CS Exit Ethiopia

Complete Study Notes

Computer Science Exit Examination

1 selected course | 18 November 2026

Selected courses

01. Network and System Administration

https://csexitethiopia.com

abelasnake2003@gmail.com | Telegram: @Abechu7

Course contents

Each course begins on a new page. PDF bookmarks provide question, study-note section and revision-topic navigation.

Network and System Administration 6

How to use these study notes

START01 A practical study cycle

Start with a small set of questions before reading.

Mark the exact thing that stopped you: an unknown term, a rule, a calculation, a trace or a choice between alternatives.

Use the question lookup to open the relevant section.

Read one section, close it, and explain the rule in your own words.

Solve the question again and say why each other choice fails.

Then change one assumption or input and solve the variation.

This checks understanding beyond recognition.

A useful 60-minute session: 10 minutes of cold questions, 20 minutes of focused reading, 20 minutes of closed-book solving and 10 minutes updating an error log.

Adjust the timing to your needs.

Revisit mistakes after a gap, for example the next day, several days later and the following week.

These intervals are a practical starting plan, not a fixed learning law.

Record the section ID, your mistaken rule, the corrected rule and one boundary case.

Keep the log short enough to review.

Mix related courses during revision: programming with data structures; databases with security; networking with administration.

Explain how the same word changes meaning across contexts.

START02 What to emphasize

Emphasize definitions, key differences, algorithm costs, SDLC models, databases, networking and security.

Use short oral questions to check understanding and correct common misconceptions.

Combine those habits with the tracing, calculation, design and diagnosis demanded by the Ministry learning outcomes.

Allocate revision by both exam share and your error log.

A course with fewer items can still deserve more time if your foundations are weak.

The blueprint is a planning baseline, not a prediction of exact questions.

Definitions: distinguish nearby concepts, not just their names.

Calculations: write units, formulas, assumptions and intermediate steps.

Code and SQL: check validity before tracing the text; state any changes before analyzing a corrected version.

Algorithms and protocols: know the input, decision rule, state change, termination and failure cases.

Design choices: state the criterion.

A choice can change when cost, reliability, consistency or latency matters more.

Course group	Items in the 100-item blueprint	Revision implication
Web Programming	9	Reserve extra practice for HTML, CSS, JavaScript, PHP, client/server behavior and safe data handling.
Data Structures and Algorithms	7	Trace operations and compare their assumptions and costs.
Each of the other fourteen courses	6 each	Study all sixteen; do not abandon a whole course based on a short note or an alleged repeated-question list.

START03 How to handle exam traps

Underline NOT, FALSE, BEST and EXCEPT before comparing choices.

Check whether the stem supplies enough facts to select one answer.

A long scenario is not automatically an analysis question, and a familiar answer letter is not evidence.

When code is invalid, a diagram is absent, or two choices express the same rule, identify the clarification needed before answering.

Learn the correct rule and the smallest clarification needed.

Do not memorize an arbitrary key as a universal fact.

Distinguish worst case from average case, and an upper bound from a tight bound.

Check language version, SQL dialect, addressing assumptions, graph weights and tie-breaking rules.

For subnetting and scheduling, calculate rather than guess from a familiar number.

Use timed mixed papers only after you can explain your errors.

Revisit the reading sections behind recurring mistakes.

Network and System Administration

Complete Study Notes

NSA01 | Roles, principles and change discipline

System administrators maintain hosts, accounts, storage, software and availability; network administrators maintain connectivity, addressing, routes, network devices and performance.

Security administrators coordinate controls and policy.

These duties overlap, so account management, testing patches and training users are not exclusive to one job title.

Patience, communication, ethics, technical understanding and organization matter alongside command knowledge.

Use documentation, standardization, least privilege, automation and measured change.

Record the current state, intended change, rollback and validation.

Schedule maintenance according to service impact and urgency, not merely because a calendar date arrived.

Fault management restores failures; performance management measures service behavior; configuration management tracks settings; security management controls access; accounting tracks usage.

Data Communication and Computer Networking module

Network and System Administration module

NSA02 | Network foundations for administrators

Clients request services; servers provide them.

Peer-to-peer systems can do both.

