Chapter 4:
ACL and SELinux management

Learning objectives

Upon completing this chapter, the learner should be able to:

- Create and manage access control lists (ACLs)
- Diagnose and correct permission problems
- Understanding ACL mask
- Understanding default ACL
- Describe Security Enhanced Linux and its terminology
- Understanding SELinux working modes
- Understanding context settings and policy

Key terms

ACL access control list

ACL mask

ACL default

setfacl

getfacl

SELinux

semanage

setenforce

getenforce

Permissive mode

Enforced mode

Disabled mode

Policy

Context

Rules

Labels

Table of content

1. Working with ACL

Linux system does not allow giving permissions to more than one user or one group on the same file, another downside is that it does not allow administrators to set default permissions in a sophisticated way, where the permissions that are set can differ on different directories. Access Control Lists (ACL) subsystem do offers these features besides adding other great functionalities to your server.

Although there is one drawback: Not all utilities support it. Therefore, you might lose ACL settings when copying or moving files, and your backup software might not be able to back up ACL settings.

For instance, the tar utility does not support ACLs, so to make sure that your ACL settings are not lost when you make a backup, use star instead of tar. star works with exactly the same options as tar, it just adds support for ACL settings as well.

ACLs are often applied to directories as a structural measure and not on individual files. Therefore, you will not have lots of them, but just a few applied in smart places in the file system. Hence, it is relatively easy to restore the original ACLs you were working with, even if your backup software does not support them.

1.1. Preparing Your File System for ACLs

Before working with ACL on none xfs filesystems, the acl package must be installed on the system:

```
[root@server1 /]# yum install acl
```

And then the file system metadata needs to be prepared and extended, because there is not always default support for ACLs in the file system, as in the ext2, 3 and 4 file systems. So if while setting ACLs to a file system you are getting an “operation not supported” message, your file system probably lacks support for ACLs.

If you are not using the xfs filesystem in partitioning the hard disk, it’s important to make sure that the filesystem has been mounted with the ‘acl’ option. To check the existence of acl on the sda1 partition use: tune2fs -l /dev/sda1 command and notice the option acl among other options.

If your filesystem has not been mounted with the acl option, you can re-mount it giving the needed option:

```
# mount -o remount -o acl /dev/sda1
```

However, notice that the mount options set this way, will not be persistent, and will not survive a reboot. If you want to obtain persistence, you have to modify the filesystem mount options in /etc/fstab, assigning the acl option statically:

/dev/sda1 / ext4 acl, errors=remount-ro 0 1

1.2. Viewing ACL Settings with getfacl

Before setting ACLs, it is always a good idea to show current ACL settings using getfacl, the ls -l command does not show any existing ACLs; it just shows a + after the listing of the permissions, which indicates that ACLs apply to the file as well.

In Listing below, you can see the current permissions as shown with ls -l and also as shown with getfacl. If you look closely enough, you can see that the information shown is exactly the same.

```
[root@server1 ~]# ls -ld dir
drwxr-xr-x. 2 root root 6 Feb 6 11:28 dir
[root@server1 ~]# getfacl dir
 #file: dir/
 #owner: root
 #group: root
user::rwx
group::r-x
other::r-x
```

In the result of the getfacl command, you can see that the permissions are shown for three different entities: the user, the group, and others.

1.3. Changing ACL Settings with setfacl

Now let's add an ACL to give read and execute permissions to the group sales as well, the command to use for this is setfacl -m g:sales:rx. /dir, where:

- -m indicates that the current ACL settings need to be modified
- g:sales:rx tells the command to set the ACL to read and execute (rx) for the group (g) sales

In Listing below, you can see what the command looks like, as well as the output of the getfacl command after changing the current ACL settings.

```
[root@server1 ~]# setfacl -m g:sales:rx dir
[root@server1 ~]# getfacl dir
 #file: dir/
 #owner: root
 #group: root
user::rwx
group::r-x
group:sales:r-x
mask::r-x
other::r-x
[root@server1:~]# ll -d dir/
drwxr-xr-x+ 2 root root 6 Dec  3 23:50 dir/
[root@server1:~]#
```

Notice the + sign after the permissions when listing the ~/dir directory, this means that ACL is applied on this directory. Now that you understand how to set a group ACL, it is easy to understand ACLs for users and others as well. For instance, the command setfacl -m u:user1:rwx dir gives permissions to user user1 on the dir directory without making her the owner and without changing the current owner assignment.

