

Reverse Engineering Assignment

Clayton B. Hodges

Colorado Mesa University

CSCI 420: Software Security

Contents

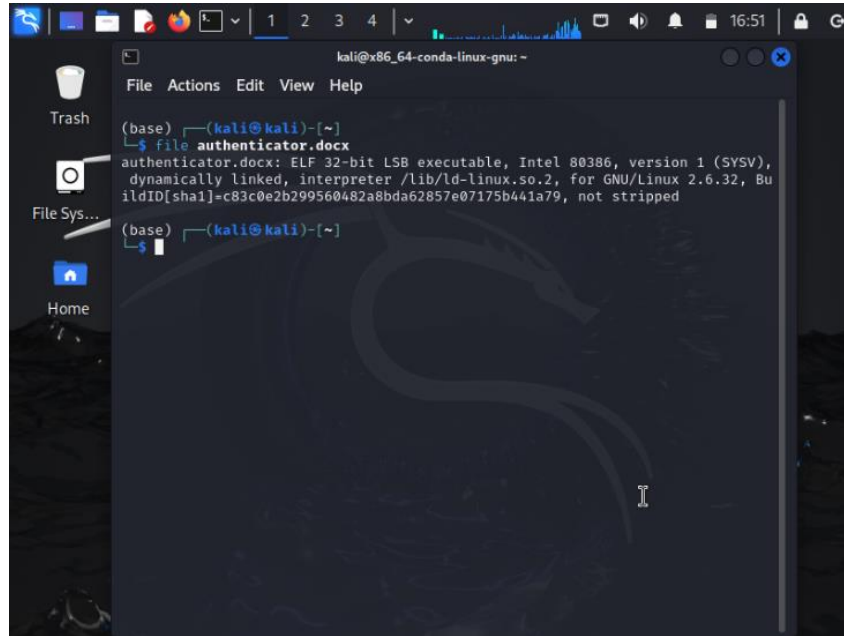
1.	Task Completion.....	2
1.1	File Fomat	2
1.2	Password Reverse Engineering.....	2
1.3	Modified Binary	4
1.4	Checksums & md5 & sha1	5

1. Task Completion

1.1 File Fomat

➤ file authenticator.docx

- This command displays file information of the authenticator.docx file, which is an executable (ELF).



```
kali@x86_64-conda-linux-gnu: ~  
File Actions Edit View Help  
(base) (kali@kali)-[~]  
└─$ file authenticator.docx  
authenticator.docx: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),  
dynamically linked, interpreter /lib/ld-linux.so.2, for GNU/Linux 2.6.32, Bu  
ildID[sha1]=c83c0e2b299560482a8bda62857e07175b441a79, not stripped  
(base) (kali@kali)-[~]  
└─$
```

1.2 Password Reverse Engineering

➤ cp authenticator.docx authenticator2.docx

- This command copied the contents of the original file (authenticator.docx) to a new file (authenticator2.txt) for modification purposes.

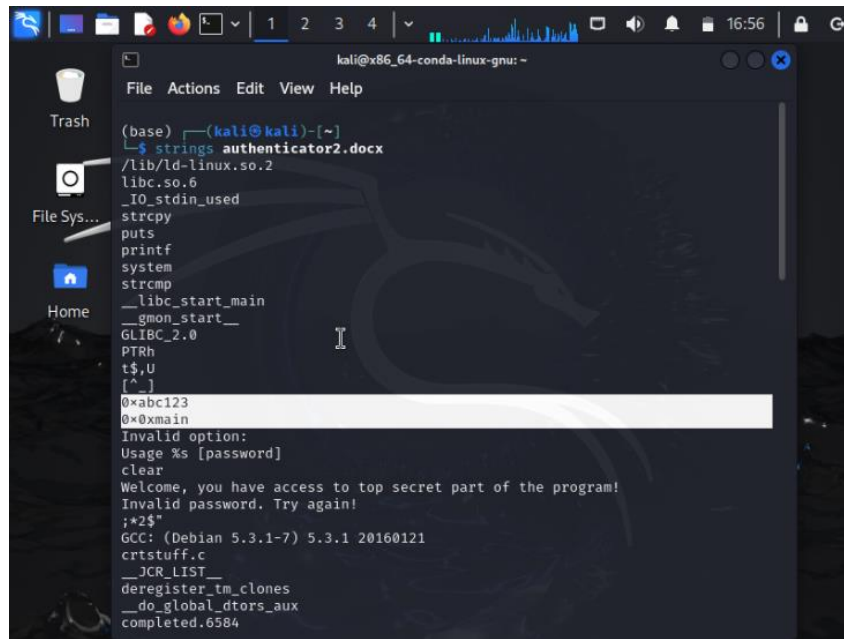
➤ chmod +x authenticator2.docx

- This command added executable permission to the authenticator2.docx file.

