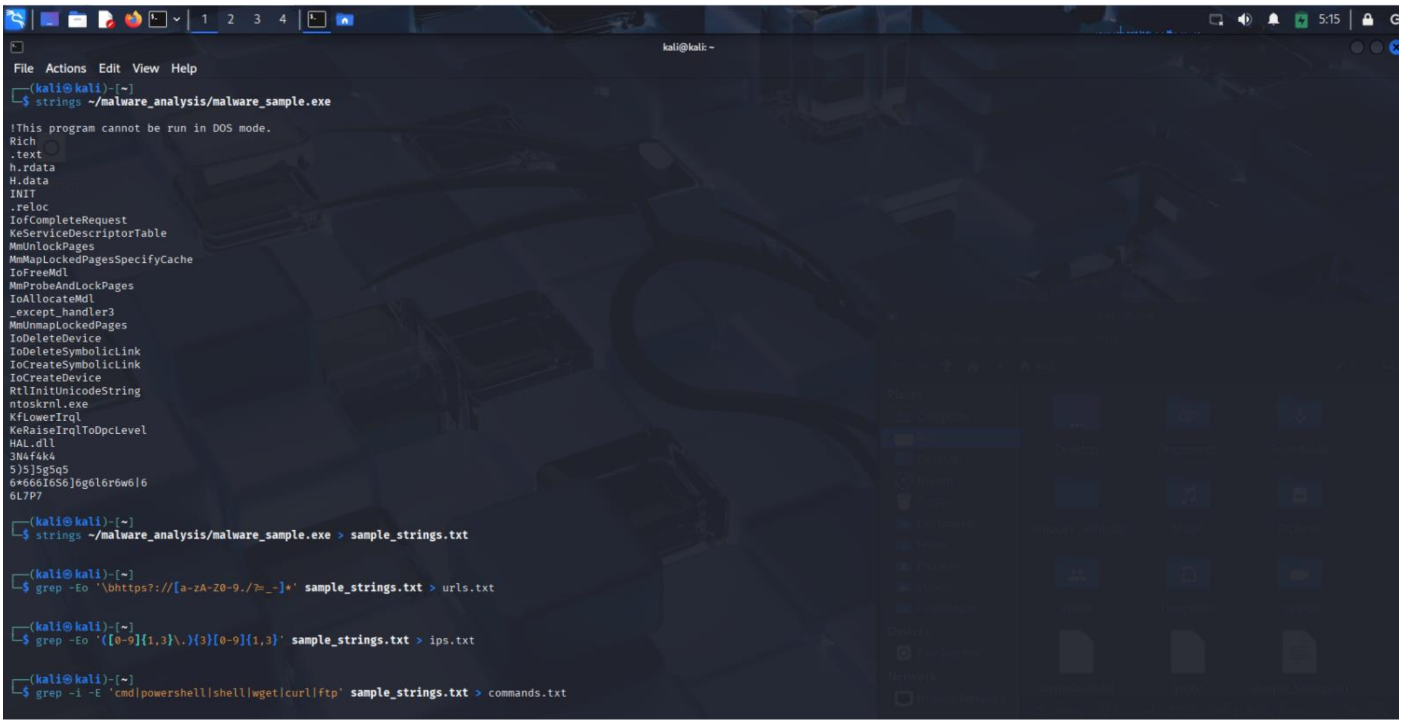
- Always analyze samples within a secure Kali Linux VM that is not connected to our main network for safety
- Document findings by saving extracted info into separate .txt files
- Create a report summarizing notable findings (URLs, IPs, and suspicious commands) for incident response or forensic review.
- Optionally, keep all shell commands and outputs in a Markdown or log file for audit trails.

