Radare2

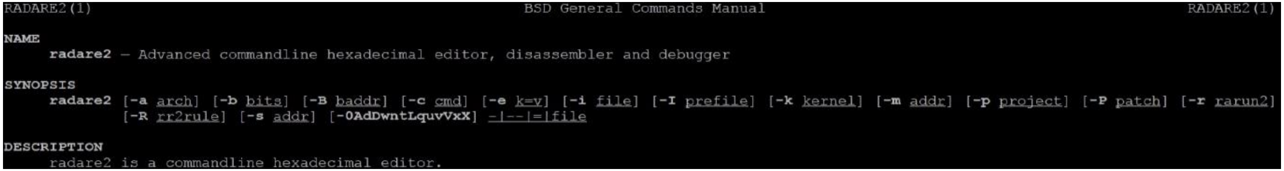
Information Security Inc.

Contents

- What is Radare2?
- Dependencies
- Testing Setup
- Installing Radare2
- Using Radare2
- References

What is Radare2?

- Advanced commandline hexadecimal editor, disassembler and debugger



```
RADARE2(1)                       BSD General Commands Manual                       RADARE2(1)

NAME
     radare2 — Advanced commandline hexadecimal editor, disassembler and debugger

SYNOPSIS
     radare2 [-a arch] [-b bits] [-B baddr] [-c cmd] [-e k=v] [-i file] [-I prefile] [-k kernel] [-m addr] [-p project] [-P patch] [-r rarun2]
             [-R rr2rule] [-s addr] [-0AdDwntLquvVxX] -|--|=|file

DESCRIPTION
     radare2 is a commandline hexadecimal editor.
```

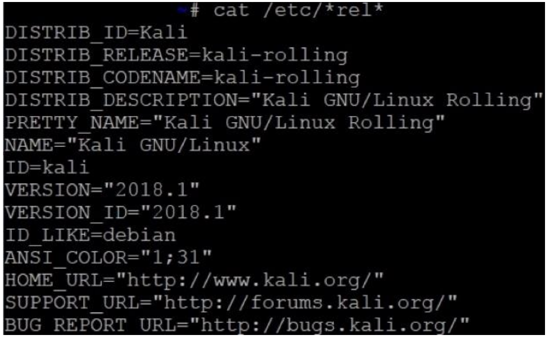
Dependencies

- radare2 can be built without any special dependency, just use make and get a working toolchain (gcc, clang, tcc, ..)
- Optionally you can use libewf for loading EnCase disk image.
- To build the bindings you need latest valabind, g++ and swig2

```
 ____  ___  ___  ___ ____  ___    ____
|  _ \/   \|   \/   \  _ \/ _ \  (__  \
|    (  -  | |  ) -  |   (   _/  /  __/
|__\__|_|__|___/__|__|_\__|___|  |____|

            www.radare.org
```

Testing Setup

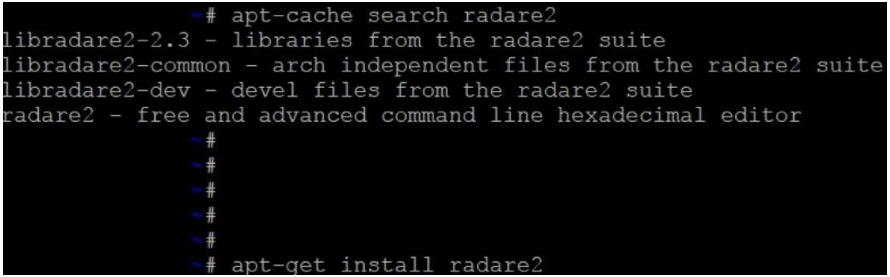
- Kali Linux 2018.1



```
~# cat /etc/*rel*
DISTRIB_ID=Kali
DISTRIB_RELEASE=kali-rolling
DISTRIB_CODENAME=kali-rolling
DISTRIB_DESCRIPTION="Kali GNU/Linux Rolling"
PRETTY_NAME="Kali GNU/Linux Rolling"
NAME="Kali GNU/Linux"
ID=kali
VERSION="2018.1"
VERSION_ID="2018.1"
ID_LIKE=debian
ANSI_COLOR="1;31"
HOME_URL="http://www.kali.org/"
SUPPORT_URL="http://forums.kali.org/"
BUG_REPORT_URL="http://bugs.kali.org/"
```