➤ strings authenticator2.docx

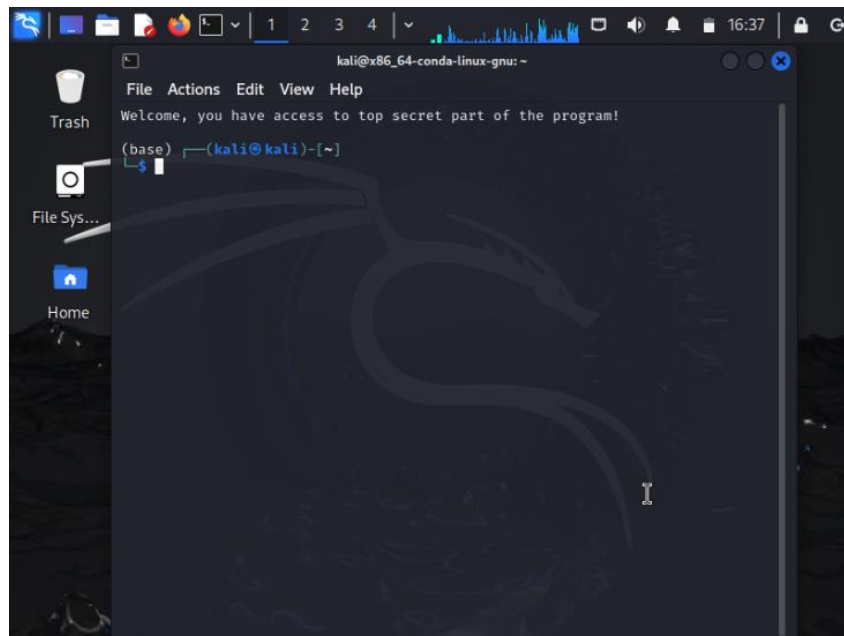
- This command listed the strings in the authenticator2.docx file. At the top, two strings appeared like passwords, 0xabc123 and 0x0xmain (see image below).

A3 – REVERSE ENGINEERING



A terminal window on a Kali Linux system. The command `strings authenticator2.docx` has been executed. The output lists various strings found in the file, including system paths like `/lib/ld-linux.so.2` and `libc.so.6`, standard library functions like `_IO_stdin_used`, `strcpy`, `puts`, `printf`, `system`, `strcmp`, `__libc_start_main`, `__gmon_start__`, `GLIBC_2.0`, `PTRh`, `t$,U`, `[^_]`, `0xabc123`, and `0x0xmain`. It also shows an 'Invalid option' error, usage instructions, a 'clear' command, a welcome message, an invalid password error, a shell prompt `;$*`, and compiler information: `GCC: (Debian 5.3.1-7) 5.3.1 20160121`, `crtstuff.c`, `__JCR_LIST__`, `deregister_tm_clones`, `__do_global_ctors_aux`, and `completed.6584`.

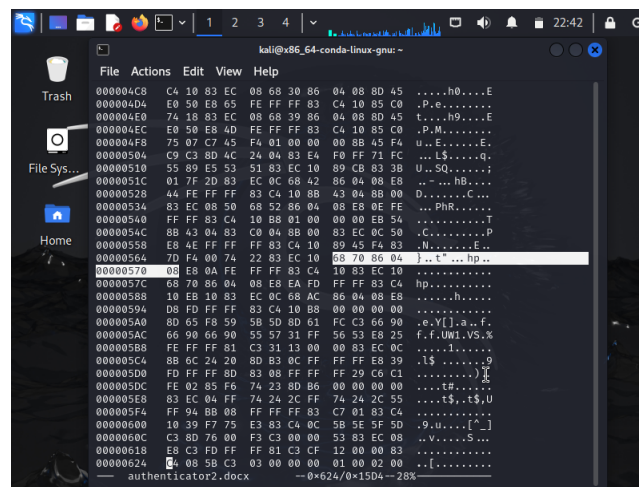
- `./authenticator2.docx 0xabc123` or `./authenticator2.docx 0x0xmain`
 - This command executed `authenticator2.docx` with its appropriate password, clearing the screen and outputting 'Welcome, you have access to top secret part of the program!' (see image below).



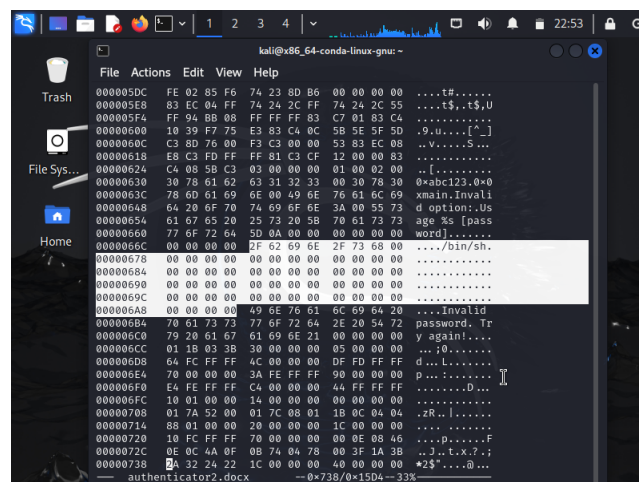
A terminal window on a Kali Linux system. The command `./authenticator2.docx 0xabc123` has been executed. The output is `Welcome, you have access to top secret part of the program!`. The terminal prompt is `(base) (kali@kali)~`.

