**Reverse engineering tools**

* Objdump and Hexdump
* *Objdump* programs can be used to present information from the object files.
* objdump -M intel -d sampleak > disassembly.asm
* The *hd* or *hexdump* command in Linux is used to filter and display the specified files, or standard input in a human readable specified format. Eg: hd -c input.txt
* **Readelf**
  + *readelf* displays information about one or more ELF format object files. The options control what particular information to display.
* Strings
* We can search for all printable strings in the file with the *strings* command.
* **Dynamic analysis**

There are a few tools in Linux that can be used to display more detailed information: *ltrace*, *strace*, and *gdb* can be used for this reversing activity.

* ltrace:

The output of ltrace shows a readable code of what the program did. ltrace logged library functions that the program called and received. Can be executed: ltrace ./sampleak

* strace:

strace is another tool we can use, which logs system calls. strace logs every system call that happened, starting from when it was being executed by the system. Can be executed: strace ./sampleak

* Gdb

.gdbinit is the user initialization file contains commands that are executed upon the start of GDB. It is located in the home directory under the path:

~/.gdbinit

./gdbinit can contain the following lines:

set disassembly-flavor intel

tui enable

layout asm

layout regs

start

break main

Show stored values on the stack

(gdb) print $esp

Display executable sections:

(gdb) main info sec

Breakpoints and control flow

|  |  |
| --- | --- |
| start | Take first step into the program |
| step / s | Step into the current instruction |
| next / ni | Step over the current instruction |
| finish | Step out of the current function |
| b [line] | Set breakpoint at given line in current file |
| b [file]:[line] | Set breakpoint at given line in given file |
| b [func] | Set breakpoint at start of given function |
| delete | Clear all breakpoints |
| run / r | Run the program if not already running |
| continue / c | Continue until breakpoint is hit |
| kill / k | Kill program but don't quit GDB |
| quit / q | Kill the program if running and quit GDB |
| (Enter key) | Repeat the last command |
| i r | Show the register information |

### Printing variables

| ptype [var] | Print type info of the given variable |
| --- | --- |
| print [var] | Print value of the given variable |
| print \*[var] | Print the de referenced value of the variable |
| info args | Print function arguments |
| info locals | Print local variables |

### Stack

| bt | Print full call stack |
| --- | --- |
| up | Move up the call stack |
| down | Move down the call stack |
| frame [n] | Move to the given stack frame |

* **Ghidra**

We can make use of Ghidra’s decomplier. When we select a portion of assembly code, its decompiled version is shown in the decompiler window.

**Network analysis**

* **Linux commands used:**
* To list all TCP or UDP ports that are being listened on, including the services using the ports and the socket status use the following command: sudo netstat -tunlp
* To filter the results, we can use the grep command . For example, to find what process listens on TCP port no, type:
* sudo netstat -tnlp | grep :<port n>
* To get a list of all listening ports with ss you would type: sudo ss -tunlp
* **TCPDUMP commands for network filtering:**

List the interfaces:

tcpdump -list -interface

tcpdump -D

Capture traffic of an interface:

tcpdump -i eth0

### Port filter: sudo tcpdump -n port <portno>

### Host filter: sudo tcpdump -n host <ip>

### Packet filter: tcpdump -i eth0 icmp -c 10

Filteringby source/destination:

sudo tcpdump -n src host <ip>

sudo tcpdump -n dst port <portno>

Packet directions:

tcpdump -i eth0 icmp -c 5 -Q in

tcpdump -i eth0 icmp -c 5 -Q out

### Capturing traffic of a particular interface: tcpdump -i eth0

### Verbose mode: tcpdump -i eth0 -c 5 -v

### Printing each packet in ASCII: tcpdump -i eth0 -c 5 -A

Reference:[*https://opensource.com/article/18/10/introduction-tcpdump*](https://opensource.com/article/18/10/introduction-tcpdump)

[*https://www.geeksforgeeks.org/netstat-command-linux/*](https://www.geeksforgeeks.org/netstat-command-linux/)

[*https://www.cprogramming.com/gdb.html*](https://www.cprogramming.com/gdb.html)