

## **Lab 2**

### **Objectives:**

1. Understand and implement access control
2. Filter network traffic through IPTables
3. File permissions
4. Write functions

**Task 1.** Given the security levels TOP SECRET, SECRET, CONFIDENTIAL, and UNCLASSIFIED (ordered from highest to lowest), and the categories A, B, and C, specify what type of access (read, write, both, or neither) is allowed in each of the following situations. Assume that discretionary access controls allow anyone access unless otherwise specified. Simple security property says that a subject can write to object if subject compartment dominates object compartment. \*-property says that subject can write to object if object compartment dominates subject compartment. Let  $(L, C)$  and  $(L', C')$  be compartments for different entities.  $((L, C) \text{ dominates } (L', C') \Leftrightarrow L' \leq L \text{ and } C' \subseteq C)$  is the principle we are going to apply to specify what type of access that the following sentences have.

(a) Alice, cleared for (top secret, {A,C}) wants to access a document classified (secret, {A,C})  
According to the BLP model she will have read access only to that document

(b) Bob, cleared for (Confidential, {C}), wants to access a document classified (Unclassified, {A,B})  
Bob doesn't have access to the document at all, they are not connected in any way

(c) Claire, cleared for (Secret, {C}), wants to access a document classified (Unclassified, {C}).  
According to the BLP model she will have read access only to that document

(d) Harry, cleared for (Top secret, {A,C}), wants to access a document classified (Confidential, {B}).  
Harry should not have access to that document

(e) Loma has no clearance, but wants to access a document classified (Confidential, {B}).  
Loma will only have write access to that document but not read access

**Task 2.** Classify each of the following system as an example of a mandatory, discretionary, or attribute based access control policy. In each system, state who is the creator, who is the owner of the object, what is the system, who is the admin of the system, and who decides the permission for your selected type of access control.

(a) In a Linux system, a file's permission is set by the owner of the file.

Classification: Discretionary access control

Creator: the owner

Owner: the owner

System: Linux

Admin: the owner

Who decides permission: the owner of the file

(b) In a software repository, a file can be accessible to an agent based on the owner's choice.

Classification: Discretionary access control

Creator: the user who created the initial file

Owner: the user who created the file or owns it

System: Software Repository

Admin: the person who is entrusted with managing the repository (owner or repo)

Who decides permission: the owner of the file

(c) In a classified NSA database, only generals with top secret clearance can search in the database.

Classification: Role-based access control

Creator: someone within the NSA

Owner: most likely the NSA

System: NSA database

Admin: person or group within the NSA

Who decides permission: the permission is determined by security and the NSA most likely has different ranks to determine who gets access, so the NSA decides and we are told that only generals have clearance to search in this database

**Task 3.** Consider a computer system with three users: Alice, Bob, and Cyndy. Alice owns the file a, and Bob and Cyndy can read it. Bob owns the file b, and Cyndy can read and write the file b, but Alice can only read it. Cyndy owns the file c, but neither Alice and Bob can read or write it. If a user owns a file, he/she can also execute the file.

(a) Create the ACM (access control matrix) of the system.

	File A	File B	File C
Alice	RWX	R	
Bob	R	RWX	
Cyndy	R	RW	RWX

(b) Show the ACL of the ACM and the CL of the ACM.

$Alice \rightarrow [(A, Rwx), (B, R)]$	$A \rightarrow [(Alice, Rwx), (Bob, R), (Cyndy, R)]$
$Bob \rightarrow [(A, R), (B, Rwx)]$	$B \rightarrow [(Alice, R), (Bob, Rwx), (Cyndy, RW)]$
$Cyndy \rightarrow [(A, R), (B, RW), (C, Rwx)]$	$C \rightarrow [(Cyndy, Rwx)]$

(c) Cyndy allows Alice to read c, and Alice removes Bob's ability to read a. Show the ACM after these changes.

