Lab 12: Fuzzing

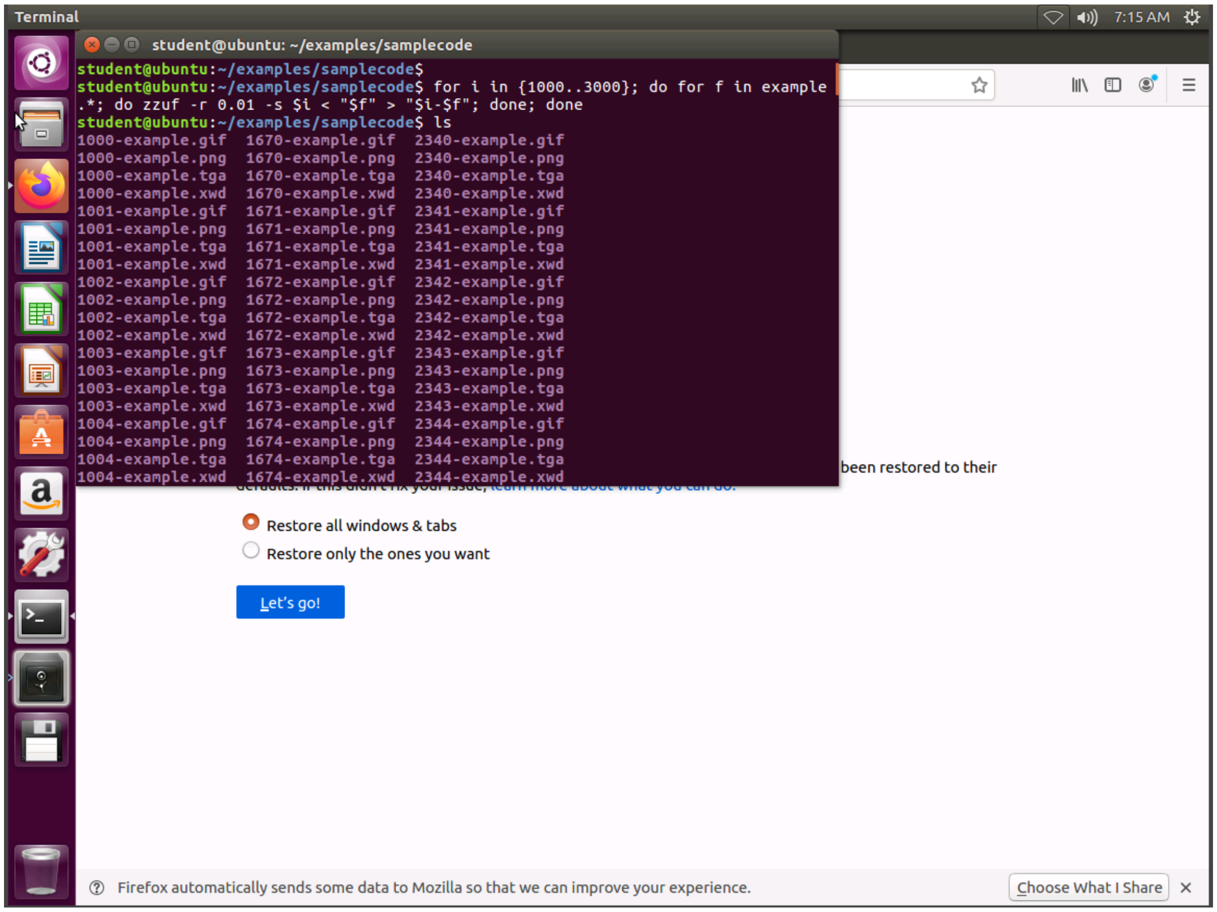
INFO40587: ETHICAL HACKING

Kevin Harianto| 991602128| August 6, 2024

Exercise 1: Fuzzing with zzuf

1.1 OUTPUT SCREENSHOTS

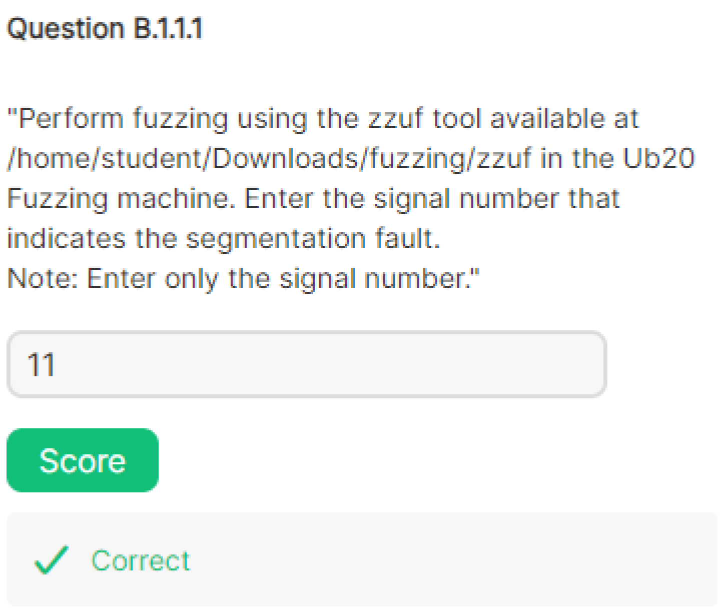
Exercise 1, Step 22: Thus, for every example file, we create 2000 malformed variants, all named in the form [number]-example.[extension]. The -r parameter for zzuf is the amount of change you want in a file. 0.01 means that 1% of the file gets changed randomly. The -s parameter is the seed. For every different s value, we get a different output. You can certainly adapt the number of variants but judging from experience, 2000 is a reasonable number to start with . Type **ls** and press **Enter**. An example of the output is shown in the following screenshot.



```
student@ubuntu:~/examples/samplecode$
student@ubuntu:~/examples/samplecode$ for i in {1000..3000}; do for f in example
.*; do zzuf -r 0.01 -s $i < "$f" > "$i-$f"; done; done
student@ubuntu:~/examples/samplecode$ ls
1000-example.gif  1670-example.gif  2340-example.gif
1000-example.png  1670-example.png  2340-example.png
1000-example.tga  1670-example.tga  2340-example.tga
1000-example.xwd  1670-example.xwd  2340-example.xwd
1001-example.gif  1671-example.gif  2341-example.gif
1001-example.png  1671-example.png  2341-example.png
1001-example.tga  1671-example.tga  2341-example.tga
1001-example.xwd  1671-example.xwd  2341-example.xwd
1002-example.gif  1672-example.gif  2342-example.gif
1002-example.png  1672-example.png  2342-example.png
1002-example.tga  1672-example.tga  2342-example.tga
1002-example.xwd  1672-example.xwd  2342-example.xwd
1003-example.gif  1673-example.gif  2343-example.gif
1003-example.png  1673-example.png  2343-example.png
1003-example.tga  1673-example.tga  2343-example.tga
1003-example.xwd  1673-example.xwd  2343-example.xwd
1004-example.gif  1674-example.gif  2344-example.gif
1004-example.png  1674-example.png  2344-example.png
1004-example.tga  1674-example.tga  2344-example.tga
1004-example.xwd  1674-example.xwd  2344-example.xwd
```

Terminal
7:15 AM
student@ubuntu: ~/examples/samplecode
been restored to their
Restore all windows & tabs
Restore only the ones you want
Let's go!
Firefox automatically sends some data to Mozilla so that we can improve your experience.
Choose What I Share

1.2 Questions


Question B.1.1.1

"Perform fuzzing using the zzuf tool available at /home/student/Downloads/fuzzing/zzuf in the Ub20 Fuzzing machine. Enter the signal number that indicates the segmentation fault.
Note: Enter only the signal number."