1.3 Modified Binary

- `cp authenticator.docx authenticator2.docx`
 - This command made a copy (2) of the `authenticator.docx` file for me to experiment with.
- `hexedit authenticator2.docx`
 - This command opened a hex editor for the `authenticator2.docx` file, allowing me to push the memory address of the 'top secret program' string instead of the memory address of the 'clear' string (see below image).

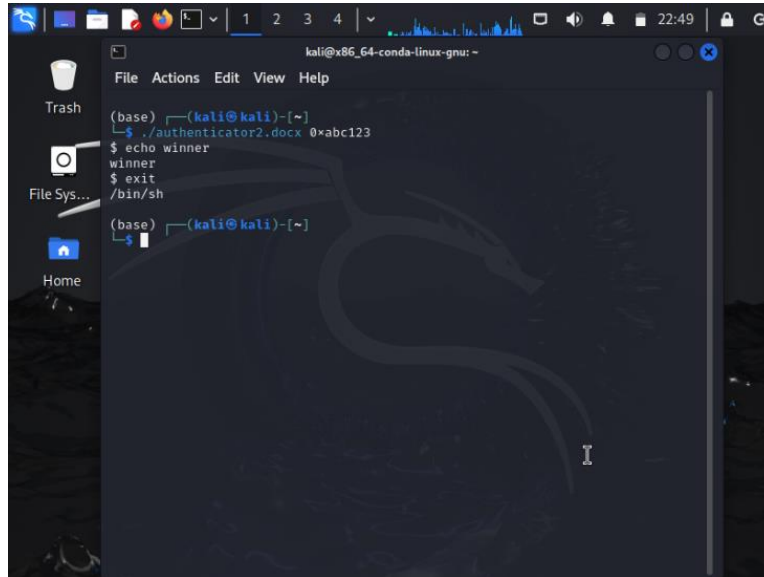


- I then proceeded to replace the contents of the 'top secret program' string with `"/bin/sh"` so that 'system' would be called with this value (see below).



A3 – REVERSE ENGINEERING

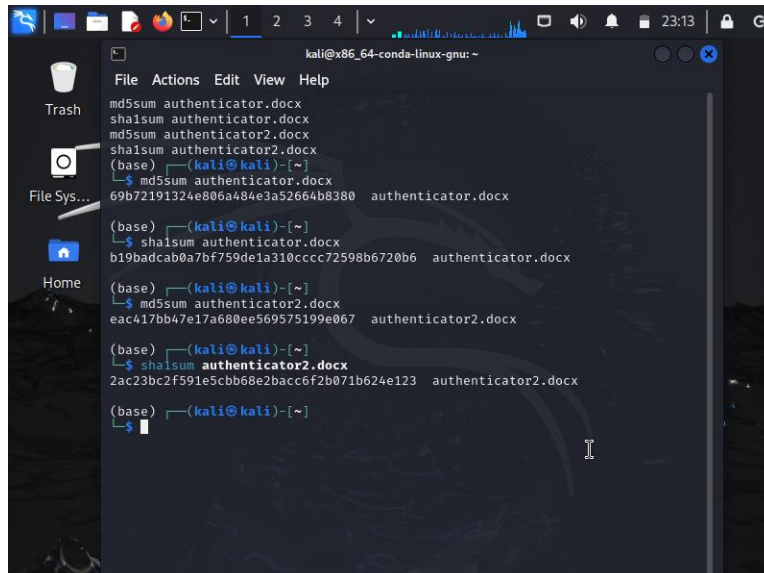
- `./authenticator2.docx 0xabc123`
 - This command executed `authenticator2.docx` with a password, but now the output executes `/bin/sh`.



```
kali@x86_64-conda-linux-gnu: ~  
File Actions Edit View Help  
(base) (kali@kali)-[~]  
└─$ ./authenticator2.docx 0xabc123  
$ echo winner  
winner  
$ exit  
/bin/sh  
(base) (kali@kali)-[~]  
└─$
```

1.4 Checksums & md5 & sha1

- Checksums are used to check the integrity of files; they are hashes that create unique values for each set of data. If the data changes, the hash will change too.



```
kali@x86_64-conda-linux-gnu: ~  
File Actions Edit View Help  
md5sum authenticator.docx  
sha1sum authenticator.docx  
md5sum authenticator2.docx  
sha1sum authenticator2.docx  
(base) (kali@kali)-[~]  
└─$ md5sum authenticator.docx  
69b72191324e806a484e3a52664b8380 authenticator.docx  
(base) (kali@kali)-[~]  
└─$ sha1sum authenticator.docx  
b19badcab0a7bf759de1a310cccc72598b6720b6 authenticator.docx  
(base) (kali@kali)-[~]  
└─$ md5sum authenticator2.docx  
eac417bb47e17a680ee569575199e067 authenticator2.docx  
(base) (kali@kali)-[~]  
└─$ sha1sum authenticator2.docx  
2ac23bc2f591e5cbb68e2bacc6f2b071b624e123 authenticator2.docx  
(base) (kali@kali)-[~]  
└─$
```