CompTIA Linux+

Certification Exam
Pre-draft Exam Objectives
Exam Number: XK0-006

- Pre-draft Exam Objectives summarize the tasks and skills identified in the Job Task Analysis (JTA) workshop that provide directional information about the upcoming exam version.
- The Draft Exam Objectives will replace the Pre-draft Exam Objectives after approximately two months when the skills have been peer-evaluated and validated through a JTA survey of job role practitioners.
- Pre-draft Exam Objectives may contain typos and errata that will be corrected during the development process.
- CompTIA will not accept feedback on the Pre-draft Exam Objectives document. If errors are found, please wait until the Draft Exam Objectives are posted, and then provide feedback using the Draft Exam Objectives Feedback form.

1.0 System Management

1.1 Explain basic Linux concepts

- Boot process
  - Boot loader
    - Configuration files
  - Kernel
    - Parameters
  - initrd
  - PXE
- Filesystem Hierarchy Standard
  - /
  - /boot
  - /etc
  - /var
  - /bin
  - /usr
  - /home
  - /tmp
  - /sbin
  - /lib
  - /dev
  - /proc
- Server architectures
  - aarch64
  - RISC-v
  - x86
  - x86_64/amd64
- Distributions
  - rpm-based
  - dpkg-based
- GUI
  - Display managers
  - Window managers
  - X Server
  - Wayland
- Software licensing
  - Open source software
  - Free software
  - Proprietary software
  - Copy left
- Storage types
  - File storage
  - Block storage
  - Object storage

1.2 Summarize Linux device management concepts and tools

- Kernel modules
  - modprobe
  - lsmod
  - insmod
  - rmmod
  - depmod
  - modinfo
- Device management
  - lsusb

    - lspci
    - lscpu
    - lshw
    - dmesg
    - dmidecode
    - lm_sensors
    - ipmitool
    - lsmem
- initrd management
    - dracut
    - mkinitrd
- Custom hardware
    - Embedded systems
    - GPU use cases
        - nvtop

1.3 Given a scenario, manage storage in a Linux system

- lvm
    - Logical volume
        - lvs
        - lvcreate
        - lvremove
        - lvresize / lvextend
        - lvdisplay
        - lvchange
    - Volume group
        - vgs
        - vgcreate
        - vgremove
        - vgextend
        - vgdisplay
        - vgexport
        - vgimport
        - vgscan
        - vgchange
    - Physical volume
        - pvs
        - pvcreate
        - pvremove
        - pvdisplay
        - pvscan
        - pvresize
        - pvmove
- Partitions
    - parted
    - fdisk / gdisk
    - lsblk
    - blkid
    - growpart
- Filesystems
    - Formats
        - xfs
        - ext4
        - btrfs
        - tmpfs
- Utilities

    - fsck
    - mkfs
    - xfs_repair
    - resize2fs
    - xfs_growfs
    - du
    - df
    - fio
- RAID
    - /proc/mdstat
    - mdadm
- Mounted storage
    - Mounting
        - mount
        - umount
        - /etc/fstab
        - /etc/mtab
        - /proc/mounts
        - Autofs
    - Mount options
        - rw
        - ro
        - noexec
        - nofail
        - nodev
        - nosuid
        - remount
        - noatime
        - nodiratime
    - Network mounts
        - Network file system (NFS)
        - SMB/Samba
- Inodes

1.4 Given a scenario, manage network services and configurations on a Linux server

- Network configuration
    - /etc/hosts
    - /etc/resolv.conf
    - /etc/nsswitch.conf
- NetworkManager
    - nmcli
    - nmconnect
- Netplan
    - netplan try
    - netplan apply
    - netplan status
    - Config files
        - /etc/netplan
- Common network tools
    - ping / ping6
    - traceroute
    - ss
    - nc
    - tcpdump
    - dig
    - hostname

    - ip
        - ip route
        - ip address
        - ip link
    - nmap
    - curl
    - nslookup
    - arp
    - mtr
    - ethtool
    - tracepath
    - iperf3