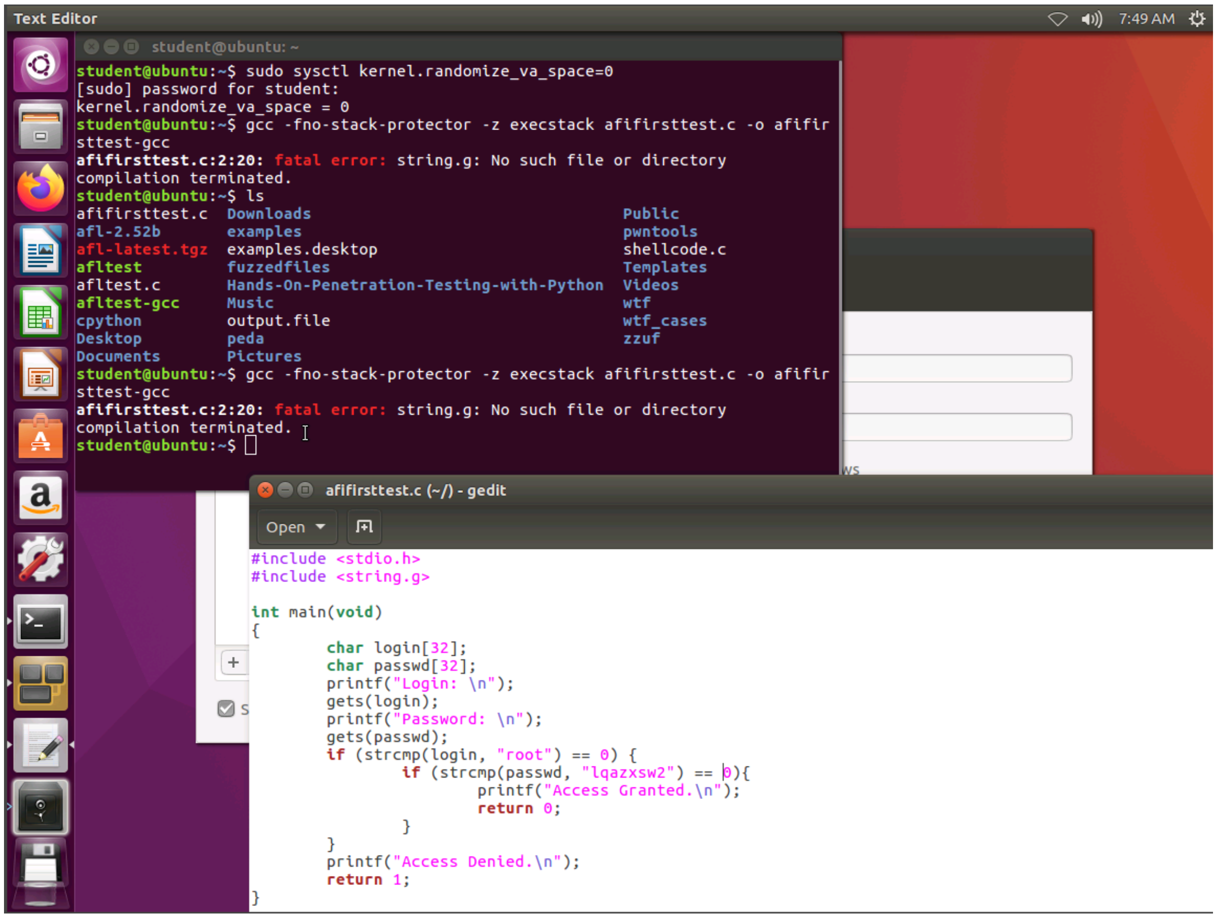
11

Exercise 3: Fuzzing with AFL

3.1 OUTPUT SCREENSHOTS

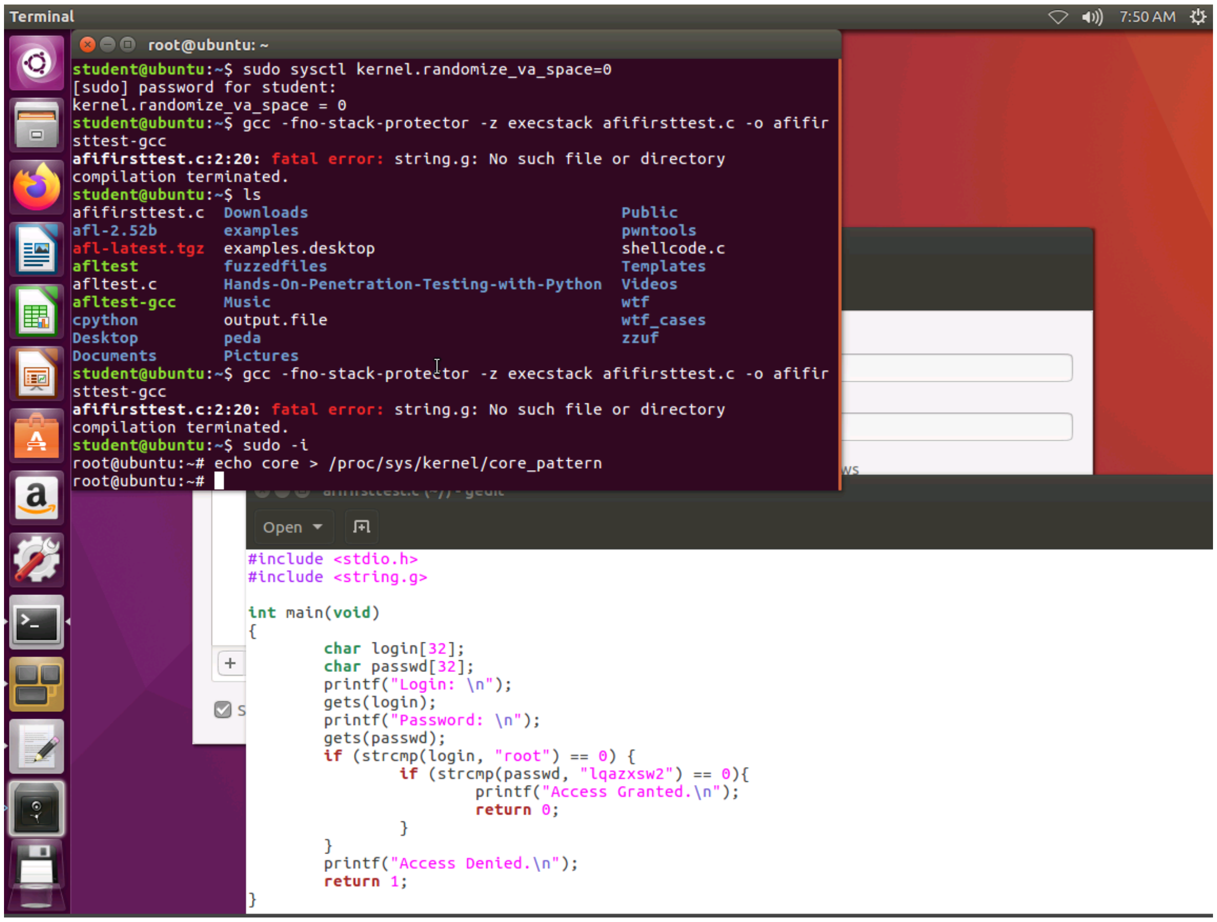
Exercise 3, Step 28: We should now be able to again run our command in the original terminal window. You might have to go to an expanded screen if you get an error message. An example of this is shown in the following screenshot.