Now if it's needed to remove an entry from the acl permissions the -x option is used for that, see below:

```
[root@server1:~]# setfacl -x g:sales dir/ && getfacl dir/
 #file: dir/
 #owner: root
 #group: root
user::rwx
group::r-x
mask::r-x
other::r-x
```

While the option -b enables removing any ACLs permissions set on file or directory: setfacl -b dir.

The setfacl command has many possibilities and options, the option -R which makes the ACL setting for all files and subdirectories currently existing in the directory where you set the ACL. It is a good idea to always use this option while changing ACLs for existing directories.

1.4. ACL mask

When applying ACL rules on a file or directory, the getfacl shows a new entry called mask, this defines the union of all permissions of the owning group, and all named user and group entries. These are exactly the entries affected by the mask entry, but has no effect on the permissions for the file owner and the other permission group. In the previous example, only reading and executing permissions could be assigned with setfacl command.

Of course we can change this option, the command setfacl -m mask:r dir will set the mask to allow only read permissions, and by changing the mask, we have effectively limited their permissions to read only. As the output of the command shows, they now are only allowed to read the file, see the #effective entry.

```
root@server1:~# setfacl -m mask:r dir
root@server1:~# getfacl dir/
 #file: dir/
 #owner: root
 #group: root
user::rwx
group::r-x                      #effective:r--
group:sales:r-x                 #effective:r--
mask::r--
other::r-x
```

Other than explicitly changed with the command above, the ACLs mask also gets automatically re-calculated when we assign or change permissions with setfacl or the chmod commands (unless the -n option is specified).

Let's demonstrate that, we will change the permissions of the sales group to rwx and then check the getfacl output:

```
root@server1:~# setfacl -m g:sales:rwx dir && getfacl dir/
 #file: dir/
 #owner: root
 #group: root
user::rwx
group::r-x
group:sales:rwx
mask::rwx
other::r-x
```

As you can see the mask got re-calculated and it now reflects the maximum permissions present for the named group sales. Obviously, since now no previously set permissions are higher than the mask, there is no need for showing the #effective permission status. but applying the -n option as below will prevent recalculating the mask as the maximum for the sales group entry.

```
[root@server1:~]# setfacl -n -m g:sales:rwx dir && getfacl
dir
 #file: dir
 #owner: root
 #group: root
user::rwx
group::r-x                      #effective:r--
group:sales:rwx                 #effective:r--
mask::r--
other::r-x
```

1.5. Working with Default ACLs

Another benefit is that you can enable inheritance by working with default ACLs, when setting a default ACL, you'll determine the permissions that will be set for all new items that are created in the directory. Be aware, though, that a default ACL does not change the permissions for existing files and subdirectories, to change those as well you need to add a normal ACL.

To set a default ACL, you just have to add the option `d` after the option `-m` (order does matter!). So, use `setfacl -m d:g:sales:rx /data` if you want group sales to have read and execute on everything that will ever be created in the `/data` directory.

When using default ACLs, it can also be useful to set an ACL for others. Normally, this does not make much sense because you can also change the permissions for others by using chmod. What you cannot do using chmod, though, is specify the permissions that should be given to others on every new file that will ever be created. If you want others not to get any permissions on anything that is created in `/data`, for example, use `setfacl -m d:o::- /data`.

Exercise: Managing advanced permissions using ACLs

1. Create the following directories: /data/account and /data/sales.
2. Change the owners of these directories using chown linda. sales /data/sales and chown linda. account /data/account.
3. Ensure the permissions to be chmod 770 /data/sales, and next chmod 770 /data/account.
4. Make sure that the group account gets read permissions on the /data/sales directory and that group sales gets read permissions on the /data/account directory. By using setfacl –m g:account:rx /data/sales and setfacl –m g: sales:rx /data/account.
5. Use getfacl to verify that the permissions have been set the way you intended to.

6. Then, you set default ACLs to make sure that on all new files the permissions are properly set for all new items. By using setfacl –m d:g:account:rwx, g:sales:rx /data/sales.

