

Linux User, Group & Permission Practice Lab

Lab Objective

এই ল্যাব শেষে স্টুডেন্ট পারবে—

- নতুন user ও group তৈরি করতে
 - user কে একাধিক group-এর member করতে
 - file ও directory permission বুঝে পরিবর্তন করতে
 - ownership ও group ownership change করতে
 - ACL (special permission) ব্যবহার করতে
-

Lab Environment

- OS: Ubuntu / CentOS / Rocky Linux
- Login: root অথবা sudo user
- Working directory: /lab

```
mkdir /lab
```

```
cd /lab
```

LAB-1: useradd দিয়ে User Create

Goal

নতুন user তৈরি করা

```
useradd student1
```

```
passwd student1
```

 Verify:

```
id student1
```

 Expected:

- user create হবে
 - home directory: /home/student1
-

LAB-2: groupadd দিয়ে Group Create

Goal

নতুন group তৈরি করা

```
groupadd linuxadmin
```

```
groupadd developers
```

 Verify:

```
getent group linuxadmin
```

LAB-3: gpasswd দিয়ে User কে Group Member করা


Goal

student1 কে linuxadmin group-এর member করা

gpasswd -a student1 linuxadmin

 Verify:

id student1

 Note:

- user এখন primary group ছাড়াও secondary group এ আছে

LAB-4: usermod দিয়ে Group Modify


Goal

student1 কে developers group-এও add করা

usermod -aG developers student1

 Verify:

groups student1

 Important:

- -aG না দিলে আগের group remove হয়ে যাবে

LAB-5: File & Directory Create

mkdir projectA

touch projectA/file1.txt

ls -ld projectA


ls -l projectA

LAB-6: chmod দিয়ে Permission Change

Goal

Owner কে full, group কে read, others কে no access

chmod 740 projectA

 Check:

ls -ld projectA

 Meaning:

- 7 → rwx (owner)
- 4 → r-- (group)
- 0 → --- (others)

LAB-7: chown দিয়ে Ownership Change (User)

Goal

projectA এর owner student1 করা

chown student1 projectA

 Verify:

ls -ld projectA

LAB-8: chgrp / chown :group দিয়ে Group Change

Goal

projectA এর group developers করা

chgrp developers projectA

OR

chown :developers projectA

 Verify:

ls -ld projectA

LAB-9: setfacl দিয়ে Special Permission (ACL)

Scenario

student1 ছাড়া আরেক user student2 কে শুধু read/write access দিতে হবে

Step-1: user create

useradd student2

passwd student2

Step-2: ACL set করা

setfacl -m u:student2:rw projectA/file1.txt

 Verify ACL:

getfacl projectA/file1.txt

 Result:

- student2 file access পাবে
 - group/others permission change না করেই
-

LAB-10: ACL Remove (Optional)

setfacl -x u:student2 projectA/file1.txt

OR

setfacl -b projectA/file1.txt

Final Practice Task (Student Assignment)

- 1 user create করো: user3
 - 2 group create করো: security
 - 3 user3 কে security group এ add করো
 - 4 /lab/securedata directory তৈরি করো
 - 5 owner → user3
 - 6 group → security
 - 7 ACL দিয়ে student1 কে read only access দাও
-

LAB-11: Execution Permission & Binary Logic Test (Very Important)

Scenario

একটি কোম্পানি এমন একটি script ব্যবহার করে যা run করলে বর্তমান date & time দেখায়।
কিন্তু security policy অনুযায়ী—

- শুধুমাত্র নির্দিষ্ট user script execute করতে পারবে
- অন্য user গুলো execute করতে পারবে না
- file এর ভিতরের command কেউ modify করতে পারবে না

Tasks

1 Script File তৈরি করো

```
mkdir -p /lab/scripts
```

```
nano /lab/scripts/showdate.sh
```

File content:

```
#!/bin/bash
```

```
date
```

Save করে বের হও।

2 User & Group Rule

ধরে নাও আগের user গুলো আছে:

- dev1
- dev2
- audit1

Rules:

- dev1 → script execute করতে পারবে
- dev2 → script execute করতে পারবে
- audit1 → execute করতে পারবে না

3 Permission Setup (Thinking Required)

 Constraint:

- script readable থাকবে
- audit1 execute করতে পারবে না
- chmod 777 ব্যবহার করা যাবে না

Student কে নিজে decide করতে হবে—

- owner
- group
- permission bits

 Hint:

- execution bit (x)
- group membership logic

4 Testing & Proof (Mandatory)

dev1 দিয়ে test:

su - dev1

/lab/scripts/showdate.sh

✓ Expected: date output দেখাবে

audit1 দিয়ে test:

su - audit1

/lab/scripts/showdate.sh

✗ Expected:

Permission denied

5 Explain (Written Answer)

Student কে লিখতে হবে—

- 1 কেন audit1 execute করতে পারছে না
 - 2 execute bit না থাকলে Linux কীভাবে block করে
 - 3 read permission থাকলেও execute কেন আলাদা
-

🔥 BONUS (Advanced Execution Logic)

Extra Rule যোগ করো:

- audit1 → file পড়তে পারবে
- কিন্তু execute করতে পারবে না

📌 Hint:

chmod 740 showdate.sh

Explain করো—

- r-- vs --x
- script vs binary execution difference


🚩 Lab Summary (Command Mapping)


Task	Command
User create	useradd
Group create	groupadd
Group member add	gpasswd / usermod
Permission change	chmod
Ownership change	chown
Group change	chgrp
Special permission	setfacl


LAB Test

Linux User, Group & Permission

Practical LAB Test (Complex Level)

 Time: 90 minutes

 Passing Criteria: Output + Verification

 Rule: Root বা sudo access ব্যবহার করতে পারবে

QUESTION-1: Multi-User & Multi-Group Scenario

Scenario

তোমার প্রতিষ্ঠানে একটি project team আছে।

Tasks

1 নিচের user গুলো তৈরি করো

- dev1
- dev2
- audit1

2 নিচের group গুলো তৈরি করো

- devteam
- auditteam

3 Rules:

- dev1, dev2 → devteam member
- audit1 → auditteam member
- dev1 → auditteam member **হবে না**

4 Verify:

- প্রতিটি user কোন কোন group এ আছে তা দেখাও

 Expected Skill:

- useradd
- groupadd
- gpasswd / usermod
- id / groups

QUESTION-2: Ownership & Permission Logic

Scenario

/lab/projects নামে একটি directory আছে

Tasks

1 Directory তৈরি করো

2 Owner হবে dev1

3 Group হবে devteam

4 Permission:

- dev1 → full access

- devteam → read + write
- others → no access

 Hint:

- chmod numeric value বুঝতে হবে
- chown + chgrp বা combined chown ব্যবহার করা


QUESTION-3: File Access Conflict (Thinking Required)

Scenario


/lab/projects/source.c নামে একটি file আছে

Rules

- dev2 → read & write
- audit1 → only read
- others → no access

 Constraint:

- group permission change করা যাবে না
- owner পরিবর্তন করা যাবে না

 Expected:

- ACL ব্যবহার বাধ্যতামূলক
- getfacl output দেখাতে হবে

QUESTION-4: Hidden Problem (Common Mistake Trap)

Scenario

student ভুল করে নিচের কমান্ড চালিয়েছে:

usermod -G devteam dev2

Tasks

- 1 সমস্যা কী হয়েছে ব্যাখ্যা করো
- 2 কীভাবে এটা fix করবে
- 3 dev2 কে আবার আগের সব group এ ফিরিয়ে আনো

 Expected Skill:

- usermod -aG
- group membership understanding

QUESTION-5: Shared Directory with Special Rule

Scenario

/lab/shared নামে একটি directory

Rules

- devteam → create, edit, delete files
- auditteam → only read files
- new files → automatically devteam group পাবে

 Hint:

- chmod
- chgrp
- special permission (setgid bit)


QUESTION-6: ACL Override Test (Advanced)

Scenario

/lab/shared/report.txt

Rules

- audit1 → read only
- dev2 → full access
- others → no access

 Important:

- chmod দিয়ে সম্ভব না
- ACL ব্যবহার করতে হবে

 Expected:

- setfacl
- effective permission বুঝতে হবে

QUESTION-7: Troubleshooting Based Question

Scenario

dev2 বলছে:

“আমি directory দেখতে পারছি, কিন্তু file খুলতে পারছি না”