NOTE: Unable to run the c program despite creating it accordingly


```
student@ubuntu:~$ sudo sysctl kernel.randomize_va_space=0
[sudo] password for student:
kernel.randomize_va_space = 0
student@ubuntu:~$ gcc -fno-stack-protector -z execstack afifirsttest.c -o afifir
sttest-gcc
afifirsttest.c:2:20: fatal error: string.g: No such file or directory
compilation terminated.
student@ubuntu:~$ ls
afifirsttest.c  Downloads                                 Public
afl-2.52b       examples                                  pwntools
afl-latest.tgz  examples.desktop                          shellcode.c
afltest         fuzzedfiles                               Templates
afltest.c       Hands-On-Penetration-Testing-with-Python  Videos
afltest-gcc     Music                                     wtf
cpython         output.file                               wtf_cases
Desktop         peda                                      zzuf
Documents       Pictures
student@ubuntu:~$ gcc -fno-stack-protector -z execstack afifirsttest.c -o afifir
sttest-gcc
afifirsttest.c:2:20: fatal error: string.g: No such file or directory
compilation terminated.
student@ubuntu:~$
```

```c
#include <stdio.h>
#include <string.g>

int main(void)
{
        char login[32];
        char passwd[32];
        printf("Login: \n");
        gets(login);
        printf("Password: \n");
        gets(passwd);
        if (strcmp(login, "root") == 0) {
                if (strcmp(passwd, "lqazxsw2") == 0){
                        printf("Access Granted.\n");
                        return 0;
                }
        }
        printf("Access Denied.\n");
        return 1;
}
```



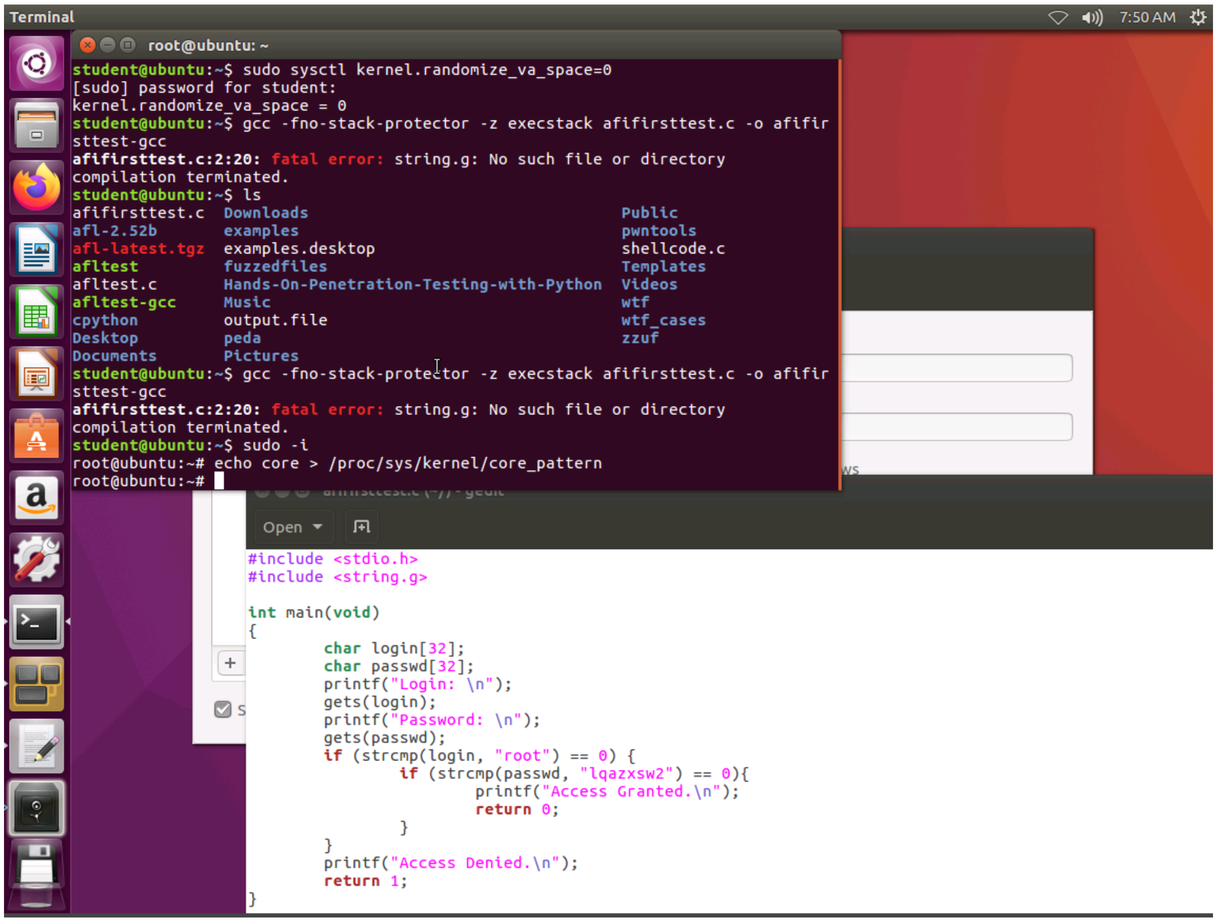
```
student@ubuntu:~$ sudo sysctl kernel.randomize_va_space=0
[sudo] password for student:
kernel.randomize_va_space = 0
student@ubuntu:~$ gcc -fno-stack-protector -z execstack afifirsttest.c -o afifir
sttest-gcc
afifirsttest.c:2:20: fatal error: string.g: No such file or directory
compilation terminated.
student@ubuntu:~$ ls
afifirsttest.c  Downloads                                  Public
afl-2.52b       examples                                   pwntools
afl-latest.tgz  examples.desktop                           shellcode.c
afltest         fuzzedfiles                                Templates
afltest.c       Hands-On-Penetration-Testing-with-Python   Videos
afltest-gcc     Music                                      wtf
cpython         output.file                                wtf_cases
Desktop         peda                                       zzuf
Documents       Pictures
student@ubuntu:~$ gcc -fno-stack-protector -z execstack afifirsttest.c -o afifir
sttest-gcc
afifirsttest.c:2:20: fatal error: string.g: No such file or directory
compilation terminated.
student@ubuntu:~$ sudo -i
root@ubuntu:~# echo core > /proc/sys/kernel/core_pattern
root@ubuntu:~#
```

```
#include <stdio.h>
#include <string.g>

int main(void)
{
        char login[32];
        char passwd[32];
        printf("Login: \n");
        gets(login);
        printf("Password: \n");
        gets(passwd);
        if (strcmp(login, "root") == 0) {
                if (strcmp(passwd, "lqazxsw2") == 0){
                        printf("Access Granted.\n");
                        return 0;
                }
        }
        printf("Access Denied.\n");
        return 1;
}
```

