LITTLE TRICK

AUTHOR:

CYBEXER TECHNOLOGIES

International Cyber Security Challenge

SEPT 2021

1. DESCRIPTION

There is a very simple program written in C that hides our flag under a password. The program prints out the flag when a correct password is entered. The binary is available for download.

2. CHALLENGE SPECIFICATIONS

- Categroty: Reverse Engineering
- Difficulty: Easy
- Estimated time: 5-10 min

3. QUESTIONS AND ANSWERS

3.1 WHAT FLAG IS PRINTED BY THE PROGRAM?

abcdefgh

4. SETUP INSTRUCTIONS

Dockerfile and *docker-compose.yml* are provided to run the task in a container. **FLAG** and **PORT** are passed from docker-compose environment, see **.env**. FLAG is set at build time; PORT can be changed without rebuild. **FLAG must be exactly 8 bytes!** Password for unlocking the flag is generated randomly at each build, but it always starts with a newline.

docker-compose build

docker-compose up

The executable is generated during build time with commands in *Dockerfile*. Container is just serving it over HTTP.

5. ARTIFACTS PROVIDED

File	SHA-256
littletrick.tar.gz	6e47bd6e0c35abaead1520e1b328e1976bd6facf91688e0313fc560e33dcf826

6. TOOLS NEEDED

- A hex editor, debugger, disassembler, e.g., gdb, IDA, Ghidra, etc.

7. WALKTHROUGH

Start by identifying what file is provided to you:

```
$ file littletrick
littletrick: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked,
interpreter /lib/ld-musl-x86_64.so.1, with debug_info, not stripped
```

A Linux binary, let's try to run it:

```
$ chmod +x littletrick
$ ./littletrick
-bash: ./littletrick: No such file or directory
```

Such message is indication of missing libraries or inappropriate binary format. We could have noticed it already from output of "file".

```
$ ldd littletrick
        linux-vdso.so.1 (0x00007ffe74df5000)
        libc.musl-x86_64.so.1 => not found
```

MUSL is an implementation of standard C library that is used in Linux distributions where small footprint is important, e.g., BusyBox or Alpine.

Fortunately, there is a package available in Ubuntu and Kali:

```
$ sudo apt install musl
```

Retry:

```
$ ./littletrick
Password? asd
Wrong password, no flag. Sorry
```

Let's inspect the binary with *strings*:

```
$ strings littletrick  | grep -A 3 password
Wrong password, no flag. Sorry
3877afdfade1d
strings is not enough to solve the challenge...
please run from terminal
```

The error message is visible, something that might be a password is on the next line followed by a hint that strings is not enough. Let's look at hex dump of the same place:

```
00002000: 5772 6f6e 6720 7061 7373 776f 7264 2c20  Wrong password,
00002010: 6e6f 2066 6c61 672e 2053 6f72 7279 000d  no flag. Sorry..
00002020: 3338 3737 6166 6466 6164 6531 640d 0000  3877afdfade1d...
00002030: 7374 7269 6e67 7320 6973 206e 6f74 2065  strings is not e
00002040: 6e6f 7567 6820 746f 2073 6f6c 7665 2074  nough to solve t
00002050: 6865 2063 6861 6c6c 656e 6765 2e2e 2e00  he challenge....
```

Text strings in C program are null terminated. In the end of second line of above output we can see "Sorry\0" (highlighted with yellow) that terminates the error message. Next byte is 0x0d (carriage return), followed by what we assumed to be the password. End of third line is again null bytes, terminating the potential password. This part is highlighted with green. Then the hint follows. Thus, password can be "**\r3877afdfade1d\r**", i.e., it starts with a newline.

Let's try:

```
$ ./littletrick
Password?
3877afdfade1d
abcdefgh
```

Done. It was a little trick indeed.

ENISA
European Union Agency for Cybersecurity

Athens Office
1 Vasilissis Sofias Str.
151 24 Marousi, Attiki, Greece

Heraklion Office
95 Nikolaou Plastira
700 13 Vassilika Vouton, Heraklion, Greece

enisa.europa.eu

ISBN xxx-xx-xxxx-xxx-x
doi:xx.xxxx/xxxxxx
TP-xx-xx-xxx-EN-C