1.5 Given a scenario, manage a Linux system using common shell operations

- Common environmental variables
    - PATH
    - HOME
    - USER
    - SHELL
    - PS1
    - DISPLAY
- Paths
    - Absolute
        - ~
        - /
    - Relative
        - .
        - ..
        - -
- Shell environment configurations
    - .bashrc
    - .bash_profile
    - .profile
- Channel redirection
    - <
    - >
    - <<
    - >>
    - |
    - Standard output
    - Standard error
    - Standard input
    - Here docs
        - <<<
- Basic Shell Utilities
    - cat
    - tail
    - head
    - sed
    - awk
    - more
    - less
    - grep
    - tee
    - wc
    - cut

  - sort
  - uniq
  - tr
  - source
  - alias
  - bc
  - echo
  - printf
  - !
  - !!
  - history
  - xargs
  - uname
- Text editors
  - vi / vim
  - nano

1.6 Given a scenario, perform backup and restore operations for a Linux server

- Archiving
  - tar
  - cpio
- Compression tools
  - gzip
  - bzip2
  - 7zip
  - xz
  - unzip
- Other tools
  - zcat
  - zgrep
  - zless
  - ddrescue
  - rsync
  - dd

1.7 Summarize virtualization on Linux systems.

- Linux hypervisors
  - qemu
  - KVM
- Virtual machines
  - Paravirtualized drivers
  - VirtIO
  - Disk image operations
    - Convert
    - Resize
    - Image properties
  - VM states
  - Nested virtualization
- VM operations
  - Resources
    - Storage
    - RAM
    - CPU
    - Network
  - Baseline image templates
  - Cloning
  - Migrations

  - Snapshots
- Bare metal vs. virtual machines
- Network types
  - Bridged
  - NAT
  - Host-only/isolated
  - Routed
  - Open
- Virtual machine tools
  - `virsh`
  - `libvirt`
  - `virt-man`

2.0 Services and User Management

2.1 Given a scenario, manage files and directories on a Linux system

- Utilities
  - `ls`
  - `mv`
  - `cp`
  - `touch`
  - `file`
  - `stat`
  - `rm`
  - `ln`
  - `cd`
  - `locate`
  - `mkdir`
  - `rmdir`
  - `find`
  - `lsof`
  - `pwd`
  - `diff`
  - `sdiff`
- Links
  - Symbolic link
  - Hard link
- Device types in /dev
  - Block devices
  - Character devices
  - Special character devices

2.2 Given a scenario, perform local account management in a Linux environment

- Add
  - `useradd`
  - `adduser`
  - `groupadd`
- Delete
  - `userdel`
  - `deluser`
  - `groupdel`
- Modify
  - `usermod`
  - `chsh`
  - `passwd`
  - `groupmod`
- Lock

  - `passwd`
  - `chage`
  - `usermod`
- Expiration
  - Configuration files
  - `chage`
- List
  - `id`
  - `groups`
  - `whoami`
  - `w`
  - `who`
  - `last`
  - `lastlog`
  - `getent passwd`
- User profile templates
  - `/etc/skel`
  - `/etc/profile`
- Account files
  - `/etc/passwd`
  - `/etc/shadow`
  - `/etc/group`
- Attributes
  - UID
  - GID
  - EUID
  - EGID
- User accounts vs. system accounts vs. service accounts
  - UID range

2.3 Given a scenario, manage processes and jobs in a Linux environment

- Process verification
  - `ps`
  - `top`
  - `pstree`
  - `htop`
  - `atop`
  - `/proc/<PID>`
  - `pidstat`
  - `lsof`
  - `strace`
  - `mpstat`
- Process ID
  - PPID
  - PID
- Process states
  - Running
  - Blocked
  - Sleeping
  - Stopped
  - Zombie
- Priority
  - `nice`
  - `renice`
- Process limits
- Job and process management