Installing Radare2

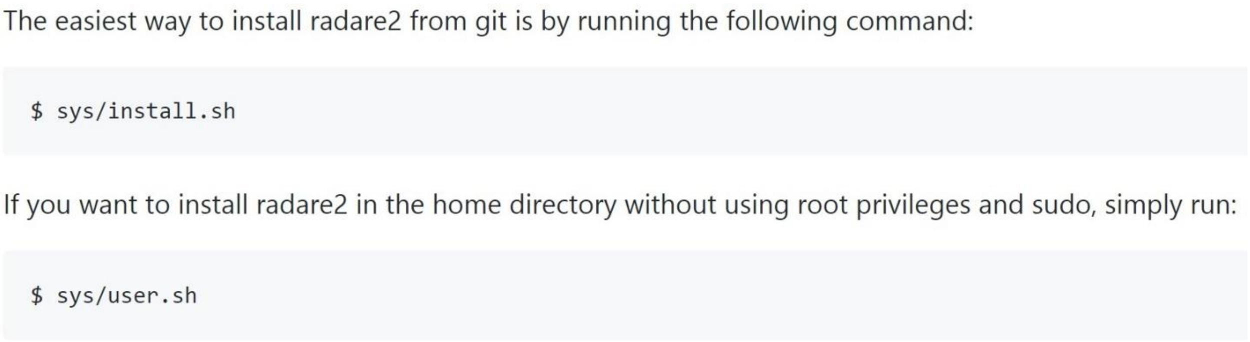
- Using apt



```
              ~# apt-cache search radare2
libradare2-2.3 - libraries from the radare2 suite
libradare2-common - arch independent files from the radare2 suite
libradare2-dev - devel files from the radare2 suite
radare2 - free and advanced command line hexadecimal editor
              ~#
              ~#
              ~#
              ~#
              ~#
              ~# apt-get install radare2
```

Installing Radare2

- From GitHub



The easiest way to install radare2 from git is by running the following command:

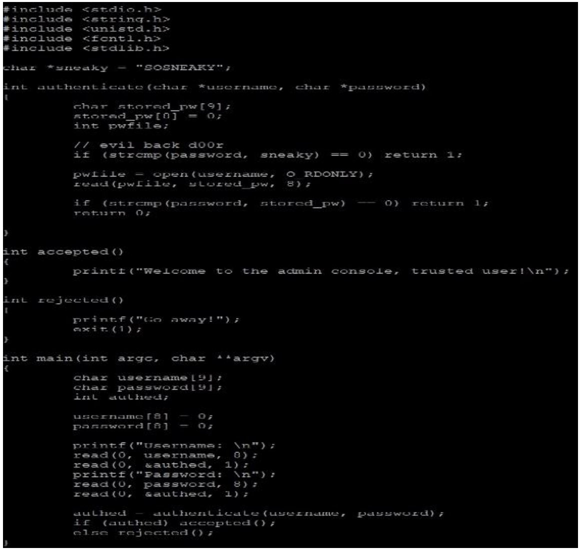
```
$ sys/install.sh
```

If you want to install radare2 in the home directory without using root privileges and sudo, simply run:

```
$ sys/user.sh
```