Exercise 3, Step 38: We need to take the crash data as an input and pipe it into our program. Change directory to **./results/crashes**. In the terminal window, enter **cat id\:000000\,sig\:11\,src\:000000\,op\:havoc\,rep\:128 | ../../aflfirsttest-gcc**. Note that "\" is required. Without it, the file will not be found it. An example of the output from the command is shown in the following screenshot.

NOTE: Unable to run the c program despite creating it accordingly. This meant that I was unable to obtain the ~-gcc file accordingly and use it.

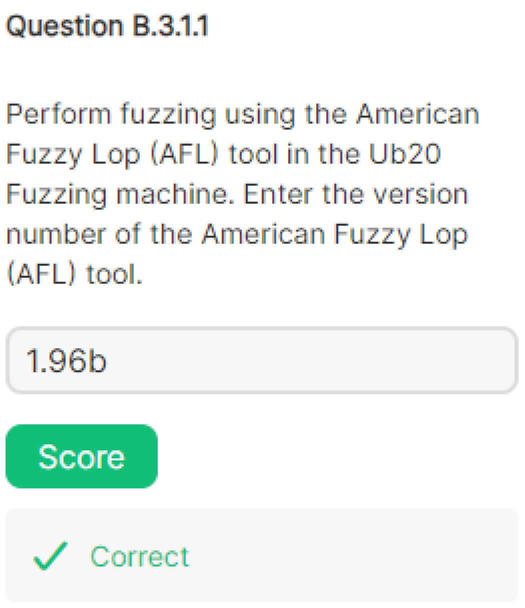


```
student@ubuntu:~$ sudo sysctl kernel.randomize_va_space=0
[sudo] password for student:
kernel.randomize_va_space = 0
student@ubuntu:~$ gcc -fno-stack-protector -z execstack afifirsttest.c -o afifir
sttest-gcc
afifirsttest.c:2:20: fatal error: string.g: No such file or directory
compilation terminated.
student@ubuntu:~$ ls
afifirsttest.c  Downloads                                 Public
afl-2.52b       examples                                  pwntools
afl-latest.tgz  examples.desktop                          shellcode.c
afltest         fuzzedfiles                               Templates
afltest.c       Hands-On-Penetration-Testing-with-Python  Videos
afltest-gcc     Music                                     wtf
cpython         output.file                               wtf_cases
Desktop         peda                                      zzuf
Documents       Pictures
student@ubuntu:~$ gcc -fno-stack-protector -z execstack afifirsttest.c -o afifir
sttest-gcc
afifirsttest.c:2:20: fatal error: string.g: No such file or directory
compilation terminated.
student@ubuntu:~$ sudo -i
root@ubuntu:~# echo core > /proc/sys/kernel/core_pattern
root@ubuntu:~#
```

```
#include <stdio.h>
#include <string.g>

int main(void)
{
        char login[32];
        char passwd[32];
        printf("Login: \n");
        gets(login);
        printf("Password: \n");
        gets(passwd);
        if (strcmp(login, "root") == 0) {
                if (strcmp(passwd, "lqazxsw2") == 0){
                        printf("Access Granted.\n");
                        return 0;
                }
        }
        printf("Access Denied.\n");
        return 1;
}
```

3.2 QUESTIONS



Question B.3.1.1

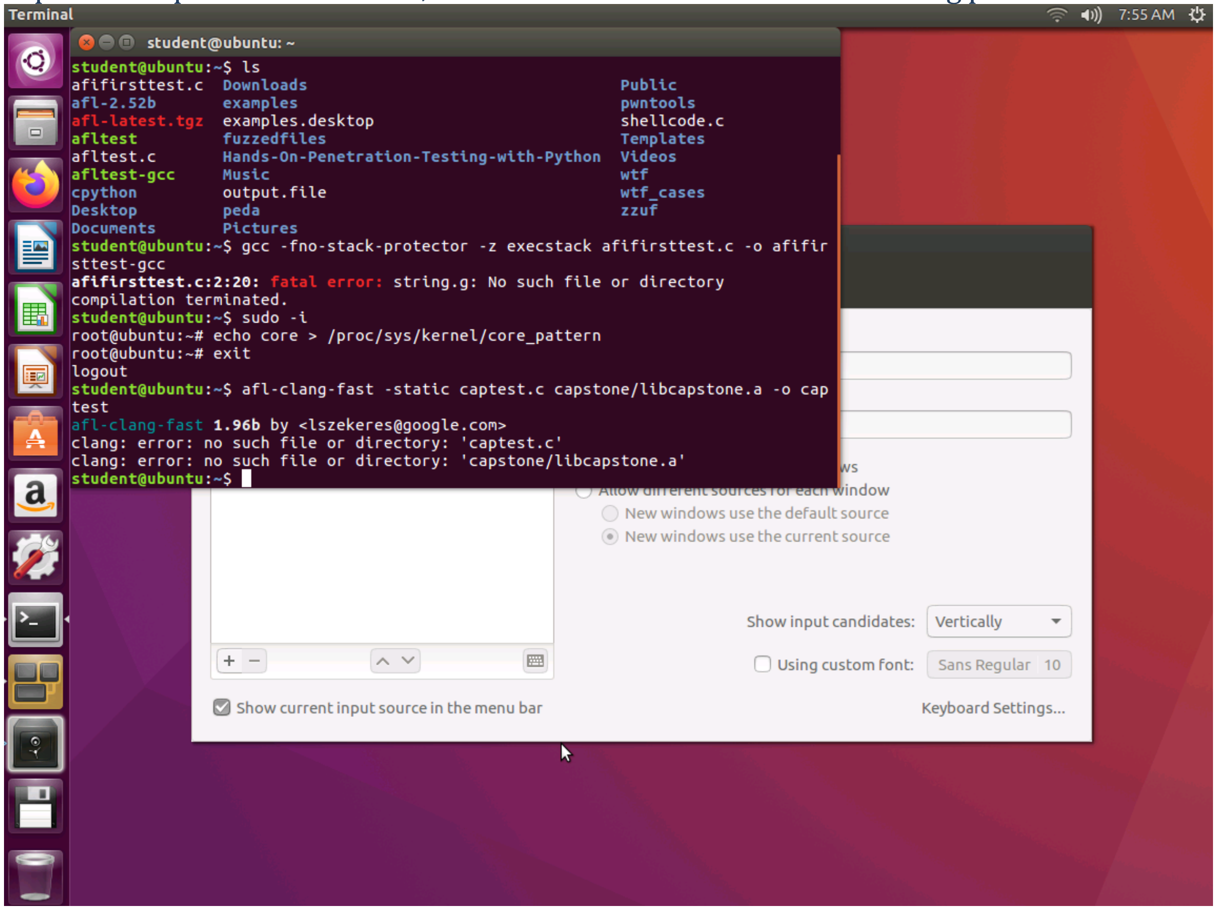
Perform fuzzing using the American Fuzzy Lop (AFL) tool in the Ub20 Fuzzing machine. Enter the version number of the American Fuzzy Lop (AFL) tool.