A server can also be a client of DNS or backups, so dedicated role does not mean it never makes requests.

A network OS is not universally installed only on a dedicated server: clients also require networking support.

Specify centralized service software versus the OS on all participants.

Switching normally forwards frames using MAC addresses within a LAN; routing forwards IP packets between networks.

TCP supplies an ordered reliable byte stream, while UDP preserves datagrams without that guarantee.

Ports distinguish application endpoints, not physical connectors.

Draw switches, links, subnets, gateways, service hosts and dependencies in a network diagram.

A topology's shape alone does not guarantee security.

Data Communication and Computer Networking module

Network and System Administration module

NSA03 | Address, prefix, gateway and loopback

An IPv4 address is 32 bits; a CIDR prefix determines its network.

Compute network=address AND mask; devices use a route/gateway for remote subnets.

Private ranges are 10/8,172.16/12 and 192.168/16; private addresses are not directly Internet-routable.

Neither IP alone nor a class label determines today's subnet.

127.0.0.1 conventionally names the local IPv4 loopback interface; 127/8 is reserved for loopback.

Loopback tests local protocol processing, not cable, switch or remote reachability.

Programs can use it for local services.

Several loopback descriptions or several 127/8 addresses may be valid; compare each with the stated role.

Example

Worked microexample

- 192.168.2.70/26 has 64-address blocks: network .64, broadcast .127, ordinary host range .65-.126. A gateway must be reachable through an appropriate route; NAT does not repair a missing route.

Data Communication and Computer Networking module

Network and System Administration module

RFC 1122 host requirements

NSA04 | DHCP exchange and lease lifecycle

DHCP supplies addresses and configuration options.

Typical initial IPv4 allocation is Discover→Offer→Request→ACK.

The server manages its scope/pool, allocated leases, exclusions and reservations; options may include mask, gateway and DNS servers.

A dynamic lease lasts for a specified interval, not forever.

A client renews before expiry; it later tries rebinding if renewal fails.

A disconnected host does not automatically renew forever.

On lease expiry the address becomes eligible for reuse, subject to server policy.

Reconnection normally triggers validation/renewal/new allocation, so an expired lease alone does not prove persistent failure.

A reservation gives a predictable address to a known client while retaining DHCP configuration.

Example

Worked microexample

- If a disconnected client's lease expires while other clients obtain addresses, its old address may be reassigned. It must not simply resume using it without validating its configuration.

RFC 2131 DHCP

NSA05 | DHCP deployment, fallback and diagnosis

Use a scope matching the subnet, exclude manually assigned server/router addresses, set correct options and provide a relay when client broadcasts must cross a router.

Static assignments and DHCP can coexist; avoid address conflicts.

DHCP lowers repetitive administration but availability/security depend on deployment and protection from rogue replies.

It does not guarantee connectivity whenever a device changes location.

For a returning client, inspect current lease/address, reachable server/relay, free pool capacity and logs.

New devices can exhaust a pool, but absent observations this is only a hypothesis.

Some clients self-configure IPv4 link-local addresses such as 169.254.2.33 after DHCP failure; these support same-link communication, not ordinary routed Internet access. 0.0.0.0 is unspecified and 255.255.255.255 is limited broadcast.

Data Communication and Computer Networking module

Network and System Administration module

RFC 3927 IPv4 Link Local

RFC 2131 DHCP

NSA06 | DNS forward/reverse resolution and evidence

Forward resolution obtains an address from a name, using A for IPv4 and AAAA for IPv6.

Reverse lookup uses a PTR record in the reverse namespace to obtain a name from an address.

CNAME aliases a name, MX identifies mail exchangers and NS identifies authoritative name servers.

Forward and reverse records need not both exist or be perfect inverses.

Resolvers cache answers for their TTL.

Query nslookup name or specify the DNS server explicitly; compare the configured resolver with the authoritative answer when appropriate.

A successful ping by address but failed name lookup suggests naming/configuration trouble.

Ping by name may display a resolved address, but nslookup is the direct DNS diagnostic. “Resolving” is a general term that includes forward resolution.