    - `fg`
    - `bg`
    - `jobs`
    - `Ctrl + z`
    - `Ctrl + c`
    - `Ctrl + d`
    - `&`
    - `exec`
    - `nohup`
    - `pkill`
    - `kill`
    - `killall`
    - Signals
        - 1 HUP
        - 9 KILL
        - 15 TERM
- Scheduling
    - `crontab`
    - `at`
    - `anacron`

2.4 Given a scenario, configure and manage software in a Linux environment

- Installation, update and removal
    - Repository
    - Source
    - Package dependencies and conflicts
    - Package managers
    - Language-specific
        - pip
        - cargo
        - npm
- Repository management
    - Enabling/disabling
    - Third party
    - GPG signatures
- Package and repository exclusions
- Update alternatives
- Software configuration
- Sandboxed applications
- Basic configurations of common services
    - DNS
    - NTP/PTP
    - DHCP
    - HTTP
        - Apache HTTPD
        - Nginx
    - SMTP
    - IMAP4

2.5 Given a scenario, manage Linux using systemd.

- Systemd units
    - services
    - timers
    - mounts
    - targets
- Utilities
    - journalctl

  - hostnamectl
  - timedatectl
  - systemctl
  - systemd-analyze
  - resolvectl
  - systemd-resolved
  - systemd-blame
- Managing unit states
  - start
  - stop
  - restart
  - mask
  - unmask
  - enable
  - disable
  - reload
  - daemon-reload
  - edit
  - status

2.6 Given a scenario, manage applications in a container on a Linux server.

- Runtimes
  - runc
  - podman
  - containerd
  - docker
- Image operations
  - Pulling images
  - Build an image
    - Dockerfile
      (i) ENTRYPOINT
      (ii) CMD
      (iii) USER
      (iv) FROM
  - Pruning
  - Tags
  - Layers
- Container operations
  - Read container logs
  - Map container volumes
  - Start/stop containers
  - Inspect containers
  - Delete a container
  - Run
  - Exec
  - Pruning
  - Tags
  - Environmental variables
- Volume operations
  - Create volume
  - Mapping volume
  - Pruning
  - SELinux context
  - Overlay
- Container networks
  - Create network

    - Port mapping
    - Pruning
    - Types
        - macvlan
        - ipvlan
        - host
        - bridge
        - overlay
        - none
- Privileged vs. unprivileged

3.0 Security

3.1 Summarize authorization, authentication, and accounting methods

- polkit
- PAM
- SSSD/winbind
- realm
- LDAP
- Kerberos
- SAMBA
- Logging
    - journalctl
    - rsyslog
    - logrotate
    - /var/log
- System audit
    - audit rules
    - auditd

3.2 Given a scenario, configure and implement firewalls on a Linux system.

- firewalld
    - firewall-cmd
    - runtime vs. permanent
    - rich rules
    - zones
    - ports vs. services
- ufw
    - ports vs. services
- nftables
- iptables
- ipset
- netfilter module
- Address translation
    - NAT
    - PAT
    - DNAT
    - SNAT
- Stateful vs. stateless
- IP forwarding
    - net.ipv4.ip_forward

3.3 Given a scenario, apply OS hardening techniques on a Linux system.

- Privilege escalation
    - sudo
        - /etc/sudoers
            (i) NOEXEC

        (ii) NOPASSWD implications
      - /etc/sudoers.d
      - visudo
      - sudo -i
      - wheel group
      - sudo group
    - su -
- File attributes
    - lsattr
    - chattr
      - immutable
      - append only
- Permissions
    - File permissions
      - chmod
        (i) Octal
        (ii) Symbolic
      - chown
      - chgrp
    - Special permissions
      - stickybit
      - setuid
      - setgid
    - Default umask
- Access control
    - ACLs
      - setfacl
      - getfacl
    - SELinux
      - restorecon
      - semanage
      - chcon
      - ls -Z
      - getenforce
      - setenforce
      - getsebool
      - setsebool
      - audit2allow
      - sealer
      - States
        (i) Enforcing
        (ii) Permissive
        (iii) Disabled
- Secure remote access
    - SSHD
      - Key vs. password authentication
      - SSH tunneling
      - PermitRootLogin
      - Disabling X forwarding
      - AllowUsers
      - AllowGroups
    - SSH agent
    - SFTP
      - chroot
    - fail2ban
- Avoid the use of unsecure access services