Using Radare2

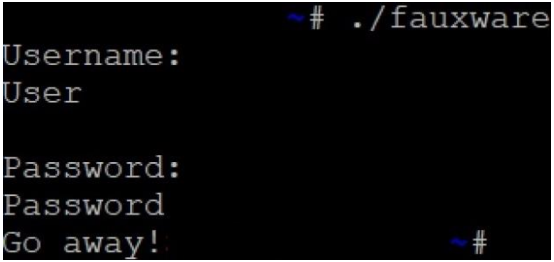
- Solving fauxware (https://github.com/angr/angr-doc/blob/master/examples/fauxware/fauxware.c)



```c
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdlib.h>

char *sneaky = "SOSNEAKY";

int authenticate(char *username, char *password)
{
        char stored_pw[9];
        stored_pw[8] = 0;
        int pwfile;

        // evil back d00r
        if (strcmp(password, sneaky) == 0) return 1;

        pwfile = open(username, O_RDONLY);
        read(pwfile, stored_pw, 8);

        if (strcmp(password, stored_pw) == 0) return 1;
        return 0;

}

int accepted()
{
        printf("Welcome to the admin console, trusted user!\n");
}

int rejected()
{
        printf("Go away!");
        exit(1);
}

int main(int argc, char **argv)
{
        char username[9];
        char password[9];
        int authed;

        username[8] = 0;
        password[8] = 0;

        printf("Username: \n");
        read(0, username, 8);
        read(0, &authed, 1);
        printf("Password: \n");
        read(0, password, 8);
        read(0, &authed, 1);

        authed = authenticate(username, password);
        if (authed) accepted();
        else rejected();
}
```

Using Radare2

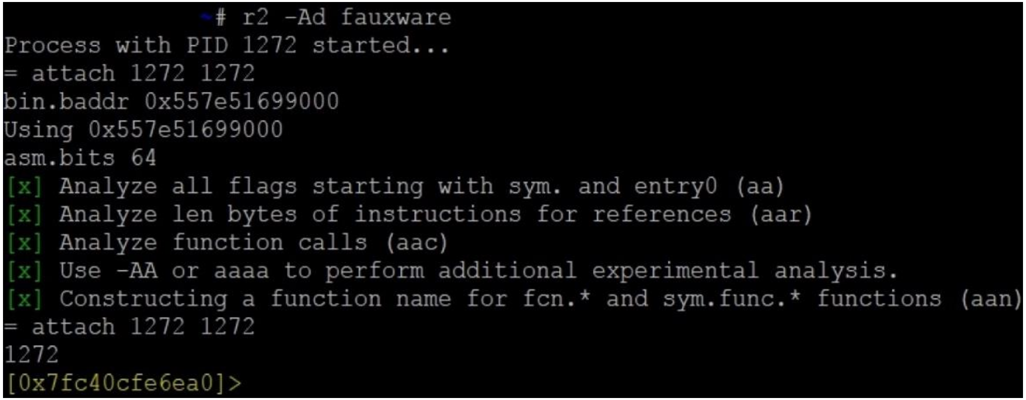
- Run the program >>> Password Challenge! Apparently, its just a simple program that tests a password entered by the user



```
                  ~# ./fauxware
Username:
User

Password:
Password
Go away!                    ~#
```

Using Radare2

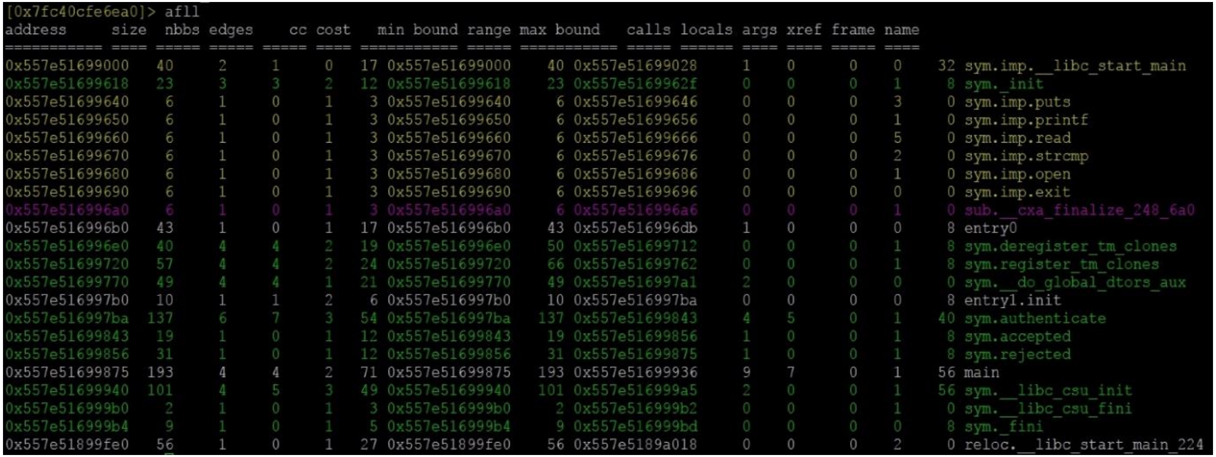
- Starting Radare2 with analyze and debug options



```
              ~# r2 -Ad fauxware
Process with PID 1272 started...
= attach 1272 1272
bin.baddr 0x557e51699000
Using 0x557e51699000
asm.bits 64
[x] Analyze all flags starting with sym. and entry0 (aa)
[x] Analyze len bytes of instructions for references (aar)
[x] Analyze function calls (aac)
[x] Use -AA or aaaa to perform additional experimental analysis.
[x] Constructing a function name for fcn.* and sym.func.* functions (aan)
= attach 1272 1272
1272
[0x7fc40cfe6ea0]>
```