Data Communication and Computer Networking module

Network and System Administration module

RFC 1035 DNS

Microsoft nslookup

NSA07 | Choose the command that exposes the required state

Windows ipconfig shows addresses, mask and gateway; ipconfig /all adds complete adapter configuration including physical/MAC address, DHCP and DNS information. /renew requests renewal, /release releases a dynamic address and /flushdns clears the client DNS cache.

Linux ip addr and ip route show interface addresses and routes; exact persistent configuration varies by distribution.

netstat displays connections/endpoints/protocols and TCP states; netstat -ano displays numeric addresses/ports and PIDs.

Use the exact command name netstat rather than netstart. ping probes IP reachability; traceroute/tracert reveals responding hops using TTL/hop-limit expiry.

ICMP filtering can prevent replies without proving the service is down. nslookup inspects DNS.

Data Communication and Computer Networking module

Network and System Administration module

Microsoft ipconfig

Microsoft netstat

Microsoft nslookup

NSA08 | A repeatable troubleshooting procedure

Define the symptom, scope, time and recent changes; gather the configuration, baseline and logs.

Form a testable hypothesis, test the least disruptive explanation, isolate the failing component, apply a targeted fix, verify the user service and document the result.

Change one relevant factor at a time when possible.

A reboot can hide evidence and is not a diagnosis by itself.

Use bottom-up checks from physical link, top-down checks from application errors, or divide-and-conquer by testing a middle layer.

Divide-and-conquer narrows the working/failing boundary; it is not random trial-and-error.

Rebooting, updates, resetting settings and scanning can all be troubleshooting actions in suitable circumstances; use fault evidence to choose among them.

Example

Worked microexample

- One client fails while others work: inspect that client's link, address and policy first. All clients fail name lookup but reach server IPs: inspect shared DNS/its configuration before replacing every cable.

Data Communication and Computer Networking module

Network and System Administration module

NSA09 | Connectivity diagnosis by layers

Check power, cable/connector and link state, then NIC/driver state, VLAN membership, address/prefix, local gateway, route, DNS and finally application port/authentication.

A recognized NIC makes a missing driver less likely, not proof of a working physical link.

Ethernet commonly uses RJ45 connectors; an RJ11 telephone jack is not the expected Ethernet connection.

If same-subnet traffic works but remote traffic fails, compare gateway/routes and filtering.

If ping fails but the application connection works, ICMP filtering is plausible.

If an application port is refused, the destination is reachable enough to reject it; inspect service/listener policy.

A timeout can result from loss, filtering or a down host and needs more evidence.

Example

Worked microexample

- After NIC replacement, recognized/enabled adapter but no link light: try a known-good Ethernet cable and switch port. Merely knowing that the OS recognizes the NIC cannot prove one particular cable is defective.

Data Communication and Computer Networking module

Network and System Administration module

NSA10 | Service catalog and remote access

DNS resolves names; DHCP supplies host configuration.

SMTP sends/relays mail, while IMAP/POP retrieve it.

FTP is a named file-transfer protocol; HTTP also transfers files, so “transfers files” is not an exclusive FTP description.

HTTP/HTTPS serve web resources.

SSH provides encrypted remote login; SFTP transfers files over SSH and differs from FTP or FTPS.

A VPN extends private connectivity over another network; secure tunneling commonly encrypts/authenticates traffic but depends on protocol/configuration.

RTP carries real-time media timing/sequence information, often over UDP; it does not guarantee timely delivery.

Network printers expose services shared by clients.

Provide accounts, permissions, configuration, monitoring and recovery for each service, not just its protocol name.

Data Communication and Computer Networking module

Network and System Administration module

NSA11 | Routing, NAT and port translation

Routing selects a next hop using a destination prefix; NAT rewrites addresses.

Many-to-one address-and-port translation (PAT/NAPT) maps internal flows to one public address using distinct port bindings.

For example, two private client sockets can appear as public-address ports 40001 and 40002, with replies translated back to the correct clients.

NAT is often used when private IPv4 clients access public networks; it does not itself replace a gateway, route, firewall or authentication.

One-to-one NAT and destination translation are other forms, so NAT does not always mean many hosts share one address.

Public addressing inside a LAN does not automatically require NAT.

Unsolicited inbound service access may require explicit mapping and filtering.

