

## Ex No-10 : Use Ghidra to disassemble and analyze the malware code.

### Aim:

To perform static and basic dynamic analysis of a benign sample binary using Linux command-line tools (file, strings, readelf, objdump, strace, ltrace, lsof) and document observable behaviors such as file I/O, dynamic symbol resolution, and network activity (if any).

### Tools / Environment:

- OS: Linux (Kali)
- Compiler: gcc
- Analysis tools: file, strings, readelf, objdump, nm, strace, ltrace, lsof
- Working folders: ~/ghidra\_cli\_practice/samples and ~/ghidra\_cli\_practice/output/report\_evidence

### STEP 1: Prepare workspace and Create/compile

```
File Actions Edit View Help
(kali@kali)-[~]
$ mkdir -p ~/ghidra_cli_practice/samples ~/ghidra_cli_practice/output
cd ~/ghidra_cli_practice/samples
pwd
/home/kali/ghidra_cli_practice/samples samples
(kali@kali)-[~/ghidra_cli_practice/samples]
$ nano benign_sample1.c
(kali@kali)-[~/ghidra_cli_practice/samples]
$ nano benign_sample2.c
(kali@kali)-[~/ghidra_cli_practice/samples]
$ ls -l benign_sample1.c benign_sample1
benign_sample2.c benign_sample2.c
total 44
-rwxrwxr-x 1 kali kali 16112 Oct 23 09:06 benign_sample1
-rw-rw-r-- 1 kali kali 303 Oct 23 09:20 benign_sample1.c
-rwxrwxr-x 1 kali kali 16112 Oct 23 09:06 benign_sample2
-rw-rw-r-- 1 kali kali 332 Oct 23 09:21 benign_sample2.c
-rw-rw-r-- 1 kali kali 40 Oct 23 09:17 report.txt
(kali@kali)-[~/ghidra_cli_practice/samples]
$ gcc -o benign_sample1 benign_sample1.c
gcc -o benign_sample2 benign_sample2.c -ldl
ls -l benign_sample1 benign_sample2
```

```
(kali@kali)-[~/ghidra_cli_practice/samples]
mkdir -p ~/ghidra_cli_practice/output/report_evidence
../output/* ~/ghidra_cli_practice/output/report_evidence/ 2>/dev/null || true
copy created artifact too
report.txt ~/ghidra_cli_practice/output/report_evidence/ 2>/dev/null || true
-lah ~/ghidra_cli_practice/output/report_evidence

al 48K
-rwxr-x 2 kali kali 4.0K Oct 23 09:25 .
-rwxr-x 3 kali kali 4.0K Oct 23 09:25 ..
-rw-r-- 1 kali kali 337 Oct 23 09:25 ltrace_bs1.txt
-rw-r-- 1 kali kali 7.4K Oct 23 09:25 objdump_full_bs1.txt
-rw-r-- 1 kali kali 1.6K Oct 23 09:25 objdump_main_bs1.txt
-rw-r-- 1 kali kali 979 Oct 23 09:25 readelf_header_bs1.txt
-rw-r-- 1 kali kali 3.8K Oct 23 09:25 readelf_symbols_bs1.txt
-rw-r-- 1 kali kali 40 Oct 23 09:25 report.txt
-rw-r-- 1 kali kali 81 Oct 23 09:25 sha256_bs1.txt
-rw-r-- 1 kali kali 2.8K Oct 23 09:25 strace_bs1.txt
-rw-r-- 1 kali kali 1.2K Oct 23 09:25 strings_bs1.txt

(kali@kali)-[~/ghidra_cli_practice/samples]
cd ~/ghidra_cli_practice/output/report_evidence

(kali@kali)-[~/ghidra_cli_practice/output/report_evidence]
cat > analysis_report.md << 'EOF' /samples
edoc> # Aim
analyze a benign sample binary using Linux commands (`file`, `strings`, `strace`, `ltrace`, `ps`, `lsof`) and o
rve its behavior.
Procedure
Created two sample C programs: `benign_sample1.c` and `benign_sample2.c`.
Compiled the programs:
`bash
gcc -o benign_sample1 benign_sample1.c
gcc -o benign_sample2 benign_sample2.c -ldl
edoc>
edoc> EOF
```