7. Add the default ACL for the directory /data/account by using setfacl –m d:g:sales: rwx, g:account:rx /data/account.

8. Verify that the ACL settings are effective by adding a new file in /data/sales. Use touch /data/sales/newfile and use getfacl /data/sales/newfile to check the current permission assignments.

2. Understanding SELinux working and modes

Security Enhanced Linux (SELinux) is an implementation of the Mandatory Access Control (MAC) architecture developed for flexible, enriched, and granular security controls in Linux. MAC is integrated into the Linux kernel as a set of patches using the Linux Security Modules (LSM) framework that allows the kernel to support various security implementations, including SELinux.

MAC provides an added layer of protection above and beyond the standard Linux Discretionary Access Control (DAC) security architecture, which includes the traditional file and directory permissions, ACL settings, setuid / setgid bit settings, su / sudo privileges, and so on. MAC limits the ability of a subject (user or process) to access an object (file, directory, file system, device, network interface, port, pipe, socket, etc.) in order to reduce or eliminate the potential damage the subject may be able to cause to the system if compromised due to the exploitation of vulnerabilities in service processes, programs, or applications.

In SELinux to specify what exactly is allowed, a policy is used. In this policy, rules define which source domain is allowed to access which target domain:

- The source domain is the object that is trying to access something. Typically, this is a process or a user
- The target domain is the object that is accessed. Typically, that is a file, directory, or a network port to define exactly what is allowed, context labels are used

These labels are the essence of SELinux because they are used to define access rules. The following table summarizes it all.

Element	Use
Policy	A collection of rules that define which source has access to which target.
Source domain	The object that is trying to access a target. Typically, a user or a process.
Target domain	The thing that a source domain is trying to access. Typically, a file or port.
Context	A security label that is used to categorize objects in SELinux.
Rule	A specific part of the policy that determines which source domain has which access permissions to which target domain.
Labels	Same as context label, defined to determine which source domain has access to which target domain.

SELinux is often disabled on servers because of laziness and because application vendors just do not know how to deal with it, if it is enabled and nothing else has been configured, all system calls are denied. On many occasions, even applications that do not know how to work with SELinux can be fully functional on a server with SELinux. It just takes a bit more work to figure out the additional rules in the policy that need to be created to use the application on an SELinux-enabled system.

- When SELinux is enabled, kernel support for SELinux is loaded, and some applications that are SELinux aware change their behavior, because specific libraries are used on a system that has SELinux enabled
- If SELinux is disabled, no SELinux activity will be happening at all

Changing between enabled and disabled mode requires a reboot of your system. This is because SELinux is a feature that is deeply related to the Linux kernel.

2.1. Controlling activation mode

If on a system SELinux is enabled, you can select to put it one of two modes:

- In enforcing mode, SELinux is fully operational and enforcing all SELinux rules in the policy and allows or denies actions based on its rules.
- Where in permissive mode, all SELinux related activity is logged, but no access is blocked. This makes this mode an excellent mode to do troubleshooting. Permissive mode is also a great way to do something and see the result from an SELinux perspective, that can help in building new and more efficient policies.

One of the key configuration files that controls the SELinux activation mode and sets its default type is called config and it is located in the /etc/selinux directory. The default contents of this file are displayed below, followed by an explanation:

```
[root@server1 ~]# cat /etc/selinux/config
 #This file controls the state of SELinux on the system.
 #SELINUX= can take one of these three values:
 #enforcing - SELinux security policy is enforced.
  #permissive - SELinux prints warnings instead of
enforcing.
 #disabled - No SELinux policy is loaded.
SELINUX=enforcing
 #SELINUXTYPE= can take one of these two values:
 #targeted - Targeted processes are protected,
 #minimum - Modification of targeted policy. Only selected
processes are protected.
 #mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

The SELINUX directive in the file sets the activation mode for SELinux, the enforcing and permissive. The third option is to completely disable it.

When activated in enforcing mode, the SELINUXTYPE directive dictates the type of policy to be enforced. SELinux supports three policies: targeted, minimum, and mls (multi-level security):

- The targeted policy allows us to modify SELinux restrictions placed on subjects and objects, and is the default policy in RHEL7
- the minimum policy is a light version of the targeted policy
- and the mls policy lets us tighten security at more granular levels

At the command prompt, we can issue the getenforce command to determine the current operating mode:

```
[root@server1 ~]# getenforce
Enforcing
```

We may alter the SELinux operating state immediately from enforcing to permissive using the setenforce command as follows:

```
[root@server1 ~]# setenforce permissive (or # setenforce 0)
```

This change, however, will be lost if the system is rebooted. To make the mode change persistent across reboots, we edit /etc/selinux/config and set the value of SELINUX to permissive. We can then reboot the system to validate the change.

Switching between enforcing and permissive is easy and can be achieved on the fly. However, disabling SELinux completely requires SELINUX to be set to disabled in the config file and the system must be rebooted.

In the future, if we wish to reactivate SELinux in either enforcing or permissive mode, we just need to change the SELINUX directive appropriately in the config file and reboot the system. The reboot will take longer than normal as SELinux will go through the process of relabeling the files.

Another useful command is sestatus. If used with the option -v, this command shows detailed information about the current status of SELinux on a server. It not only shows you which parts of SELinux are enabled but also shows the current version of the policy that is loaded and the context labels for some critical parts of the system.

```
[root@server1 ~]# sestatus -v
SELinux status: enabled
SELinuxfs mount: /sys/fs/selinux
SELinux root directory: /etc/selinux
Loaded policy name: targeted
Current mode: enforcing
Mode from config file: enforcing
Policy MLS status: enabled
Policy deny_unknown status: allowed
Max kernel policy version: 28
Process contexts:
Current context:
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
Init context: system_u:system_r:init_t:s0
/usr/sbin/sshd system_u:system_r:sshd_t:s0-s0:c0.c1023
File contexts:
Controlling terminal: unconfined_u:object_r:user_devpts_t:s0
/etc/passwd system_u:object_r:passwd_file_t:s0
/etc/shadow system_u:object_r:shadow_t:s0
/bin/bash system_u:object_r:shell_exec_t:s0
/bin/login system_u:object_r:login_exec_t:s0
/bin/sh system_u:object_r:bin_t:s0<-
system_u:object_r:shell_exec_t:s0
/sbin/agetty system_u:object_r:getty_exec_t:s0
/sbin/init system_u:object_r:bin_t:s0<-
system_u:object_r:init_exec_t:s0
/usr/sbin/sshd system_u:object_r:sshd_exec_t:s0
```

Exercise: Manipulating SELinux modes

1. Open a root console on your server and type getenforce. You'll normally see that SELinux is in enforcing mode.

2. Type setenforce 0 and type getenforce again. SELinux now switches to permissive mode.

3. Open the file /etc/sysconfig/selinux with an editor and change the line SELINUX= so that it reads SELINUX=disabled. Reboot your server.

4. After rebooting, log in to a root shell again and type getenforce. You'll see that SELinux is now in disabled mode.

5. Try using the command setenforce 1. You'll see the message "setenforce: SELinux is disabled." You cannot switch between disabled and enforcing mode without rebooting your server.

6. Open the file /etc/sysconfig/selinux again and change the line

   SELINUX=disabled back to SELINUX=enforcing. Reboot your system again.

7. After rebooting, type sestatus -v and read current status information about SELinux.

2.2. SELinux policy

The SELinux Policy is the set of rules that guide the SELinux security engine, it specifies the rules in the implemented environment. It is written in a language created specifically for writing security policy rules, these rules are preprocessed into many additional rules as part of building the policy.conf file, which is compiled into the binary policy.

According to the activated SELinux type the active policy files location will differ. For example, if SELINUXTYPE is set to targeted, the active policy files location will be in /etc/selinux/targeted/ which is the root directory for the targeted policy.

2.3. Working with context settings

Context settings are an important part of SELinux operations, the context is a label that can be applied to different elements: Files and directories, Ports, Processes or Users. Context labels define the nature of the item, and SELinux rules are created to match context labels of source objects to the context labels of target objects. So, setting correct context labels is a very important skill for system administrators.

2.3.1. Monitoring current context labels

To see current context settings on these objects, many commands offer support for the -Z option. Listing shows how ls -Z shows context settings for some directories in the /file system. Other commands also support the -Z option to show current context label settings. Some examples are ps Zaux, which shows a list of all processes, including their context label, or netstat -Ztulpen, which shows all network ports and the current context label associated with each port.