Using Radare2

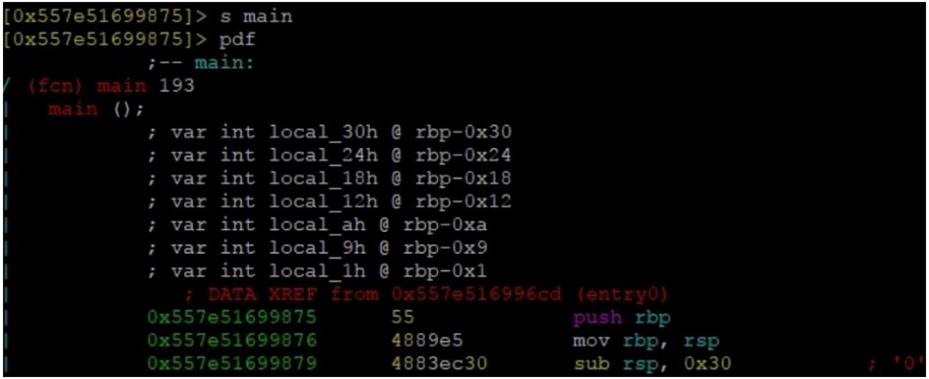
- Let’s look at the functions present in the binary, main is at address 0x557e51699875



```
[0x7fc40cfe6ea0]> afll
address      size  nbbs edges    cc cost   min bound range max bound   calls locals args xref frame name
=========== ==== ===== ===== ===== ==== =========== ===== =========== ===== ====== ==== ==== ===== ====
0x557e51699000   40     2     1     0   17 0x557e51699000    40 0x557e51699028     1    0      0    0   32 sym.imp.__libc_start_main
0x557e51699618   23     3     3     2   12 0x557e51699618    23 0x557e5169962f     0    0      0    1    8 sym._init
0x557e51699640    6     1     0     1    3 0x557e51699640     6 0x557e51699646     0    0      0    3    0 sym.imp.puts
0x557e51699650    6     1     0     1    3 0x557e51699650     6 0x557e51699656     0    0      0    1    0 sym.imp.printf
0x557e51699660    6     1     0     1    3 0x557e51699660     6 0x557e51699666     0    0      0    5    0 sym.imp.read
0x557e51699670    6     1     0     1    3 0x557e51699670     6 0x557e51699676     0    0      0    2    0 sym.imp.strcmp
0x557e51699680    6     1     0     1    3 0x557e51699680     6 0x557e51699686     0    0      0    1    0 sym.imp.open
0x557e51699690    6     1     0     1    3 0x557e51699690     6 0x557e51699696     0    0      0    0    0 sym.imp.exit
0x557e516996a0    6     1     0     1    3 0x557e516996a0     6 0x557e516996a6     0    0      0    1    0 sub.__cxa_finalize_248_6a0
0x557e516996b0   43     1     0     1   17 0x557e516996b0    43 0x557e516996db     1    0      0    0    8 entry0
0x557e516996e0   40     4     4     2   19 0x557e516996e0    50 0x557e51699712     0    0      0    1    8 sym.deregister_tm_clones
0x557e51699720   57     4     4     2   24 0x557e51699720    66 0x557e51699762     0    0      0    1    8 sym.register_tm_clones
0x557e51699770   49     4     4     1   21 0x557e51699770    49 0x557e516997a1     2    0      0    0    0 sym.__do_global_dtors_aux
0x557e516997b0   10     1     1     2    6 0x557e516997b0    10 0x557e516997ba     0    0      0    0    8 entry1.init
0x557e516997ba  137     6     7     3   54 0x557e516997ba   137 0x557e51699843     4    5      0    1   40 sym.authenticate
0x557e51699843   19     1     0     1   12 0x557e51699843    19 0x557e51699856     1    0      0    1    8 sym.accepted
0x557e51699856   31     1     0     1   12 0x557e51699856    31 0x557e51699875     1    0      0    1    8 sym.rejected
0x557e51699875  193     4     4     2   71 0x557e51699875   193 0x557e51699936     9    7      0    1   56 main
0x557e51699940  101     4     5     3   49 0x557e51699940   101 0x557e516999a5     2    0      0    1   56 sym.__libc_csu_init
0x557e516999b0    2     1     0     1    3 0x557e516999b0     2 0x557e516999b2     0    0      0    1    0 sym.__libc_csu_fini
0x557e516999b4    9     1     0     1    5 0x557e516999b4     9 0x557e516999bd     0    0      0    0    8 sym._fini
0x557e51899fe0   56     1     0     1   27 0x557e51899fe0    56 0x557e5189a018     0    0      0    2    0 reloc.__libc_start_main_224
```