Tasks

- 1 সম্ভাব্য 2টি কারণ লেখো
- 2 কোন কোন command দিয়ে সমস্যা diagnose করবে
- 3 minimum permission দিয়ে fix করো

 Expected Thinking:

- directory execute permission
- file read permission

QUESTION-8: Clean-up & Reset Task

Scenario

Exam শেষে system clean করতে হবে

Tasks

- 1 সব ACL remove করো /lab এর ভেতর
- 2 devteam group delete করো
- 3 audit1 user delete করো
- 4 Verify করে দেখাও কিছু leftover নেই

📌 Commands expected:

- setfacl -b
 - userdel
 - groupdel
 - getfacl / getent
-

🧠 BONUS (Very Hard – Optional)

👉 এমন একটি file বানাও যেখানে:

- owner → full
- group → no access
- specific user → write only
- others → read only

! Explain করো কেন numeric chmod দিয়ে এটা সম্ভব না

🌿 QUESTION–9: Basic Script Execution Control

Scenario

একটি script run করলে system date দেখাবে।

Tasks

- 1 /lab/exec/showdate.sh নামে একটি script তৈরি করো
- 2 Script এর ভিতরে date command থাকবে
- 3 user1 script execute করতে পারবে
- 4 user2 script execute করতে পারবে না

Test

- user1 দিয়ে run করে output দেখাও
- user2 দিয়ে run করে prove করো “Permission denied”

📌 Expected Skill:

- chmod
 - execute bit (x)
 - ls -l output analysis
-

🌿 QUESTION–10: Read Allowed, Execute Denied

Scenario

Security team চায় কেউ script পড়তে পারবে কিন্তু execute করতে পারবে না।

Tasks

- 1 user2 script content দেখতে পারবে
- 2 user2 script execute করতে পারবে না
- 3 user1 full access পাবে

Output Required

- cat showdate.sh (user2)
- ./showdate.sh (user2 → fail)



Focus:

- r vs x difference



QUESTION-11: Group Based Execution (Thinking Required)

Scenario

একটি group execgroup আছে।

Rules

- execgroup members → script execute করতে পারবে
- group এর বাইরে কেউ → execute করতে পারবে না

Tasks



1 Script owner root থাকবে



2 exec permission শুধুমাত্র group কে দেওয়া যাবে



3 Others execute পাবে না



Hint:

- chgrp
- chmod (group execute)



QUESTION-12: ACL Based Execution Control (Advanced)

Scenario

chmod ব্যবহার করা যাবে না।

Rules

- user1 → execute allowed
- user2 → execute denied
- owner/group unchanged থাকবে

Tasks



1 ACL ব্যবহার করে permission set করো



2 getfacl output দেখাও



Expected:

- setfacl -m u:user1:x
- ACL priority understanding



QUESTION-13: Directory Execute Permission Trap

Scenario

Script এর permission ঠিক আছে, কিন্তু execute হচ্ছে না।

Tasks



1 Diagnose করো সমস্যা কোথায়



2 Minimum permission দিয়ে fix করো



3 Explain করো কেন directory তে x দরকার



Hint:

- directory execute ≠ file execute
-


QUESTION-14: Execute Without Read (Conceptual + Practical)

Scenario

একজন user script execute করতে পারবে কিন্তু script পড়তে পারবে না।

Tasks

- 1 Permission সেট করো
- 2 Prove করো:
 - execute → works
 - cat → permission denied

 Thinking:

- --x permission logic
-

QUESTION-15: Wrong Permission Recovery

Scenario

Student ভুল করে চালিয়েছে:

```
chmod 777 showdate.sh
```

Tasks

- 1 কেন এটা insecure explain করো
 - 2 Proper permission redesign করো
 - 3 Execute access only limited users দাও
-


QUESTION-16: Execution Audit Proof

Scenario

Examiner চায় evidence।

Tasks

- 1 ls -l output দেখাও
- 2 user switch করে execute test করো
- 3 Terminal output screenshot equivalent text দাও

 Evaluation:

- proof without explanation = fail
-

BONUS

 এমন একটি script বানাও যেখানে—

- owner → execute only
- group → read only
- others → no access

Explain করো:

- script execute করতে kernel কী check করে