    - Telnet
    - FTP
    - TFTP
- Disabling unused file systems
- Removal of unnecessary SUID permissions
- Secure boot
    - UEFI

3.4 Explain account hardening techniques and best practices.

- Passwords
    - Complexity
    - Length
    - Expiration
    - Reuse
    - History
- Multifactor authentication
- Checking existing breach lists
- Restricted shells
    - /sbin/nologin
    - /bin/rbash
- pam_tally2
- Avoid running as root

3.5 Explain cryptographic concepts and technologies in a Linux environment.

- Data at-rest
    - File encryption
        - gpg
    - File system encryption
        - LUKS2
        - argon2
- Data in-transit
    - OpenSSL
    - Wireguard
    - LibreSSL
    - TLS versions
- Hashing
    - SHA256
    - HMAC
- Removal of weak algorithms
- Certificate management
    - Trusted root certificates
        - No-cost
        - Commercial
    - Avoiding self-signed certificates

3.6 Explain the importance of compliance and audit procedures.

- Detection and response
    - Anti-malware
    - Indicators of Compromise (IOC)
- Vulnerability scanning
    - CVEs
    - CVSS
    - Backporting patches
    - Service misconfigurations
    - Tools
        - Port scanners
        - Protocol analyzer

- Standards and audit
  - OpenSCAP
  - Center for Internet Security (CIS) Benchmarks
- File integrity
  - aide
  - rkhunter
  - Signed package verification
  - Installed file verification
- Secure data destruction
  - shred
  - badblocks -w
  - dd if=/dev/urandom
  - cryptographic destruction
- Software supply chain
- Security banners
  - /etc/issue
  - /etc/issue.net
  - /etc/motd

4.0 Automation, Orchestration, and Scripting

4.1 Summarize the use cases and techniques of automation and orchestration in a Linux environment.

- Infrastructure as Code
  - Ansible
    - Playbooks
    - Inventory
    - Modules
    - Ad-hoc
    - Collections
    - Facts
    - Agentless
  - Puppet
    - Classes
    - Certificates
    - Modules
    - Facts
    - Agent/Agentless
  - OpenTofu
    - Provider
    - Resource
    - State
    - API
- Unattended deployment
  - Kickstart
  - Cloud-init
- CI/CD
  - Version control
  - Shift left testing
  - GitOps
  - Pipelines
  - DevSecOps
- Deployment orchestration
  - Kubernetes
    - Configmaps
    - Secrets
    - Pods

        - Deployments
        - Volumes
        - Services
        - Variables
    - Docker Swarm
        - Service
        - Nodes
        - Tasks
        - Networks
        - Scale
    - Docker/Podman Compose
        - Compose file
        - Up/down
        - Logs

4.2 Given a scenario, perform automated tasks using shell scripting.

- Expansion
    - Parameter expansion
        - ${var}
    - Command substitution
        - $(foo)
        - `foo`
    - Subshell
        - (foo)
- Functions
- IFS/OFS
- Conditionals
    - if
    - case
- Loops
    - until
    - for
    - while
- Interpreter directive
    - #!
- Comparisons
    - Numerical
        - -gt
        - -lt
        - -eq
        - -le
        - -ne
        - -ge
    - String
        - >
        - <
        - = =
        - =
        - = ~
        - ! =
        - <=
        - >=
- Regular expressions
    - [[$foo =~ regex]]
- Test
    - -f

    - -d
    - -z
    - -n
    - !
- Variables
    - Environmental
    - Arguments
    - Assignments
        - `local`
        - `set`
        - `unset`
        - `export`
        - `alias`
        - `unalias`
    - Return codes
        - `$?`