**STEP 2 :** Static identification & strings and Capture runtime behavior & Runs the binary in the background and stores the PID.

```
(kali@kali)-[~/ghidra_cli_practice/output/report_evidence]
$ # Go to report folder
cd ~/ghidra_cli_practice/output/report_evidence && \

# Run the binary in background and capture its PID
./benign_sample1 & PID=$! && \

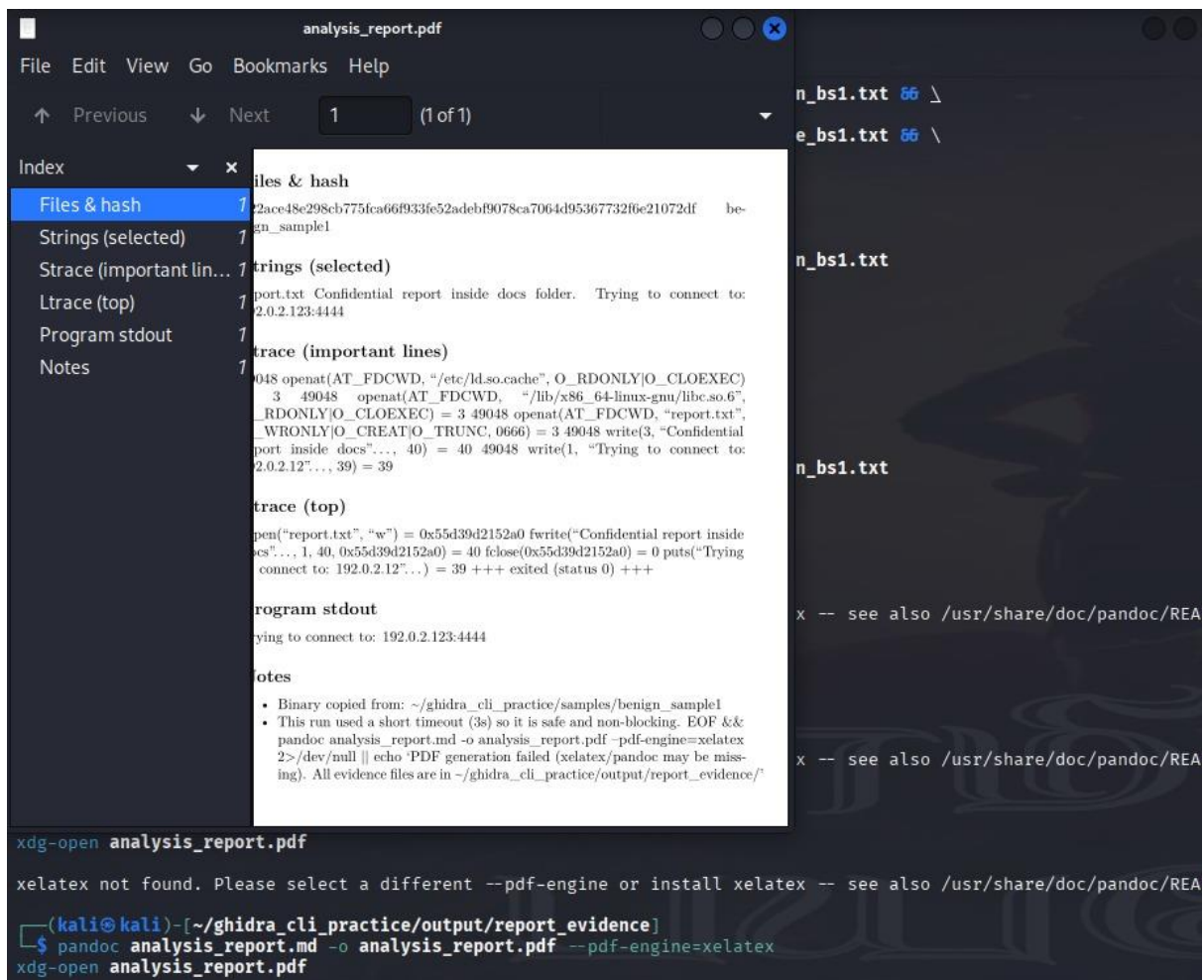
# Collect dynamic traces
strace -o strace_bs1.txt -f -p $PID & wait $PID && \
ltrace -o ltrace_bs1.txt -f -p $PID & wait $PID && \

# Collect static analysis
file benign_sample1 > file_bs1.txt && \
sha256sum benign_sample1 > sha256_bs1.txt && \
strings -n 4 benign_sample1 > strings_bs1.txt && \
readelf -h benign_sample1 > readelf_header_bs1.txt && \
readelf -s benign_sample1 > readelf_symbols_bs1.txt && \
objdump -d -M intel benign_sample1 > objdump_full_bs1.txt && \
objdump -d -M intel benign_sample1 | sed -n '/<main>:/,/^\$/p' > objdump_main_bs1.txt && \
```