	File A	File B	File C
Alice	RWX	R	R
Bob		RWX	
Cyndy	R	RW	RWX

**Task 4.** Assume the following struct is declared for the permission list of a file in Linux. Each permission (u or g or o) is represented as an octal. For example, u=7 means rwx, u=5 means r-x.

Note: "unsigned char" means a byte, not a character or a letter or a string.

```
typedef struct {
    unsigned int uid; // owner id
    unsigned int gid; // group id
    unsigned char u; // owner's permission
    unsigned char g; // group's permission
    unsigned char o; // other's permission
} Permission;
```

The permission check procedure is

- (1) A user requests an operation p on a file f.
- (2) If the user is the owner of the file, the operation will be checked against the owner's permission of the file. The result is either grant or deny.
- (3) Otherwise, if the user is not the owner but in the group of the file, the operation will be checked against the group's permission of the file. The result is either grant or deny.

(4) Otherwise, if the user is neither the owner nor in the group of the file, the operation will be checked against the other's permission of the file. The result is either grant or deny.

Write a C/C++ function "int accesscheck(unsigned int uid, unsigned int gid, unsigned int p, int f)" to enforce access control in Linux.

The arguments of the function accesscheck are explained below:

(1) uid and gid are the user id and the group id of the user who requests to take an operation on the file.

(2) f is the file id.

(3) p is the requested operation. For example, p=7 means three operations rwx, p=6 means two operations rw-, p=1 means one operation --x. The function return 1 if access is permitted, otherwise 0.

Request will be granted only if p is contained by the permission set of the file. Assume "Permission getPermission(int f)" can get the permission of the file f.

For example, f's permission is rwxr-xr-x 1000(uid) 2000(gid). Then, accesscheck(1000, 1000, 6, f) returns 1, but accesscheck(2000, 2000, 6, f) returns 0.

Copy and paste your code in report and explain each line of code of your function in comments.

```
int accesscheck(unsigned int uid, unsigned int gid, unsigned int p, int f) {  
  
    Permission new_user = getPermission(f);  
  
    // Check against owners ID  
    if (uid == new_user.uid) {  
        // checking against the owner's permission  
        if ((new_user.u & p) == p) {  
            // if it is the same as the owner's permission i will accept  
            return 1;  
        }  
    }  
    // Check against group's ID  
    } else if (gid == new_user.gid) {  
        // checking against the group's permission  
        if ((new_user.g & p) == p){  
            // if it is the same as the group's permission i will accept  
            return 1;  
        }  
    }  
    // Check against other's permission  
    } else if ((new_user.o & p) == p) {  
        // if it is the same as the permission i will grant access  
        return 1;  
    }  
    else {  
        //otherwise deny  
        return 0;  
    }  
    return 0;  
}
```

**Task 5.** You have a Linux server, but without any firewall rules in place for protection. You want to use iptables to protect the web server so that:

\* Only computers from address 172.90.0.0/16 but excluding 172.90.255.0/24 can browse the web site that is running on port 8888 in the server.

- Sudo iptables -A INPUT -p tcp --dport 8888 -j ACCEPT
- Sudo iptables -A INPUT -s 172.90.0.0/16 -j ACCEPT
- Sudo iptables -A INPUT -s 172.90.0.0/16 -j DROP

\* Only you can ssh and ping to the server from your IP 172.90.10.20.

- Sudo iptables -A INPUT -p icmp --icmp-type echo-request -s 172.90.10.20 -j ACCEPT