```
[root@server1 /]# ls -Z
lrwxrwxrwx. root root system_u:object_r:bin_t:s0 bin -> usr/bin
dr-xr-xr-x. root root system_u:object_r:boot_t:s0 boot
drwxr-xr-x root root ? dev
drwxr-xr-x. root root system_u:object_r:etc_t:s0 etc
drwxr-xr-x. root root system_u:object_r:home_root_t:s0 home
lrwxrwxrwx. root root system_u:object_r:lib_t:s0 lib -> usr/lib
lrwxrwxrwx. root root system_u:object_r:lib_t:s0 lib64 -> usr/
lib64
...
drwxrwxrwt. root root system_u:object_r:tmp_t:s0 tmp
drwxr-xr-x. root root system_u:object_r:usr_t:s0 usr
drwxr-xr-x. root root system_u:object_r:var_t:s0 var
drwxr-xr-x. root root system_u:object_r:var_t:s0 var
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0
web
```

Every context label always consists of three different parts:

- **User**: The user can be recognized by _u in the context label; it is set to system_u on most directories in the previous Listing.
- **Role**: The role can be recognized by _r in the context label. In the previous Listing Listing most objects are labeled with the object_r role.
- **Type**: The type context can be recognized by _t in the context label. In the previous Listing you can see that a wide variety of context types is applied to the directories in the /file system.

For example, to list the processes contexts use the ps -Z command:

```
[root@server1 /]# ps -Z
LABEL                                PID TTY          TIME
CMD
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 6617
pts/0 00:00:00 sudo
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 6623
pts/0 00:00:00 su
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 6624
pts/0 00:00:00 bash
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 8188
pts/0 00:00:00 ps
```

And to view the SELinux context associated with your Linux user, use the id –Z command:

```
[root@server1 /] # id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

2.4. Setting context types using semanage

As an administrator, it is important that you know how to set context types. You can set these context types on files and directories and other objects such as network ports. To set context type use semanage command, it writes the new context to the SELinux policy, from which it is applied to the file system. The semanage is a tool used to configure certain elements of SELinux policy without modifying or recompiling policy sources. This includes mapping Linux usernames to SELinux user identities and security context mappings for objects like network ports, interfaces, and hosts.

For example, by default, SELinux only allows known services to bind to known ports. If we want to modify a service to use a non-default port we will need to modify the port type with the semanage command.

2.4.1. Listing ports with semanage

Lets start by listing ports with semanage, the basic command for listing all ports is:

```
[root@server1 /]# semanage port -l
SELinux Port Type              Proto    Port Number
afs3_callback_port_t           tcp      7001
afs3_callback_port_t           udp      7001
afs_bos_port_t                  udp      7007
afs_fs_port_t                    tcp      2040
afs_fs_port_t                    udp      7000, 7005
afs_ka_port_t                   udp      7004
afs_pt_port_t                    tcp      7002
afs_pt_port_t                    udp      7002
```

To list port numbers of a specific port like http, use this command:

```
[root@server1 /]# semanage port -l | grep -w http_port_t
```

http_port_t tcp 80, 81, 443, 488, 8008, 8009, 8443, 9000

2.4.2. Creating or adding ports

Now we will create a new port for http and assign it to tcp port 2222. The -a option is to add a new port, the -t option specifies the SELinux type, and the -p option is to specify the protocol to use (in this case tcp).

```
[root@server1 /]# semanage port -a -t http_port_t -p tcp
2222
```

To view the newly created port, we use the command list command with the -C option to show only customizations.

```
[root@server1 /]# semanage port -lC
SELinux Port Type Proto Port Number

http_port_t                    tcp      2222
```

To override an existing port that was already created, use the -m option to modify:

```
[root@server1 /]# semanage port -m -t unreserved_port_t -p
tcp 2222
[root@server1 /]# semanage port -lC
SELinux Port Type              Proto    Port Number

unreserved_port_t              tcp      2222
```