1.96b

Exercise 4: Fuzzing with AFL and capstone

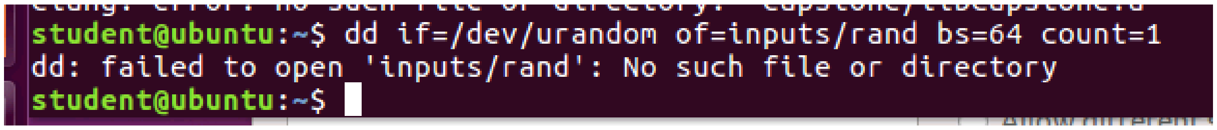
4.1 OUTPUT SCREENSHOTS

Exercise 4, Step 19: **Next, once the code is compiled, we are ready to run the fuzzer again. Type afl-fuzz -i inputs -o findings ./captest2 and press Enter. An example of the output of this command is shown in the following screenshot.**
NOTE: Unable to use the code provided by the lab nor generate the data despite following the steps stated.
NOTE: due to being unable to use the resources provided by the lab due to file and syntax errors, despite having been guided by the lab steps themselves, I was unable to obtain the expected output itself. However, I was still able to learn about the fuzzing process.

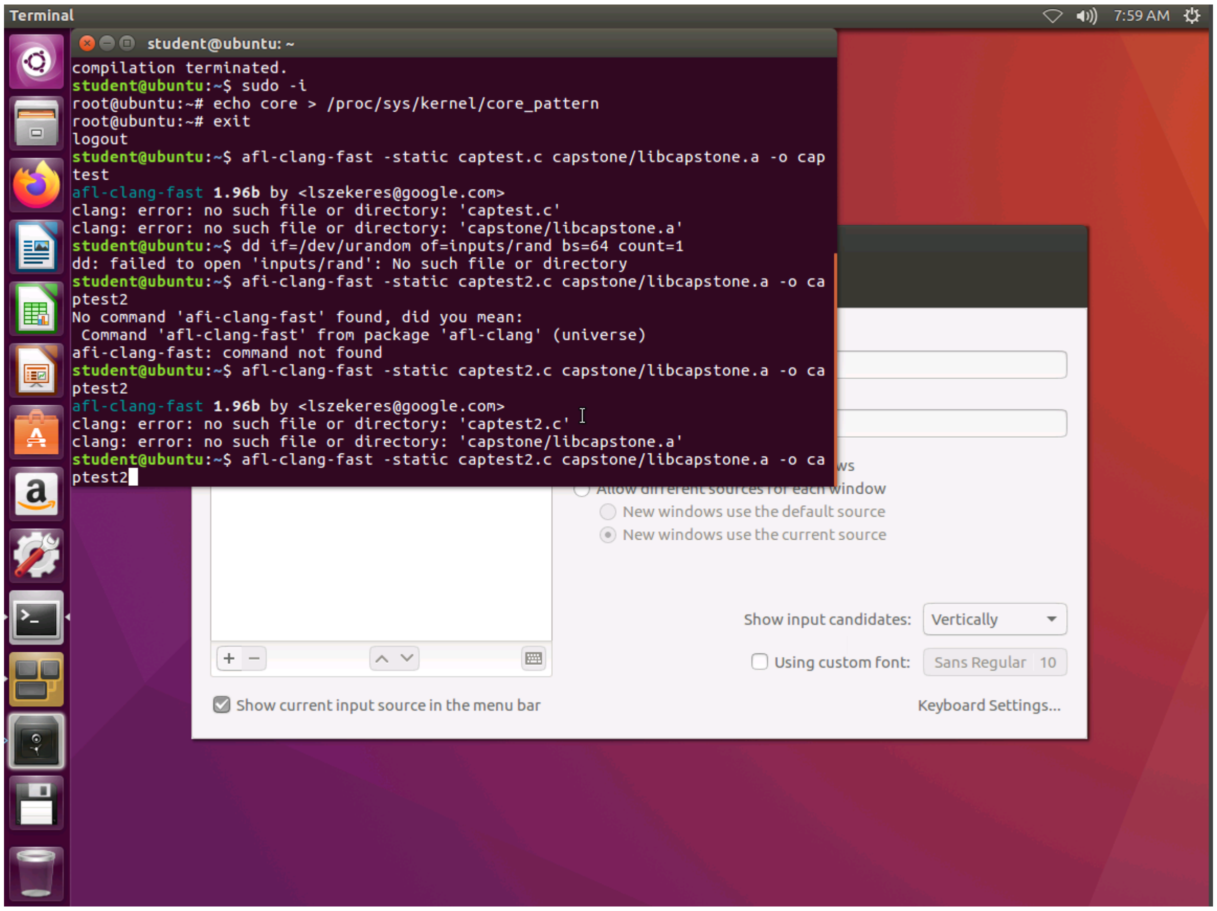


```
student@ubuntu:~$ ls
afifirsttest.c  Downloads                                Public
afl-2.52b       examples                                 pwntools
afl-latest.tgz  examples.desktop                         shellcode.c
afltest         fuzzedfiles                              Templates
afltest.c       Hands-On-Penetration-Testing-with-Python Videos
afltest-gcc     Music                                    wtf
cpython         output.file                              wtf_cases
Desktop         peda                                     zzuf
Documents       Pictures
student@ubuntu:~$ gcc -fno-stack-protector -z execstack afifirsttest.c -o afifir
sttest-gcc
afifirsttest.c:2:20: fatal error: string.g: No such file or directory
compilation terminated.
student@ubuntu:~$ sudo -i
root@ubuntu:~# echo core > /proc/sys/kernel/core_pattern
root@ubuntu:~# exit
logout
student@ubuntu:~$ afl-clang-fast -static captest.c capstone/libcapstone.a -o cap
test
afl-clang-fast 1.96b by <lszekeres@google.com>
clang: error: no such file or directory: 'captest.c'
clang: error: no such file or directory: 'capstone/libcapstone.a'
student@ubuntu:~$
```

New windows use the default source
New windows use the current source
Show input candidates: Vertically
Using custom font: Sans Regular 10
Show current input source in the menu bar
Keyboard Settings...