Safety Notes

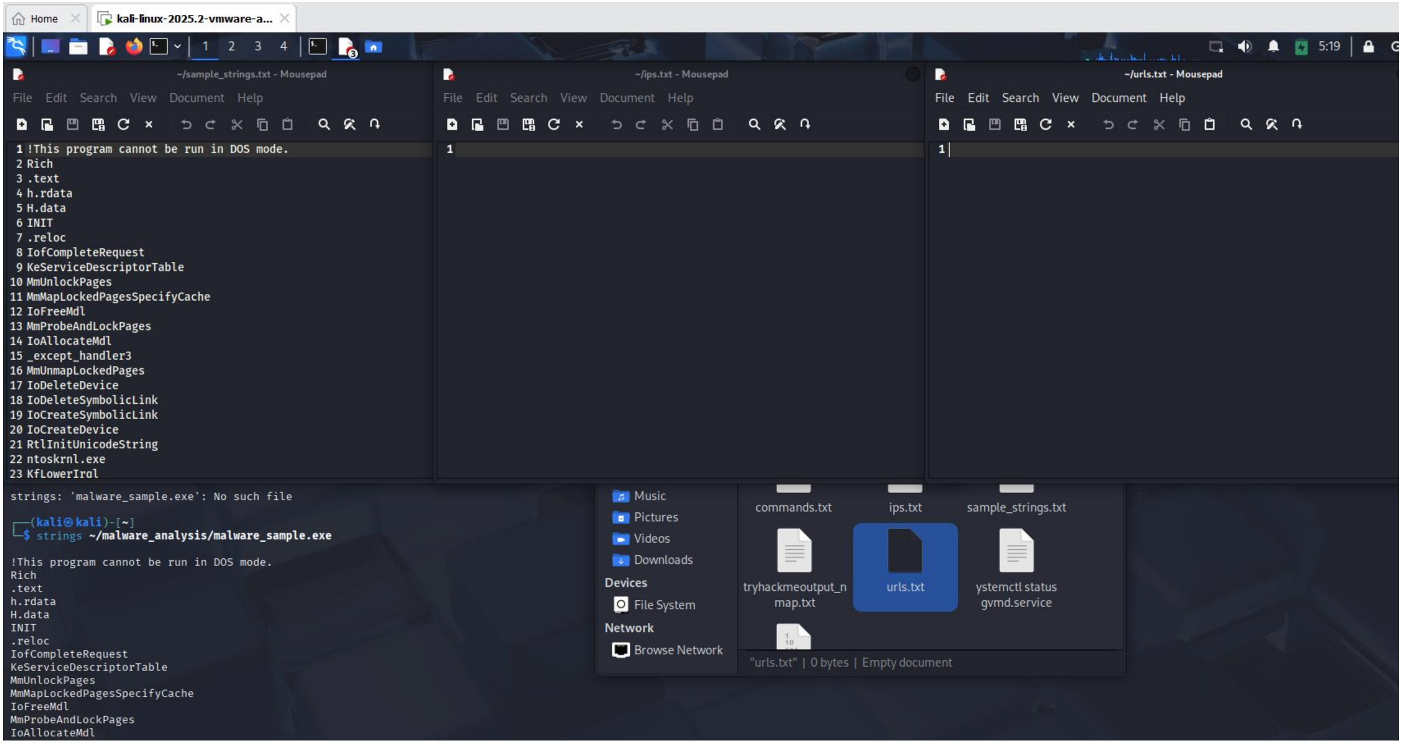
- Do all static analysis in the Kali VM, not on the host machine.
- Don't run or interact with the sample except through static tools like strings.
- Save findings in text files for evidence and documentation.

These are the findings:

sample_strings.txt ips.txt urls.txt commands.txt



```
┌──(kali㉿kali)-[~]
└─$ strings ~/malware_analysis/malware_sample.exe

!This program cannot be run in DOS mode.
Rich
.text
h.rdata
H.data
INIT
.reloc
IofCompleteRequest
KeServiceDescriptorTable
MmUnlockPages
MmMapLockedPagesSpecifyCache
IoFreeMdl
MmProbeAndLockPages
IoAllocateMdl
_except_handler3
MmUnmapLockedPages
IoDeleteDevice
IoDeleteSymbolicLink
IoCreateSymbolicLink
IoCreateDevice
RtlInitUnicodeString
ntoskrnl.exe
KfLowerIrql
KeRaiseIrqlToDpcLevel
HAL.dll
3N4f4k4
5)5]5g5q5
6*666I6S6]6g6l6r6w6|6
6L7P7

┌──(kali㉿kali)-[~]
└─$ strings ~/malware_analysis/malware_sample.exe > sample_strings.txt

┌──(kali㉿kali)-[~]
└─$ grep -Eo '\bhttps?://[a-zA-Z0-9./?=_-]*' sample_strings.txt > urls.txt

┌──(kali㉿kali)-[~]
└─$ grep -Eo '([0-9]{1,3}\.){3}[0-9]{1,3}' sample_strings.txt > ips.txt

┌──(kali㉿kali)-[~]
└─$ grep -i -E 'cmd|powershell|shell|wget|curl|ftp' sample_strings.txt > commands.txt
```



```
!This program cannot be run in DOS mode.
Rich
.text
h.rdata
H.data
INIT
.reloc
IofCompleteRequest
KeServiceDescriptorTable
MmUnlockPages
MmMapLockedPagesSpecifyCache
IoFreeMdl
MmProbeAndLockPages
IoAllocateMdl
_except_handler3
MmUnmapLockedPages
IoDeleteDevice
IoDeleteSymbolicLink
IoCreateSymbolicLink
IoCreateDevice
RtlInitUnicodeString
ntoskrnl.exe
KfLowerIrql
```

```
strings: 'malware_sample.exe': No such file

┌──(kali㉿kali)-[~]
└─$ strings ~/malware_analysis/malware_sample.exe

!This program cannot be run in DOS mode.
Rich
.text
h.rdata
H.data
INIT
.reloc
IofCompleteRequest
KeServiceDescriptorTable
MmUnlockPages
MmMapLockedPagesSpecifyCache
IoFreeMdl
MmProbeAndLockPages
IoAllocateMdl
```