4.3 Summarize Python basics used for Linux system administration

- Setting up a virtual environment
- Built-in modules
- Installing dependencies
- Python fundamentals
    - Indentations
    - Current versions
    - Data types and structures
        - int
        - string
        - float
        - list
        - dict
        - bool
    - Extensible using modules and packages
- PEP 8 best practices

4.4 Given a scenario, implement version control using Git.

- `clone`
- `pull`
- `fetch`
- `commit`
- `add`
- `push`
- `checkout`
- `branch`
- `rebase`
- `config`
- `log`
- `merge`
    - `squash`
- `tag`
- `stash`
- `reset`
- `history`
- `diff`
- `.gitignore`

- `init`

4.5 Summarize best practices and responsible uses of AI

- Common use cases
  - Generation of code
  - Generation of regular expressions
  - Generation of infrastructure as code
  - Document code / create documentation
  - Recommendations for how to improve compliance
  - Security review
  - Code optimization
  - Code linting
- Best Practices
  - Avoid copy/paste without review/QA
  - Verify output
  - Data governance
    - Security of data
      (i) LLM training
      (ii) Human review
    - Local models, private vs. public AI
  - Adhere to corporate policy
- Prompt engineering

5.0 Troubleshooting

5.1 Summarize monitoring concepts and configurations in a Linux system.

- Service monitoring
  - Service-level agreement (SLA)
  - Service-level objective (SLO)
  - Service-level indicator (SLI)
- Data acquisition methods
  - SNMP
    - Traps
    - MIBs
  - Agent/agentless
  - Webhooks
  - Health checks
  - Log aggregation
- Configurations
  - Thresholds
  - Alerts
  - Events
  - Notifications
  - Logging

5.2 Given a scenario, analyze and troubleshoot hardware, storage, and Linux operating system issues

- Common issues
  - Kernel panic
  - Data corruption issues
  - Kernel corruption issues
  - Package dependency issues
  - File system will not mount
  - Server not booting
  - Can’t ping server
  - OS file system full
  - Server inaccessible
  - Device failure

- Inode exhaustion
- Partition not writeable
- Segmentation fault
- GRUB misconfiguration
- Killed processes
- PATH misconfiguration issues
- Systemd unit failures
- Missing or disabled drivers
- Unresponsive process
- Quota issues
- Memory leaks

5.3 Given a scenario, analyze and troubleshoot networking issues on a Linux system.

- Common issues
  - Misconfigured firewalls
  - DHCP issues
  - DNS issues
  - Interface misconfiguration
    - MTU mismatch
    - Bonding
    - MAC spoofing
    - Subnet
  - Routing issues
    - Gateway
  - Server unreachable
  - IP conflicts
  - Dual stack issues (IPv4 & IPv6)
  - Link down
  - Link negotiation issues

5.4 Given a scenario, analyze and troubleshoot security issues on a Linux system

- Common issues
  - SELinux
    - Policy issues
    - Context issues
    - Booleans issues
  - File and directory permission issues
    - ACLs
    - Attributes
  - Account access
  - Unpatched vulnerable systems
  - Exposed or misconfigured services
  - Remote access issues
  - Certificate issues
  - Misconfigured package repository
  - Use of obsolete and insecure protocols and ciphers
  - Cipher negotiation issues

5.5 Given a scenario, analyze and troubleshoot performance issues

- Common symptoms
  - Swapping
  - Out of memory
  - Slow application response
  - System unresponsiveness
  - High CPU usage
  - High load average
  - High context switching
  - High failed login attempts

- Slow boot
- High I/O wait
- Packet drops
- Jitter
- Random disconnects
- Random timeouts
- High latency
- Slow response times
- High disk latency
- Low throughput
- Blocked processes
- Hardware errors
- Sluggish terminal behavior
- Exceeding baselines
- Slow remote storage response
- CPU bottleneck