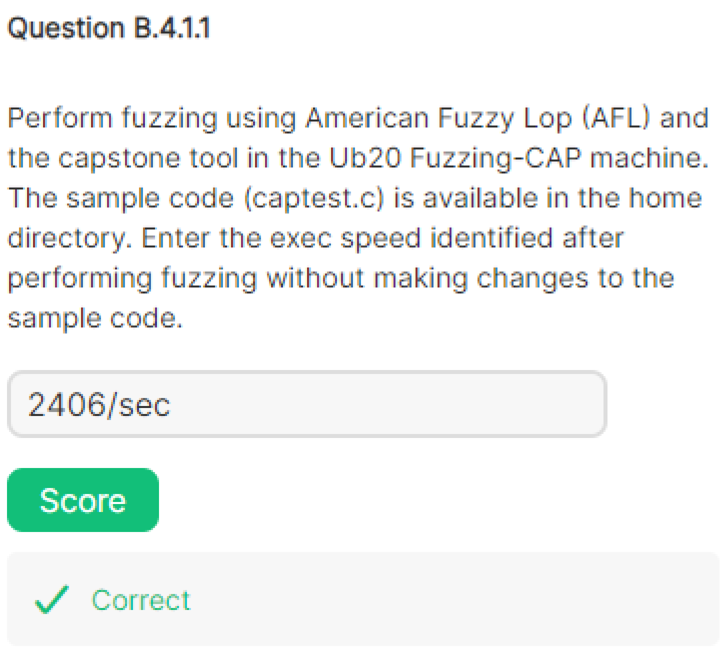
```
student@ubuntu:~$ dd if=/dev/urandom of=inputs/rand bs=64 count=1
dd: failed to open 'inputs/rand': No such file or directory
student@ubuntu:~$
```



```
compilation terminated.
student@ubuntu:~$ sudo -i
root@ubuntu:~# echo core > /proc/sys/kernel/core_pattern
root@ubuntu:~# exit
logout
student@ubuntu:~$ afl-clang-fast -static captest.c capstone/libcapstone.a -o cap
test
afl-clang-fast 1.96b by <lszekeres@google.com>
clang: error: no such file or directory: 'captest.c'
clang: error: no such file or directory: 'capstone/libcapstone.a'
student@ubuntu:~$ dd if=/dev/urandom of=inputs/rand bs=64 count=1
dd: failed to open 'inputs/rand': No such file or directory
student@ubuntu:~$ afi-clang-fast -static captest2.c capstone/libcapstone.a -o ca
ptest2
No command 'afi-clang-fast' found, did you mean:
 Command 'afl-clang-fast' from package 'afl-clang' (universe)
afi-clang-fast: command not found
student@ubuntu:~$ afl-clang-fast -static captest2.c capstone/libcapstone.a -o ca
ptest2
afl-clang-fast 1.96b by <lszekeres@google.com>
clang: error: no such file or directory: 'captest2.c'
clang: error: no such file or directory: 'capstone/libcapstone.a'
student@ubuntu:~$ afl-clang-fast -static captest2.c capstone/libcapstone.a -o ca
ptest2
```

Terminal
7:59 AM
student@ubuntu: ~
Allow different sources for each window
New windows use the default source
New windows use the current source
Show input candidates: Vertically
Using custom font: Sans Regular 10
Show current input source in the menu bar
Keyboard Settings...

4.2 QUESTIONS



Question B.4.1.1

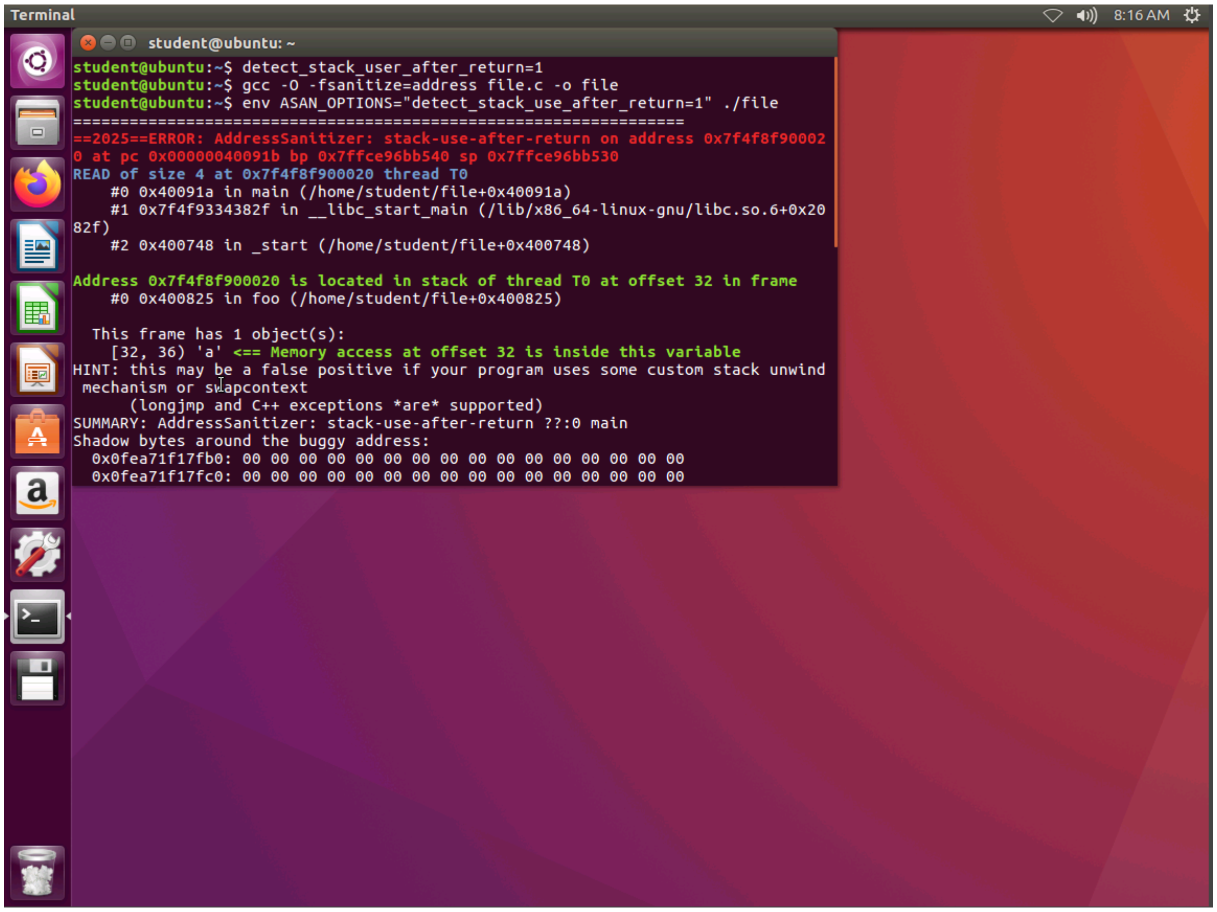
Perform fuzzing using American Fuzzy Lop (AFL) and the capstone tool in the Ub20 Fuzzing-CAP machine. The sample code (captest.c) is available in the home directory. Enter the exec speed identified after performing fuzzing without making changes to the sample code.