Data Communication and Computer Networking module

Network and System Administration module

NSA12 | Proxies, gateways and quality of service

A proxy acts on behalf of a client or backend service and can cache, filter or mediate application requests.

A default IP gateway is a next-hop router; an application gateway may translate protocols.

It need not be a modem/router combination.

These terms describe functions that can coexist on one machine.

QoS classifies, queues and prioritizes traffic to meet performance objectives such as low delay or limited loss.

It can operate across layers and support applications, so it is not exclusively an application-layer service.

Installing a new service is an administrative action; email configuration is another action and proxies/agents are components.

Distinguish service-management actions from components using a stated taxonomy.

Data Communication and Computer Networking module

Network and System Administration module

NSA13 | Workgroups, domains and directory services

A workgroup has locally managed accounts on individual machines; sharing may require local credentials and per-host configuration.

A domain centralizes identities and policies through directory infrastructure.

These are administrative arrangements, not physical topologies.

Avoid treating an old recommended workgroup size as a hard protocol limit.

HomeGroup is a historical Windows sharing feature, not another name for an AD domain.

A domain controller runs AD DS, authenticates domain accounts and participates in directory replication.

Redundant controllers improve service continuity only if dependencies and failure domains are considered; two virtual controllers on one failed physical host are not independent availability.

Provisioning requires correct addressing, DNS, time and supported OS/hardware; version-specific minimum sizes are not universal architecture rules.

Data Communication and Computer Networking module

Network and System Administration module

NSA14 | LDAP operations and secure directory access

LDAP accesses directory entries/attributes identified by distinguished names.

Bind authenticates a session; search retrieves matching entries; compare tests an attribute value; add/delete/modify change entries; modify-DN renames or moves; unbind ends the session.

StartTLS upgrades a connection to TLS where supported; LDAPS uses TLS from connection setup.

Protect credentials and verify server identity.

LDAP commonly uses TCP 389 and LDAPS TCP 636; an acronym/port alone does not make authentication secure.

AD also uses DNS and Kerberos; LDAP is not identical to the entire directory product.

Do not equate modern TLS-protected LDAP or LDAPS with retired LDAPv2.

Data Communication and Computer Networking module

Network and System Administration module

Microsoft, Enable LDAP over SSL

NSA15 | AD logical structure and delegation

A directory object represents a user, computer, group or resource and has schema-defined attributes.

Security principals have SIDs; display names are not the sole security identity.

A domain groups directory objects and a replicated domain database.

A tree groups domains in a contiguous DNS namespace; a forest groups trees with a shared schema/configuration and global-catalog framework.

OUs organize objects and provide convenient scopes for delegated administration and Group Policy; they are not separate forests/security boundaries.

Use groups to assign access and OUs to structure policy/delegation.

Schema changes require care because they propagate broadly; ordinary directory objects can be deleted, unlike the special restrictions on schema definitions.

Apply schema restrictions to the specific directory object type being discussed.

Data Communication and Computer Networking module

Network and System Administration module

NSA16 | AD sites, replication and trusts

Sites group IP subnets to model network locality, influence controller selection and manage replication traffic.

They are distinct from domains/OUs.

AD replicates directory changes among controllers; notification can trigger pulls within a site while schedules/link costs influence traffic between sites.

Replication is not an independent backup of previous good states.

If domain A trusts B, B's identities can be authenticated for resources in A, subject to permissions.

Trust alone grants no automatic file access.

Direction (one/two way) and transitivity (extends/does not extend) are independent.

An explicit trust is not necessarily one-way or nontransitive; the trust type/configuration determines that.

State the direction in words before drawing arrows.

Data Communication and Computer Networking module

Network and System Administration module

Microsoft AD trust concepts

NSA17 | Accounts, rights and password diagnosis

Create named accounts, assign only needed groups/rights, disable departed users and audit privileged activity.

Authentication verifies the account; authorization permits operations.

A user right such as remote logon differs from a permission on one file.

Group-based administration is easier to review than many hidden individual grants.

If only one domain user cannot change a password, inspect that account's cannot-change-password setting and relevant policy/permission before blaming all directory infrastructure.

Other possibilities include minimum password age, password rules or account state; the setting is a plausible exam cause, not a conclusion from failure alone.

