Stephen Yeomans
CS561 SDaT
CTF

All instructions are provided in the containers.

Problem 1

```
└─# cat Readme
#For this challenge, you need to listen on your network for the first flag.
#You know that the port is between 57000 and 57200, and it is being broadcasted via udp.
#You have netcat at your disposal as well as a bash script you can write to.
#Use tee to write to the file, do not use tail unless you can write the script in 10 lines or less.
```

Problem 2

```
└─# cat Readme
#Your next flag is hidden in a process. Find the file and read it for your next instructions.
```

Problem 3

```
└─# cat Readme
#Your first task is to compare these files and find the phrase that is different.
#Use the words that you find different and search for a file containing these words for your next task.
#This file is found in the etc folder
```

Instructions how to complete.

I intentionally got rid of cat, head, and more. I wanted users to find other ways to read/write to files. I did give hints to use tee to write to files as people might try to use tail but struggle since tail will only read the last 10 lines you enter, as intended.

Problem 1

This problem is done by going through all ports from 57000 to 57200. I made it to have to create a bash script but you could do this by hand.
Example bash script:

```
└─$ tail script.sh
#!/bin/bash
i=57000
until [ $i -gt 57200 ]
do
 nc -w 1 -luvp $i
  ((i=i+1))
done
```

Example correct output:

```
listening on [any] 57121 ...
no connection : Connection timed out
listening on [any] 57122 ...
no connection : Connection timed out
listening on [any] 57123 ...
connect to [127.0.0.1] from localhost [127.0.0.1] 45893
f$gh*x
```

Problem 2

You are prompted to look at the current running processes and find where the file is being run. Running 'ps' doesn't net you anything, so you run 'ps aux' or 'ps -e' in order to see all the processes running. After seeing Problem2.sh, you can tail/nl that file which prompts you to look for a file called randomFile3.sh. Using the find command, you can see there are about 9 files that are named this throughout the container. All you must do is look though them all to get the second flag.
Example first and second command:

```
└─$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0   2576   928 pts/0    Ss+  01:08   0:00 /bin/sh -c ./wrapper.sh
root           7  0.0  0.1   4340  3204 pts/0    S+   01:08   0:00 /bin/bash ./wrapper.sh
root           8  0.0  0.1   4340  3180 pts/0    S+   01:08   0:00 /bin/bash /usr/local/games/Problem2.sh
root           9  0.0  0.1   4340  3224 pts/0    S+   01:08   0:00 /bin/bash /home/labuser/CTF/Problem1/Problem1Script.sh
root          23  0.0  0.2  15396  4760 ?        Ss   01:08   0:00 sshd: /usr/sbin/sshd [listener] 0 of 10-100 startups
root          24  0.0  0.0   2936   896 pts/0    S+   01:08   0:00 tail -f /dev/null
root          75  0.0  0.5  17564 10968 ?        Ss   01:09   0:00 sshd: labuser [priv]
labuser       90  0.0  0.3  17824  6896 ?        S    01:09   0:00 sshd: labuser@pts/1
labuser       91  0.0  0.1   4184  3492 pts/1    Ss   01:09   0:00 -bash
root        1319  0.0  0.0   2904   912 pts/0    S+   01:15   0:00 sleep 1
root        1321  0.0  0.0   2904   856 pts/0    S+   01:15   0:00 sleep 1
labuser     1322  0.0  0.1   8084  3948 pts/1    R+   01:15   0:00 ps aux

┌──(labuser㉿5a28718d4c63)-[~/CTF/Problem1]
└─$ tail /usr/local/games/Problem2.sh
#The flag is located in a file called randomFile3.sh
i=1
while :
do
  if [ $i -eq 2 ]; then
    echo
  else
    sleep 1
  fi
done
```

Example third and fourth command:

```
┌──(labuser㉿5a28718d4c63)-[~/CTF/Problem1]
└─$ find / -name "randomFile3.sh"
```

```
└─$ tail /usr/bin/randomFile3.sh
#This file! *3dT3bA2$
i=1
while :
do
  if [ $i -eq 2 ]; then
    echo
  else
    sleep 1
  fi
done
```

Problem 3

For the last flag, you must first find the diferent phrase between the list. After finding the different phrase, you have to search this phrase inside the /etc/ folder to find the next instruction, which is to find the SHA256sum of a phrase to match the one given to you. The phrase is WordsSymbolsNumbers#!, where # is numbers increasing from 0 to 10000. I made this one out of 10000 to really encourage users to write a bash script. They could do it on a different machine but theres only so much you can do. I provided a bash script they could edit inside the folder.

Example first command:

```
└─# diff file1.txt file2.txt
949c949
< decay
---
> water bottle
```

Example second command:

```
┌──(labuser㉿0172a1f751c4)-[/etc]
└─$ grep -rnw "water bottle"
grep: gshadow: Permission denied
grep: .pwd.lock: Permission denied
opt/login:1:water bottle
grep: shadow: Permission denied
grep: security/opasswd: Permission denied
grep: gshadow-: Permission denied
grep: shadow-: Permission denied
grep: ssh/ssh_host_ecdsa_key: Permission denied
grep: ssh/ssh_host_rsa_key: Permission denied
grep: ssh/ssh_host_ed25519_key: Permission denied
```

Example third command:

```
┌──(labuser㉿1dcf3b6ccdbf)-[~]
└─$ tail /etc/opt/login
water bottle
#Now create a script checking the sha256sum and find the correct number for the flag
#The string to check against is WordsSymbolsNumbers#! where # increases from 0 to 10000.
#There is no leading 0s, so the first one to check against would be WordsSymbolsNumbers0!
#The number you fill in that matches the sha256sum is the flag for this level
#If using echo, the sha256sum is f103d7afc8a3011f9d93e5380cbe98a969e19ad735a1cfdfa9188fa955182bde
#If using printf, the sha256sum is a7a34e63d19d1eeb42cc97cd7493f51649a1abe17e4648324ca6b58146588158
#You can find the script file in the Problem3 directory
#Use tee to write the file, do not use tail unless you can write the file in 10 lines or less.
```

Example script for third flag:

```
┌──(labuser㉿1dcf3b6ccdbf)-[~/CTF/Problem3]
└─$ ./script.sh
6590

┌──(labuser㉿1dcf3b6ccdbf)-[~/CTF/Problem3]
└─$ nl script.sh
     1  #!/bin/bash
     2  str="f103d7afc8a3011f9d93e5380cbe98a969e19ad735a1cfdfa9188fa955182bde"
     3  b=1
     4  until [[ $b -gt 10000 ]]; do
     5      echo "WordsSymbolsNumbers${b}!" > /home/labuser/CTF/Problem3/textForScript.txt
     6      e=$(sha256sum /home/labuser/CTF/Problem3/textForScript.txt)
     7      g=${e%  *};
     8      if [[ "$g" == "$str" ]]; then
     9        echo "$b"
    10      fi
    11      b=$((b+1))
    12  done
```

The final flag would be CTF_SDat{f$gh*x_*3dT3bA2$_6590}