**STEP 3:** Kills the binary safely after traces are done.

```
(kali@kali)-[~/ghidra_cli_practice/output/report_evidence]
$ sudo kill -9 $PID 2>/dev/null
```

**STEP 4:** Converts your Markdown report to PDF and opens it.

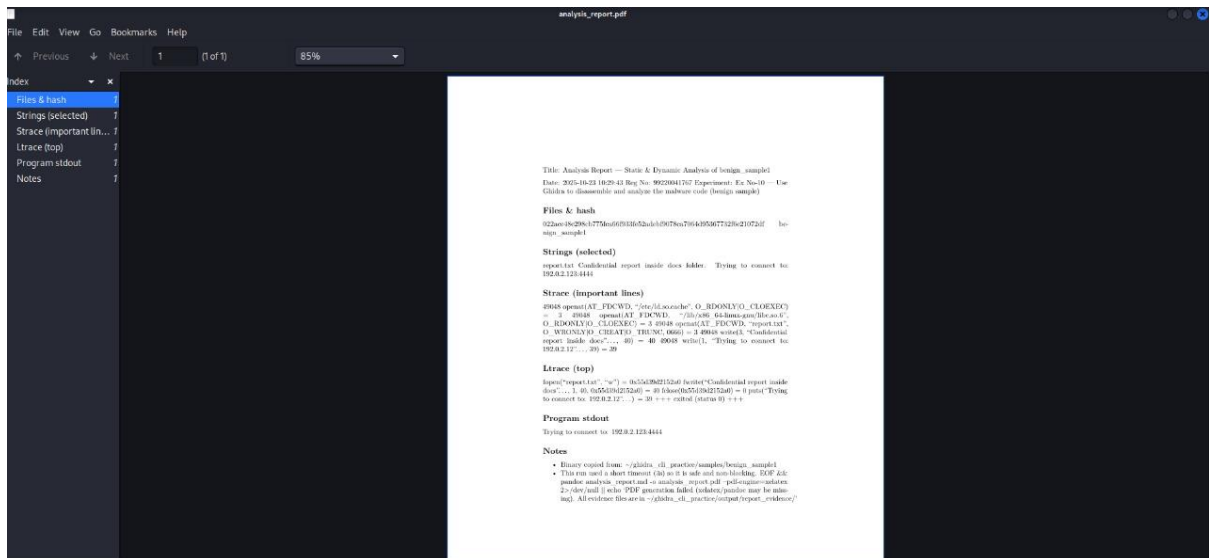


The screenshot shows a terminal window with a dark background. At the top, a window titled "analysis\_report.pdf" is open, displaying a PDF viewer interface. The left sidebar shows an "Index" with sections like "Files & hash", "Strings (selected)", "Strace (important lines)", "Ltrace (top)", "Program stdout", and "Notes". The main content area shows the "Files & hash" section, listing files and their hashes. Below this, the "Strings (selected)" section is visible, showing a list of strings. The "Strace (important lines)" section shows a list of system calls. The "Ltrace (top)" section shows a list of library calls. The "Program stdout" section shows the output of the program. The "Notes" section contains a list of notes.

```
xdg-open analysis_report.pdf

xelatex not found. Please select a different --pdf-engine or install xelatex -- see also /usr/share/doc/pandoc/REA

(kali@kali)-[~/ghidra_cli_practice/output/report_evidence]
$ pandoc analysis_report.md -o analysis_report.pdf --pdf-engine=xelatex
xdg-open analysis_report.pdf
```