2406/sec

Exercise 5: Additional Capabilities of Address Sanitizer

5.1 OUTPUT SCREENSHOTS

Exercise 5, Step 6:
Next, we can add **detect_stack_use_after_return=1** to
the **ASAN_OPTIONS** environment variable before running the program by
entering **env ASAN_OPTIONS="detect_stack_use_after_return=1" ./file.**

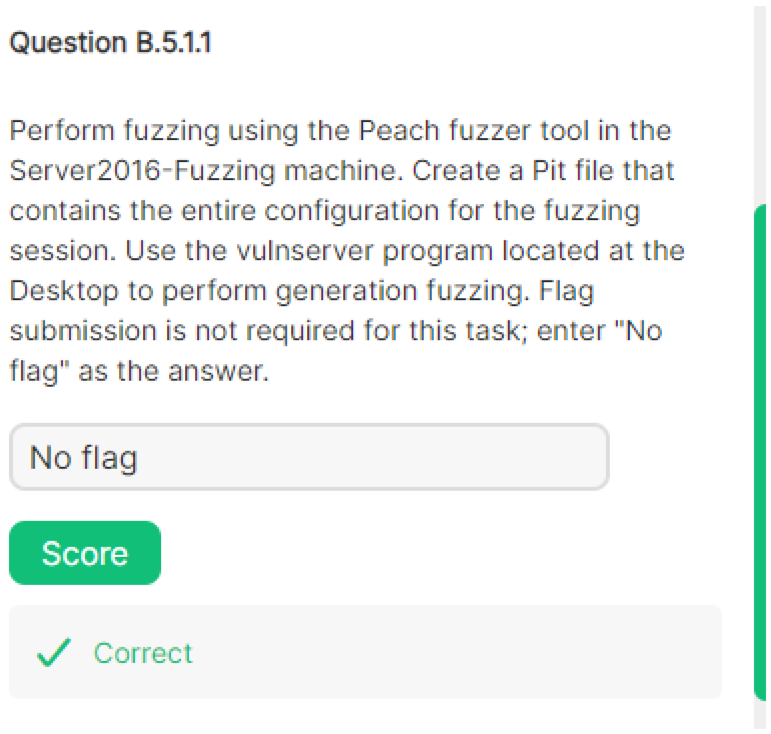


```
student@ubuntu:~$ detect_stack_user_after_return=1
student@ubuntu:~$ gcc -O -fsanitize=address file.c -o file
student@ubuntu:~$ env ASAN_OPTIONS="detect_stack_use_after_return=1" ./file
=================================================================
==2025==ERROR: AddressSanitizer: stack-use-after-return on address 0x7f4f8f90002
0 at pc 0x00000040091b bp 0x7ffce96bb540 sp 0x7ffce96bb530
READ of size 4 at 0x7f4f8f900020 thread T0
    #0 0x40091a in main (/home/student/file+0x40091a)
    #1 0x7f4f9334382f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x20
82f)
    #2 0x400748 in _start (/home/student/file+0x400748)

Address 0x7f4f8f900020 is located in stack of thread T0 at offset 32 in frame
    #0 0x400825 in foo (/home/student/file+0x400825)

  This frame has 1 object(s):
    [32, 36) 'a' <== Memory access at offset 32 is inside this variable
HINT: this may be a false positive if your program uses some custom stack unwind
 mechanism or swapcontext
      (longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-use-after-return ??:0 main
Shadow bytes around the buggy address:
  0x0fea71f17fb0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0fea71f17fc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

5.2 QUESTIONS



Question B.5.1.1

Perform fuzzing using the Peach fuzzer tool in the Server2016-Fuzzing machine. Create a Pit file that contains the entire configuration for the fuzzing session. Use the vulnserver program located at the Desktop to perform generation fuzzing. Flag submission is not required for this task; enter "No flag" as the answer.

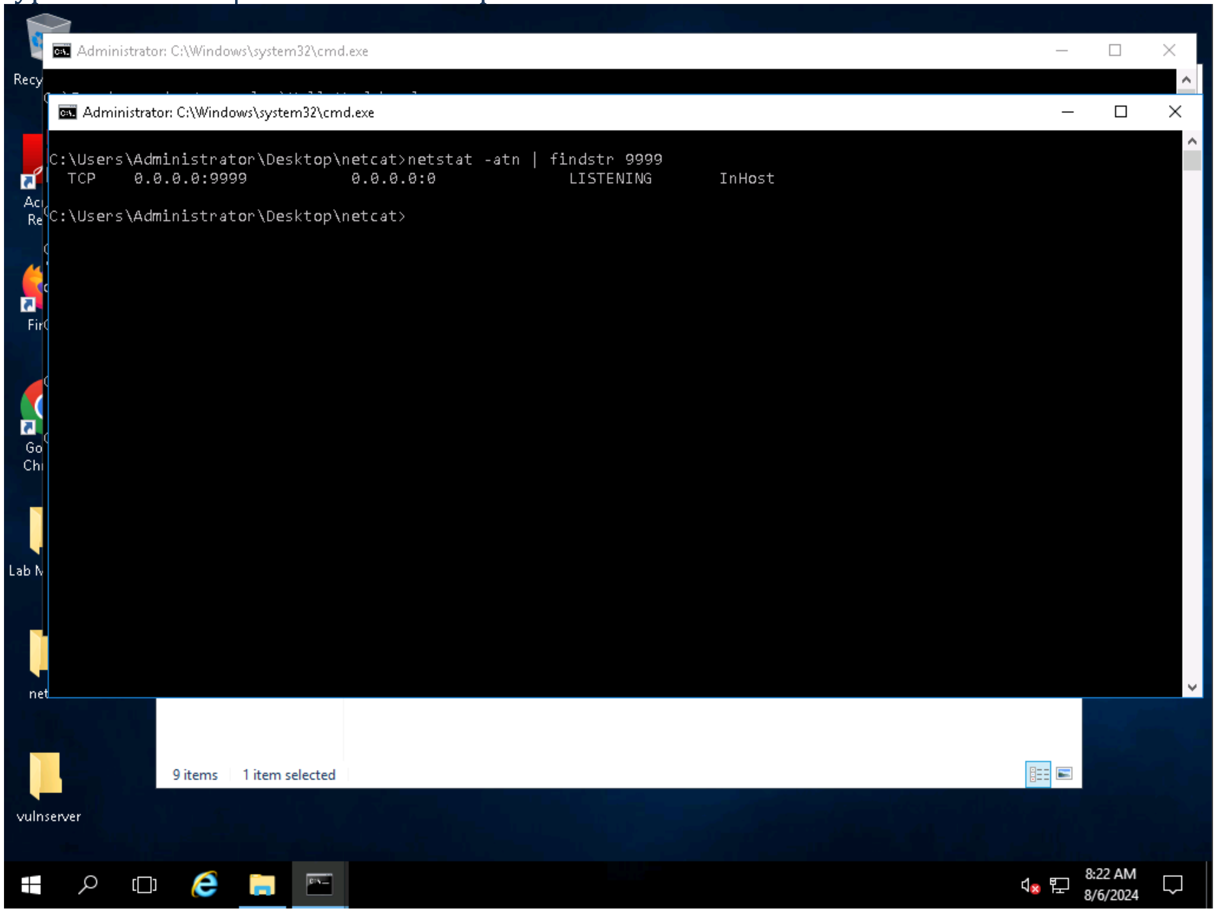
No flag

✓ Correct

Exercise 6: Fuzzing with Peach

6.1 OUTPUT SCREENSHOTS

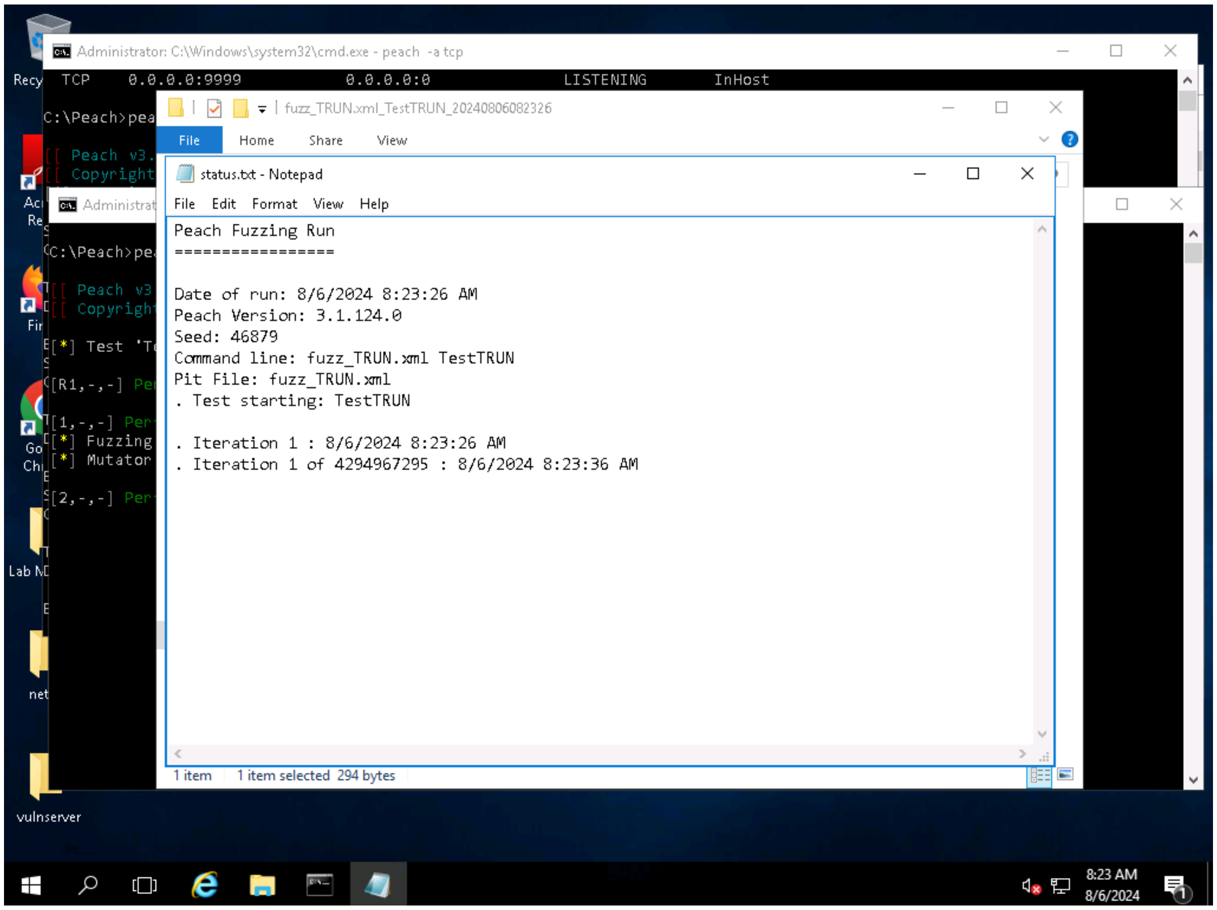
Exercise 6, Step 33: To get an idea of what is occurring, in another command prompt, type netstat -atn | findstr 9999 and press **Enter**.


Administrator: C:\Windows\system32\cmd.exe
Administrator: C:\Windows\system32\cmd.exe
9 items
1 item selected
vulnserver
8:22 AM
8/6/2024