Using Radare2

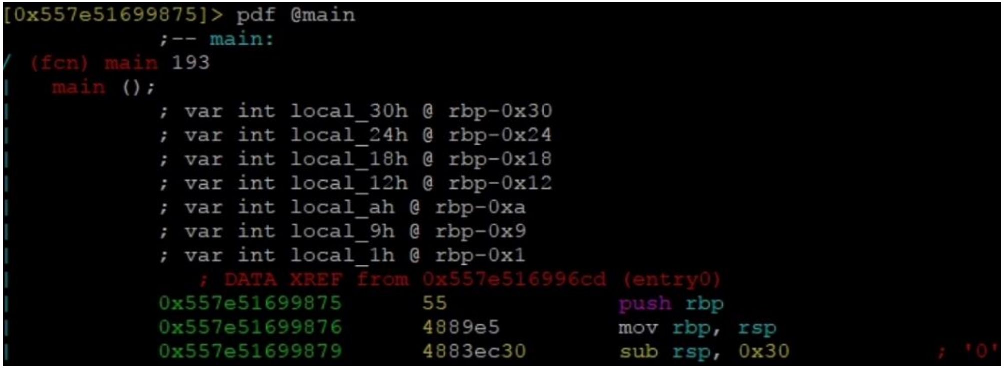
- The code in the main function, “s main” (seek main)



```
[0x557e51699875]> s main
[0x557e51699875]> pdf
            ;-- main:
/ (fcn) main 193
|   main ();
|           ; var int local_30h @ rbp-0x30
|           ; var int local_24h @ rbp-0x24
|           ; var int local_18h @ rbp-0x18
|           ; var int local_12h @ rbp-0x12
|           ; var int local_ah @ rbp-0xa
|           ; var int local_9h @ rbp-0x9
|           ; var int local_1h @ rbp-0x1
|              ; DATA XREF from 0x557e516996cd (entry0)
|           0x557e51699875      55             push rbp
|           0x557e51699876      4889e5         mov rbp, rsp
|           0x557e51699879      4883ec30       sub rsp, 0x30           ; '0'
```