2.4.3. Deleting ports

We use the option -d to delete a port record. To delete unreserved_port_t on tcp port 2222, we use the command:

```
[root@server1 /]# semanage port -d -t unreserved_port_t -p
tcp 2222
```

Quiz

Chapter review questions

1. If you want to obtain persistency to acl mount option you have to modify the filesystem mount options in:
   a. /etc/fstab
   b. mount command
   c. fstab command
   d. /etc/inittab

2. Which of the following commands grants rw permissions for the group sales to all new files that will be created in the /data directory and all of its subdirectories:
   a. setfacl –m d:g:sales:rw /data
   b. setfacl –m d:g:sales:rwx /data
   c. setfacl –R –m g:sales:rwx /data
   d. setfacl –R –m g:sales:rw /data

3. Which command enables you to make sure that others have no access to any new files that will be created in the /data directory, assuming that you want others to have read permissions on all other files?
   a. setfacl –m d:o::– /data
   b. setfacl –m o::– /data
   c. umask 027 /data
   d. umask 027

4. What would the command setfacl –m u::7, g::4, o:4, u:user3:7 file do?
   a. assign rwx permissions to file owner and user3, and read-only permission to everyone else
   b. assign rw permissions to file owner and user3, and read-only permission to everyone else
   c. assign rwx permissions to file owner and user, and read-only permission to everyone else
   d. assign r-x permissions to file owner and user3, and read-only permission to everyone else

5. When recalculating the ACL mask value after a setfacl command, it's done by:
   a. getting the minimum permissions of the acceptable acl entries
   b. getting the intersection permissions of the acceptable acl entries
   c. getting the maximum permissions of the acceptable acl entries
   d. getting the last used permissions of the acceptable acl entries

6. To change the ACL mask to grant read and write permissions on a directory d1 use:
   a. setfacl -m mask:r d1
   b. getfacl -m mask:rw d1
   c. setfacl -x mask:rw d1
   d. setfacl -m mask:rw d1

7. To remove all ACL entries defined on a file we use:
   a. setfacl -b
   b. setfacl -x
   c. setfacl -m
   d. setfacl -z

8. To set the default ACL permissions to read and execute to a group named g1 on a files in directory d1 use:
   a. setfacl –m d:g:g1:rx d1 */
   b. setfacl –d m:g:g1:rx d1 */
   c. setfacl –m d:g:g1:rx d1 /
   d. setfacl –d m:g:g1:rx d1 /

9. Setting a default ACL, you'll determine the permissions that will be set for all items in the directory.
   a. True
   b. False

10. Setting a default ACL can be applied on files as well as on directories.
   a. True
   b. False

11. A specific part of the policy that determines which source domain has which access permissions to which target domain.
   a. Context
   b. Labels
   c. Rule
   d. sub Policy

12. A collection of rules that define which source has access to which target.
   a. Context
   b. Labels
   c. Rule
   d. Policy

13. A security label that is used to categorize objects in SELinux.

a. Context

b. Labels

c. Rule

d. Policy

14. Which of the following is not a valid SELinux mode?

a. Enforcing

b. Permissive

c. Disabled

d. Enabled

15. Which of the following commands enables you to see the current SELinux mode?

a. sestatus

b. lsmode

c. semode

d. getenforce

16. You want to put SELinux temporarily in permissive mode. Which command do you use?

a. # setenforce enforcing

b. # setenforce permissive

c. # getenforce permissive

d. # getenforce permissive

17. To which of the following can SELinux security not be applied?

a. Users

b. Files

c. Ports

d. It can be applied to all of the above

18. Which command – line switch is used with many commands to display SELinux – related information?

a. –S

b. –X

c. –Z

d. –D

19. Which file do you need to change if you want to completely disable SELinux?

a. /etc/selinux/config

b. /var/log/selinux/

c. /selinux/config

d. /etc/selinux.conf

20. Which command can be used to ensure modified SELinux contexts survive file system relabeling?

a. chcon

b. semanage

c. setenforce

d. getenforce

Answers to chapter Review Questions:

1. a

2. a

3. a

4. a

5. c

6. d

7. a

8. c

9. b

10. b

11. c

12. d

13. a

14. d

15. d

16. b

17. d

18. c

19. a

20. b