```
C:\Users\Administrator\Desktop\netcat>netstat -atn | findstr 9999
  TCP    0.0.0.0:9999           0.0.0.0:0              LISTENING       InHost

C:\Users\Administrator\Desktop\netcat>
```

Exercise 6, Step 34: You may see warnings, or the vulnerable server may crash and you may need to restart testing again. Depending on how lucky (or unlucky) you are, you may need to generate many test cases—even as many as a thousand to get a reliable crash to debug. The key is the process of reviewing the **Logs** folder.


Administrator: C:\Windows\system32\cmd.exe - peach -a tcp
fuzz_TRUN.xml_TestTRUN_20240806082326
File
Home
Share
View
status.txt - Notepad
File Edit Format View Help
1 item 1 item selected 294 bytes
vulnserver
8:23 AM
8/6/2024

```
TCP    0.0.0.0:9999           0.0.0.0:0              LISTENING       InHost
```

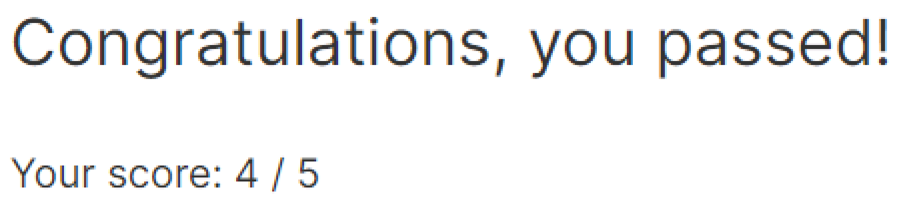
```
Peach Fuzzing Run
=================

Date of run: 8/6/2024 8:23:26 AM
Peach Version: 3.1.124.0
Seed: 46879
Command line: fuzz_TRUN.xml TestTRUN
Pit File: fuzz_TRUN.xml
. Test starting: TestTRUN

. Iteration 1 : 8/6/2024 8:23:26 AM
. Iteration 1 of 4294967295 : 8/6/2024 8:23:36 AM
```

6.2 QUESTIONS

NOTE: There were no questions



Congratulations, you passed!

Your score: 4 / 5