Using Radare2

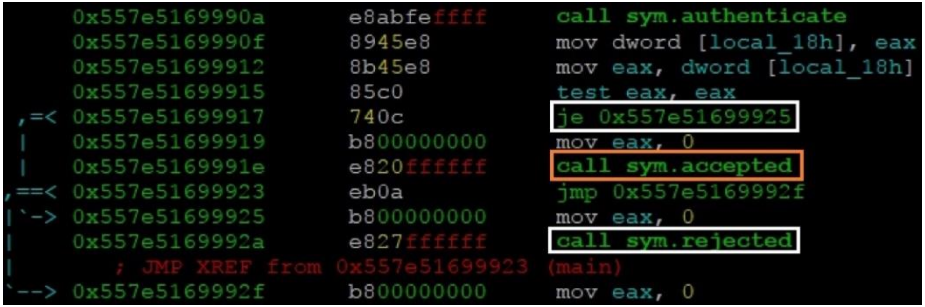
- The code in the main function, “pdf @main” (print disassemble function)



```
[0x557e51699875]> pdf @main
            ;-- main:
/ (fcn) main 193
|   main ();
|           ; var int local_30h @ rbp-0x30
|           ; var int local_24h @ rbp-0x24
|           ; var int local_18h @ rbp-0x18
|           ; var int local_12h @ rbp-0x12
|           ; var int local_ah @ rbp-0xa
|           ; var int local_9h @ rbp-0x9
|           ; var int local_1h @ rbp-0x1
|              ; DATA XREF from 0x557e516996cd (entry0)
|           0x557e51699875      55             push rbp
|           0x557e51699876      4889e5         mov rbp, rsp
|           0x557e51699879      4883ec30       sub rsp, 0x30          ; '0'
```

Using Radare2

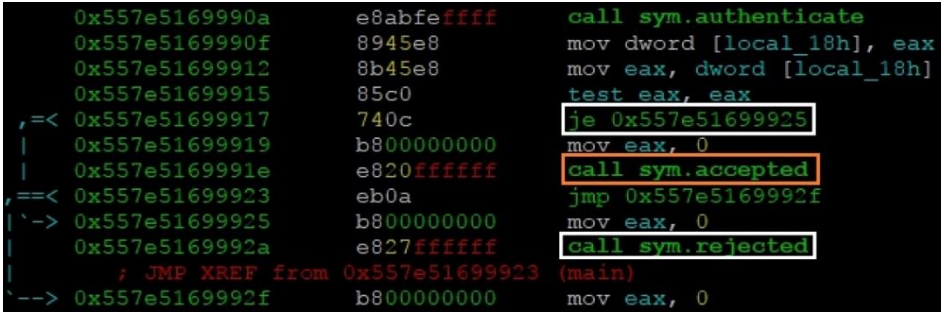
- We can see the code jumps to 0x557e51699925 (sym.rejected) if eax is zero (test eax,eax)



```
      0x557e5169990a      e8abfeffff      call sym.authenticate
      0x557e5169990f      8945e8          mov dword [local_18h], eax
      0x557e51699912      8b45e8          mov eax, dword [local_18h]
      0x557e51699915      85c0            test eax, eax
 ,=< 0x557e51699917      740c            je 0x557e51699925
 |    0x557e51699919      b800000000      mov eax, 0
 |    0x557e5169991e      e820ffffff      call sym.accepted
,==< 0x557e51699923      eb0a            jmp 0x557e5169992f
|`-> 0x557e51699925      b800000000      mov eax, 0
|     0x557e5169992a      e827ffffff      call sym.rejected
|        ; JMP XREF from 0x557e51699923 (main)
`--> 0x557e5169992f      b800000000      mov eax, 0
```

Using Radare2

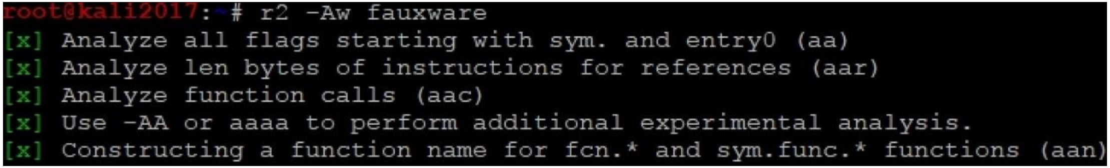
- Let’s modify the program and unconditionally jump to 0x557e5169991e (sym.accepted)



```
      0x557e5169990a    e8abfeffff    call sym.authenticate
      0x557e5169990f    8945e8        mov dword [local_18h], eax
      0x557e51699912    8b45e8        mov eax, dword [local_18h]
      0x557e51699915    85c0          test eax, eax
  ,=< 0x557e51699917    740c          je 0x557e51699925
  |   0x557e51699919    b800000000    mov eax, 0
  |   0x557e5169991e    e820ffffff    call sym.accepted
 ,==< 0x557e51699923    eb0a          jmp 0x557e5169992f
 |`-> 0x557e51699925    b800000000    mov eax, 0
 |    0x557e5169992a    e827ffffff    call sym.rejected
 |       ; JMP XREF from 0x557e51699923 (main)
 `--> 0x557e5169992f    b800000000    mov eax, 0
```