## STEP 5: Appendix — Files

```
(kali@kali)-[~]
$ cd ~/ghidra_cli_practice/output/report_evidence
pwd
ls -lah

/home/kali/ghidra_cli_practice/output/report_evidence
total 132K
drwxrwxr-x 2 kali kali 4.0K Oct 23 10:31 .
drwxrwxr-x 3 kali kali 4.0K Oct 23 09:25 ..
-rw-rw-r-- 1 kali kali 1.5K Oct 23 10:29 analysis_report.md
-rw-rw-r-- 1 kali kali 64K Oct 23 10:31 analysis_report.pdf
-rwxrwxr-x 1 kali kali 16K Oct 23 10:29 benign_sample1
-rw-rw-r-- 1 kali kali 224 Oct 23 10:41 file_bs1.txt
-rw-rw-r-- 1 kali kali 0 Oct 23 10:39 ltrace_bs1.txt
-rw-rw-r-- 1 kali kali 7.4K Oct 23 10:41 objdump_full_bs1.txt
-rw-rw-r-- 1 kali kali 1.6K Oct 23 10:41 objdump_main_bs1.txt
-rw-rw-r-- 1 kali kali 39 Oct 23 10:29 prog_stdout.txt
-rw-rw-r-- 1 kali kali 979 Oct 23 10:41 readelf_header_bs1.txt
-rw-rw-r-- 1 kali kali 3.8K Oct 23 10:41 readelf_symbols_bs1.txt
-rw-rw-r-- 1 kali kali 40 Oct 23 10:38 report.txt
-rw-rw-r-- 1 kali kali 81 Oct 23 10:41 sha256_bs1.txt
-rw-rw-r-- 1 kali kali 0 Oct 23 10:39 strace_bs1.txt
-rw-rw-r-- 1 kali kali 1.2K Oct 23 10:41 strings_bs1.txt

(kali@kali)-[~/ghidra_cli_practice/output/report_evidence]
$ cat >> analysis_report.md <<'EOF'

## Appendix (evidence files attached)
- report.txt
- strings_bs1.txt
- sha256 bs1.txt
- strace bs1.txt
- ltrace bs1.txt
- objdump_main bs1.txt
- readelf header bs1.txt

EOF

(kali@kali)-[~/ghidra_cli_practice/output/report_evidence]
$ tail -n 40 analysis_report.md

## Files & hash
022ace48e298cb775fca66f933fe52adebf9078ca7064d95367732f6e21072df benign_sample1

## Strings (selected)
report.txt
Confidential report inside docs folder.
Trying to connect to: 192.0.2.123:4444
```

```
(kali@kali)-[~/ghidra_cli_practice/output/report_evidence]
$ tail -n 40 analysis_report.md

## Files & hash
022ace48e298cb775fca66f933fe52adebf9078ca7064d95367732f6e21072df benign_sample1

## Strings (selected)
report.txt
Confidential report inside docs folder.
Trying to connect to: 192.0.2.123:4444

## Strace (important lines)
49048 openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
49048 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
49048 openat(AT_FDCWD, "report.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3
49048 write(3, "Confidential report inside docs " ..., 40) = 40
49048 write(1, "Trying to connect to: 192.0.2.12" ..., 39) = 39

## Ltrace (top)
fopen("report.txt", "w") = 0x55d39d2152a0
fwrite("Confidential report inside docs " ..., 1, 40, 0x55d39d2152a0) = 40
fclose(0x55d39d2152a0) = 0
puts("Trying to connect to: 192.0.2.12" ...) = 39
+++ exited (status 0) +++

## Program stdout
Trying to connect to: 192.0.2.123:4444

## Notes
- Binary copied from: ~/ghidra_cli_practice/samples/benign_sample1
- This run used a short timeout (3s) so it is safe and non-blocking.
EOF ## pandoc analysis_report.md -o analysis_report.pdf --pdf-engine=xelatex 2>/dev/null || echo 'PDF generation failed (xelatex/pandoc may be missing). All evidence files are in ~/ghidra_cli_practice/output/report_evidence/'

## Appendix (evidence files attached)
- report.txt
- strings_bs1.txt
- sha256_bs1.txt
- strace_bs1.txt
- ltrace_bs1.txt
- objdump_main_bs1.txt
- readelf_header_bs1.txt
```

## Conclusion:

The benign sample binaries were successfully analyzed using Linux commands (file, strings, strace, ltrace, ps, lsof), revealing their runtime behavior and file operations.

This demonstrates how basic command-line tools can safely inspect and understand program behavior without using a full GUI reverse-engineering tool like Ghidra.