DUMPSBOSS.

Linux Security

ECCouncil 212-77

Version Demo

Total Demo Questions: 7

Total Premium Questions: 77

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Introduction to Linux	2
Topic 2, Linux Installation	1
Topic 3, Linux Commands	12
Topic 4, Linux File System	5
Topic 5, Linux File and Directory Permissions	9
Topic 6, Linux User Management	11
Topic 7, Linux Network Configuration	9
Topic 8, Linux Security Administration	24
Topic 9, Linux Firewall	1
Topic 10, Linux Incident Response	3
Total	77

QUESTION NO: 1

Which of the following types of information is returned by typing ifconfig eth0?

(Choose two)

A. The names of programs that are using eth0

B. The IP address assigned to eth0

C. The hardware address of eth0

D. The hostname associated with eth0

ANSWER: B C

Explanation:

Running `ifconfig eth0` displays the configuration and status information for the network interface named `eth0`. This includes the IPv4 address configured on that interface, conventionally shown in the `inet` field. The address identifies the interface at the IP layer and is used for network communication on the attached network.

The command also displays the interface's hardware address, typically labeled `ether`. This is the MAC address used for Layer 2 Ethernet communication. Along with these key values, output commonly contains the subnet mask, broadcast address, IPv6 addresses when configured, interface flags such as `UP` and `RUNNING`, MTU, and packet/error counters. The exact fields can vary with the installed net-tools version and system configuration, but both the assigned IP address and hardware address are standard information reported for an Ethernet interface. Although modern Linux systems commonly use the `ip` command instead, `ifconfig` remains available on systems with the net-tools package installed. See the ifconfig manual page and the Linux network device documentation.

QUESTION NO: 2

Which of the following are appropriate practices when administering cron jobs that run with root privileges? (Choose two)

A. Use absolute paths in commands and scripts.

B. Ensure scripts used by root cron jobs are not writable by ordinary users.

C. Place root cron scripts in world-writable directories.

D. Redirect all cron error output to `/dev/null`.

ANSWER: A B

Explanation:

Root cron jobs should use absolute paths for commands and scripts so that execution does not depend on an unexpected `PATH` value. Scripts and directories used by privileged scheduled jobs should also be protected from modification by unprivileged users; otherwise, a user may be able to substitute or alter code that cron later executes as root.

It is also good practice to review cron output and logs, but redirecting errors to `/dev/null` removes potentially important evidence of failures or misuse. The crontab format and command execution behavior are described in the crontab(5) manual page.

QUESTION NO: 3

You're using a communications protocol that cannot handle encrypted passwords.

You need to disable the Shadow Password Suite. Which of the following pairs of commands restores the original clear-text passwords that you had encrypted into the

/etc/shadow and /etc/gshadow files?

A. grpunconv; passunconv

B. grpunconv; pwunconv

C. gconv; passunconv

D. gconv; pwunconv

ANSWER: B

Explanation:

`grpunconv; pwunconv` is the correct command pair for undoing shadow-password and shadow-group separation. The `grpunconv` utility takes the administrative password information held in `/etc/gshadow` and merges it back into `/etc/group`. The `pwunconv` utility then transfers password fields from `/etc/shadow` back into `/etc/passwd`, returning the account database to the traditional non-shadow layout required by older software or protocols that read password fields directly from `/etc/passwd`. Running the group conversion before the password conversion is a valid sequence because these utilities operate on their respective group and user account databases independently.

Strictly speaking, these utilities restore password *hashes*, rather than recovering users' original clear-text passwords; a properly stored password hash is not decrypted by this process. The practical outcome is that the password hashes are again present in the publicly readable account file, which is why this configuration is substantially less secure and should be used only when legacy compatibility makes it necessary. The shadow utilities also preserve file consistency through their normal account-file processing behavior. See the pwunconv manual page and the grpunconv manual page.

QUESTION NO: 4

You need to view only journal entries recorded since the current system boot. Which command should you use?

A. journalctl -b

B. journalctl -f

C. journalctl -k

D. journalctl --vacuum-time=1d

ANSWER: A

Explanation:

`journalctl -b` displays entries from the current boot. The `-b` option selects the boot that is currently running unless another boot offset or boot ID is supplied. This is useful when troubleshooting a service failure, startup problem, or suspected security event that occurred after the latest reboot.

The systemd journal can retain entries from previous boots when persistent journal storage is configured, so selecting the current boot helps limit the output to the relevant time period. See the journalctl documentation.

QUESTION NO: 5

Which of the following statements are true of Linux filesystem Access Control Lists (ACLs)? (Choose two)

A. They can grant access to a named user without changing file ownership.

B. They can define permissions for a named group.

C. They automatically bypass SELinux policy decisions.

D. They encrypt the contents of protected files.

ANSWER: A B

Explanation:

ACLs can grant permissions to a named user without changing the file owner or owning group. They can also provide permissions to a named group beyond the traditional owner, group, and other permission classes. The `setfacl` utility is used to modify ACL entries, while `getfacl` displays them.

An ACL does not override every other security mechanism; mandatory access controls such as SELinux can still deny an operation. ACLs also do not encrypt file contents. See the setfacl(1) manual page and the acl(5) manual page.

QUESTION NO: 6

You've installed a new application on a test computer. During testing, you've run into a number of problems that you are unable to troubleshoot. Where should you look first for help?

A. Before doing anything else, install the application on a production computer and see if you still have the same problems.

B. The log files associated with the application.

C. The Web site home page for the application.

D. Newsgroups where other users discuss the application.

ANSWER: B

Explanation:

The log files associated with the application are the best first source of help because they provide direct, time-specific diagnostic evidence from the failed application processes. Application logs commonly record startup failures, configuration parsing errors, missing dependencies, permission denials, database or network connection failures, stack traces, and relevant error codes. Reviewing these records lets an administrator identify what occurred at the time of the problem and gather precise details before escalating the issue or searching external support resources.

On Linux systems, an application may write logs to its own directory, to locations under `/var/log`, or to the systemd journal. The `journalctl` utility can be used to inspect journal entries for a service, including recent messages and logs from the current boot. This evidence-based troubleshooting approach is especially appropriate on a test computer because it supports safe diagnosis without introducing changes to production systems. After identifying the relevant error message, the administrator can use it to consult vendor documentation or seek targeted assistance if needed. See the systemd journalctl documentation and the Red Hat guidance on troubleshooting with log files.

QUESTION NO: 7

server/computer combination appears in both hosts.allow and hosts.deny. What's the result of this configuration when TCP Wrappers runs?

A. hosts.deny takes precedence; the client is denied access to the server.

B. hosts.allow takes precedence; the client is granted access to the server.

C. The system's administrator is paged to decide whether to allow access.

D. TCP Wrappers refuses to run and logs an error in /var/log/messages.

ANSWER: B

Explanation:

When a client/server combination matches entries in both files, **hosts.allow takes precedence; the client is granted access to the server.** TCP Wrappers evaluates its access-control tables in a defined order: it first checks

`/etc/hosts.allow`. If a matching rule is found there, access is allowed immediately and processing stops; the matching entry in `/etc/hosts.deny` is not considered. This ordering lets an administrator implement a default-deny policy in `hosts.deny`, such as denying all clients, while adding narrowly scoped exceptions for approved hosts or networks in `hosts.allow`. The result depends on a TCP Wrappers-enabled service being invoked through the wrapper mechanism or linked with libwrap, but for such a service the allow-table match is decisive. The tcp_wrappers manual documents the lookup sequence and states that access is granted when a match is found in the allow file before the deny file is consulted. See the hosts_access(5) manual page and the hosts.allow(5) manual page.