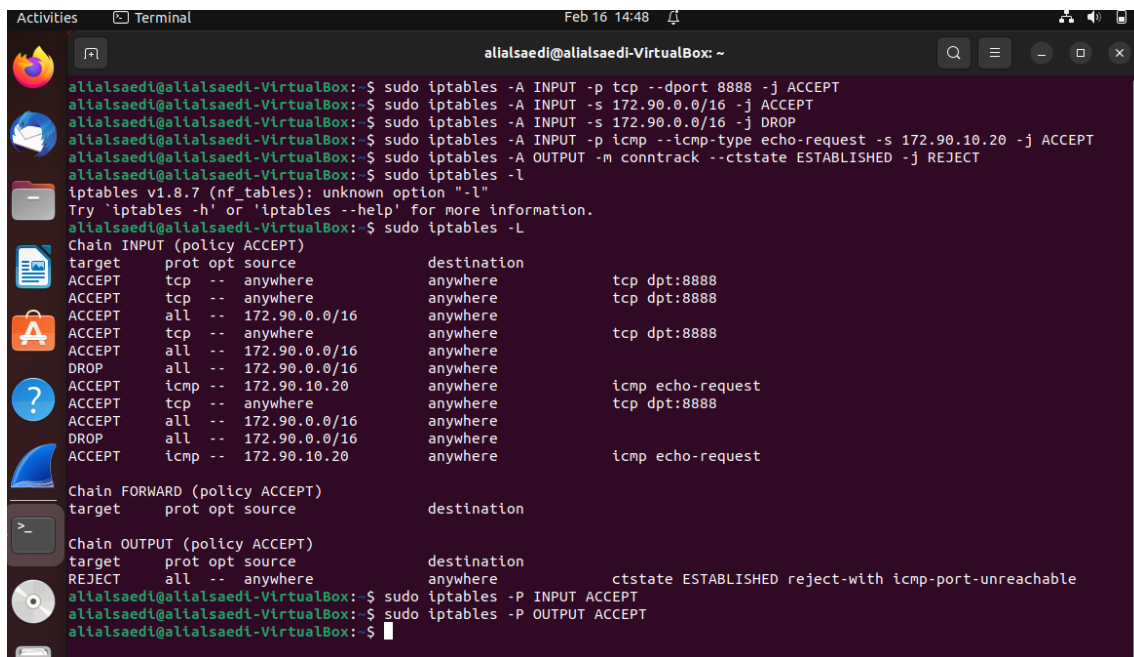
\* The server cannot initiate anything to send out.

- Sudo iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED -j REJECT

\* The default policy for all chains is ACCEPT. Show the commands of iptables that add the Linux firewall rules to enforce the protection.

- Sudo iptables -P INPUT ACCEPT
- Sudo iptables -P OUTPUT ACCEPT

Here are all the commands in order



```
ali@ali:~$ sudo iptables -A INPUT -p tcp --dport 8888 -j ACCEPT
ali@ali:~$ sudo iptables -A INPUT -s 172.90.0.0/16 -j ACCEPT
ali@ali:~$ sudo iptables -A INPUT -s 172.90.0.0/16 -j DROP
ali@ali:~$ sudo iptables -A INPUT -p icmp --icmp-type echo-request -s 172.90.10.20 -j ACCEPT
ali@ali:~$ sudo iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED -j REJECT
ali@ali:~$ sudo iptables -L
iptables v1.8.7 (nf_tables): unknown option "-l"
Try 'iptables -h' or 'iptables --help' for more information.
ali@ali:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT    tcp  --  anywhere              anywhere            tcp dpt:8888
ACCEPT    tcp  --  anywhere              anywhere            tcp dpt:8888
ACCEPT    all  --  172.90.0.0/16         anywhere
ACCEPT    tcp  --  anywhere              anywhere            tcp dpt:8888
ACCEPT    all  --  172.90.0.0/16         anywhere
DROP      all  --  172.90.0.0/16         anywhere
ACCEPT    icmp --  172.90.10.20          anywhere            icmp echo-request
ACCEPT    tcp  --  anywhere              anywhere            tcp dpt:8888
ACCEPT    all  --  172.90.0.0/16         anywhere
DROP      all  --  172.90.0.0/16         anywhere
ACCEPT    icmp --  172.90.10.20          anywhere            icmp echo-request

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
REJECT    all  --  anywhere              anywhere            ctstate ESTABLISHED reject-with icmp-port-unreachable
ali@ali:~$ sudo iptables -P INPUT ACCEPT
ali@ali:~$ sudo iptables -P OUTPUT ACCEPT
ali@ali:~$
```