Using Radare2

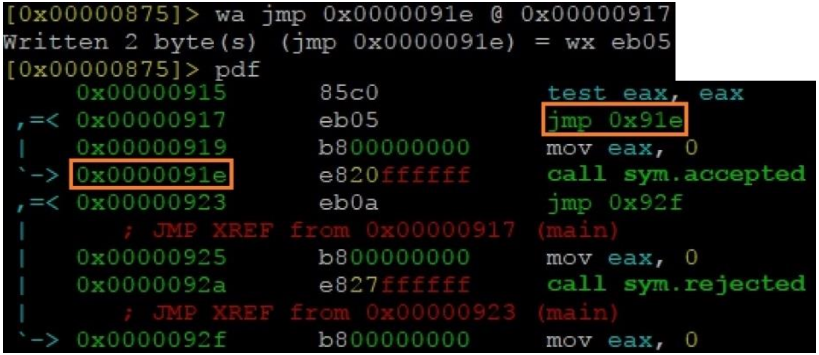
- Open the program in writing mode using the “w” keyword



```
root@kali2017:~# r2 -Aw fauxware
[x] Analyze all flags starting with sym. and entry0 (aa)
[x] Analyze len bytes of instructions for references (aar)
[x] Analyze function calls (aac)
[x] Use -AA or aaaa to perform additional experimental analysis.
[x] Constructing a function name for fcn.* and sym.func.* functions (aan)
```

Using Radare2

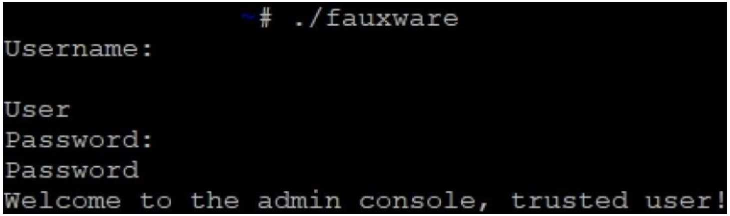
- Open the program in writing mode using the “w” keyword, use the command “wa jmp 0x0000091e @ 0x00000917



```
[0x00000875]> wa jmp 0x0000091e @ 0x00000917
Written 2 byte(s) (jmp 0x0000091e) = wx eb05
[0x00000875]> pdf
         0x00000915      85c0          test eax, eax
     ,=< 0x00000917      eb05          jmp 0x91e
     |   0x00000919      b800000000    mov eax, 0
     `-> 0x0000091e      e820ffffff    call sym.accepted
     ,=< 0x00000923      eb0a          jmp 0x92f
     |      ; JMP XREF from 0x00000917 (main)
     |   0x00000925      b800000000    mov eax, 0
     |   0x0000092a      e827ffffff    call sym.rejected
     |      ; JMP XREF from 0x00000923 (main)
     `-> 0x0000092f      b800000000    mov eax, 0
```

Using Radare2

- Run the program again and got “Welcome to the admin console, trusted user!” Done!



```
               ~# ./fauxware
Username:

User
Password:
Password
Welcome to the admin console, trusted user!
```

References

- GitHub
https://github.com/radare/radare2

- Official website
http://www.radare.org/r/

- Tutorials
https://moveax.me/radare-basics/
https://www.megabeets.net/a-journey-into-radare-2-part-1/
https://www.megabeets.net/a-journey-into-radare-2-part-2/