Exam Report: 8.9.9 Practice Questions

Date: 4/27/2020 10:57:40 am
Time Spent: 5:16

Candidate: Garsteck, Matthew
Login: mGarsteck

Overall Performance

Your Score: 29%

Passing Score: 80%

View results by: Objective Analysis Individual Responses

Individual Responses

Question 1: Incorrect

You must change ownership of a script called *myscript* so that only the user owner has read/write access to it and only the user owner and group owner have execute permissions.

Which of the following commands will accomplish this task?

- **chmod 750 myscript**
- **chmod 611 myscript**
- **chmod 771 myscript**
- ~~**chmod 755 myscript**~~

Explanation

chmod 750 is correct because it allows the user owner full read, write, and execute permissions, and the group owner read and execute permissions.

POSIX file permissions can be displayed in a number of ways. The traditional method is rwxrwxrwx. With this method, the *r* means read, the *w* means write, and the *x* means execute. These permissions are displayed for the user owner (first set); group owner (second set); and the world, or everyone (third set).

POSIX permissions can also be converted into binary using three bits, one for read, one for write, and one for execute. In this notation, rwxrwxrwx would be 111111111. Displaying the permissions in binary is a bit long-winded, so binary permissions are often converted into octal numbers (digits 0-7). In octal, rwxrwxrwx would be represented as 777(111 binary = 4+2+1 = 7 octal).

chmod 771 is incorrect because it gives execute permissions to everyone. **chmod 611** is incorrect because it does not give write permissions to the group owner. **chmod 755** is incorrect because it gives execute and read permissions to everyone and does not give write permissions to the group owner.

References

Linux Pro - 8.9 Permissions
[e_chmod_lp5.exam.xml Q_PERMFCT_LP5_01]

Question 2: Correct

Which of the following sets of octal permissions would allow everyone to execute the following file as if they were the root user?

-rwxr-xr-x 1 root root 8045 July 24 2018 myscript

- **0755**
- **4755**
- **4744**

- **1655**

Explanation

4755 is correct because it sets the SUID bit for everyone to execute the file as if they were the root user.

File permissions can be displayed in a number of ways. The traditional method is rwxrwxrwx. With this method, *r* means read, *w* means write, and *x* means execute. These permissions are displayed for the user owner (first set), group owner (second set) and the world, or everyone (third set).

Permissions can also be converted into binary using three bits, one for read, one for write, and one for execute. In this notation, rwxrwxrwx would be 111111111. Displaying the permissions in binary is a bit long-winded, so binary permissions are often converted into octal numbers (digits 0-7). In octal, rwxrwxrwx would be represented as 777 (111 binary = 4+2+1 = 7 octal).

A fourth group of rwx (111 binary or 7 octal) can be appended to the beginning of the permissions to set SUID (100 binary or 4 octal), SGID (010 binary or 2 octal), and/or the sticky bit (001 binary or 1 octal). This is represented as 0777 octal if none of these are set.

0755 is incorrect because even though everyone can execute the file, it will be executed with the permissions of the user that executes it. **4744** is incorrect because even though the SUID bit is set, only the user owner can execute it. **1655** is incorrect because the group owner and everyone can execute it, but it will be executed with the permissions of the user that executes it.

References

Linux Pro - 8.9 Permissions
[e_chmod_lp5.exam.xml Q_PERMFCT_LP5_02]

Question 3: Incorrect

There is a directory called *projects* owned by the user **cmarcela** and the group *rd*. **cmarcela** has left the company. You need to give ownership of the projects directory, its files, and its subdirectories to the user **ebates**, who is a member of the *mgmt* group.

Which of the following commands should you use to change the user ownership?

- **chown .ebates projects**
- **chown -R ebates.mgmt projects**
- ~~**chown ebates projects**~~
- **chown -R ebates projects**
- **chgrp -R mgmt projects**

Explanation

The **chown -R ebates projects** command allows you to change the user ownership for the project directory recursively (**-R**).

The **chown ebates projects** command changes ownership only for the projects directory, not for its files or subdirectories. The **chgrp -R mgmt projects** command changes the group ownership of the projects directory recursively. The **chown -R ebates.mgmt projects** command changes user and group ownership of the directory recursively.

References

Linux Pro - 8.9 Permissions
[e_chmod_lp5.exam.xml Q_PERMFCT_LP5_03]

Question 4: Incorrect

Which of the following sets of permissions represent the minimal permissions required to allow a user to list the contents of a directory?

- rw-
- r-x

~~rwx~~

r--

Explanation

To list the contents of a directory, a user must have the read (r) permission and the execute (x) permission.

References

Linux Pro - 8.9 Permissions
[e_chmod_lp5.exam.xml Q_PERMFCT_LP5_04]

Question 5: Incorrect

You have a critical file called *yearend.xls*. You have set the file permissions so that only the owner of the file can modify it and only group owners can read it.

Which of the following file listings show that you have set the permissions correctly?

~~-r-xr-x--- 1 george acctg 26256 June 1 04:16 yearend.xls~~

-r--r--r-- 1 george acctg 26256 June 1 04:16 yearend.xls

-rwxrwxr-- 1 george acctg 26256 June 1 04:16 yearend.xls

-rw-r----- 1 george acctg 26256 June 1 04:16 yearend.xls

Explanation

The permissions of-*rw-r*----- allows the user to read or write to the file, the group to only read the file, and nobody else any rights to the file.

The permissions of *-rwxrwx*--- allows the user and group full rights to the file and no rights to anyone else.

The permissions of *-r-xr-x*--- allows the user and group read and execute rights to the file and no rights to anyone else.

The permissions of *-r--r--r--* allows the user, group, and the world read-only rights to the file.

References

Linux Pro - 8.9 Permissions
[e_chmod_lp5.exam.xml Q_PERMFCT_LP5_05]

Question 6: Incorrect

You are called into the office of a newly hired manager. He has copied a file from his old place of work onto his new workstation, but is now receiving an error message that access is denied each time he tries to access the file. He is not the owner of the file or a member of the group the file is associated with.

Which of the following represents the LEAST set of file permissions needed for him to be able to read and write to the file?

11
1

222

~~444~~

666

Explanation

Because he is not a member of the group or the owner of the file, the only set of permissions applying to the manager is the last number. The ability to read the file has a value of 4, while the ability to write to the file has a value of 2. Adding the two together, the needed permission for the user is 6.

References

Linux Pro - 8.9 Permissions
[e_chmod_lp5.exam.xml Q_PERMFCT_LP5_06]

Question 7: Correct

A service account is created by the system or an application and cannot be used to log in to the system.

Which of the following methods can be used to verify that a service account cannot login to the system?

- View the entry for the service account in /etc/shadow and look for /sbin/nologin.
- Verify that file and directory permissions have been removed for the service account to the /boot partition.
- View the ACLs for /bin/login to ensure that the service account is not listed.
- View the entry for the service account in /etc/passwd and look for /sbin/nologin.

Explanation

View the entry for the service account in /etc/passwd and look for /sbin/nologin. This ensures that the service account will not have permission to log in.

The other options do not provide any information related to a service account's permission to login.

References

Linux Pro - 8.9 Permissions
[e_chmod_lp5.exam.xml Q_PERMFCT_LP5_SERVICE_ACCT]