Membership in Administrators does not inherently prohibit changing one's password.

Privileged reset and user-initiated password change are different operations.

Data Communication and Computer Networking module

Network and System Administration module

NSA18 | Windows permissions and effective access

Windows file/folder permissions include read, write, read/execute, modify and full control, plus more granular rights such as delete.

Inherited and explicit entries, group membership, deny entries and object scope affect effective access; use the effective-access view for a real account rather than adding labels informally.

Network access can be constrained by both share and filesystem permissions.

Deleting can be an explicit permission in Windows; specify the operating system before comparing permission names.

Modify and full control are not identical: full control includes changing permission/ownership capabilities.

Separate data access from logon rights.

A user allowed to read a file is not necessarily allowed to write it or administer its parent folder.

Data Communication and Computer Networking module

Network and System Administration module

NSA19 | Unix rwx and octal permissions

The nine ordinary permission bits are owner, group and other triplets.

In each triplet read=4,write=2,execute=1; sum enabled bits to get an octal digit.

A leading ls -l character denotes file type, not another permission bit.

For directories, read lists names, execute permits traversal/search, and write changes entries subject to other checks.

File deletion usually depends on parent directory permissions, with sticky-bit/ACL rules also possible.

Example

Worked microexample

- -rwxr--r-- means ordinary file, owner7/group4/other4, hence 744. 755 means rwxr-xr-x. A permission string must include all nine ordinary permission bits; missing hyphens cannot be inferred. Confirm the complete nine-bit permission string before converting it to octal.

Data Communication and Computer Networking module

Network and System Administration module

Linux man-pages chmod(2)

NSA20 | Linux paths, filesystems and checks

/ is the filesystem root; /root is the administrative user's home. /etc stores configuration, /home ordinary user homes, /var variable data/logs, /dev device nodes, /proc kernel/process information, /boot boot material and /tmp temporary files.

Cleanup/mount layouts vary; do not assume every partition has lost+found or every reboot erases every temporary directory.

ext2 is a Linux filesystem; ext4 is a later filesystem; HPFS is historically associated with OS/2, while swap is backing space rather than an ordinary user-file filesystem. fsck checks and optionally repairs filesystems through filesystem-specific tools; it is not init, halt or an inittab file.

Repairs require the filesystem's supported offline/safe procedure.

Mount attaches a filesystem to a directory; partitioning, mounting, quotas and capacity belong to storage administration.

Linux can access some foreign-origin filesystems, including HPFS when its driver is available.

Filesystem origin and Linux driver support are different questions.

Swap is a paging area rather than an ordinary user-file filesystem.

ext2 is a native Linux-filesystem example; broader Linux filesystem support can also include HPFS, so distinguish origin from driver support.

Data Communication and Computer Networking module

Network and System Administration module

util-linux fsck(8)

Linux kernel Read/Write HPFS documentation

NSA21 | Profiles, registry and persistent configuration

A local profile keeps settings on one client; a roaming profile synchronizes supported user settings/data through a server at logon/logoff.

Large profiles can slow these operations; redirected folders avoid repeatedly copying selected folders.

A central profile is backed up only if the server's backup policy actually includes it.

A cached profile is not a newly created independent local security account.

The Windows registry is a hierarchical configuration database with keys, subkeys and typed values.

HKLM describes machine settings and HKCU the current user view.

ACLs protect registry access.

Not all historical root handles apply to current versions.

Record/export the relevant configuration before an authorized change, and verify the intended service behavior afterward.

Data Communication and Computer Networking module

Network and System Administration module

NSA22 | Performance monitoring and bottleneck decisions

Measure a baseline, sample over an interval, compare with workload and correlate symptoms before changing hardware.

CPU utilization/run queues identify processing pressure; low free memory alone is weak evidence because cache may be reclaimable.

Persistent paging with high latency indicates memory pressure.

Storage capacity and I/O latency are separate: a disk can be nearly empty but saturated, or nearly full while lightly loaded.

Measure interface bytes/second, errors, drops, packet delay/loss and utilization against link rate.

Distinguish capacity (maximum available rate) from observed throughput.

Logs can reveal both security incidents and performance faults.

Performance counters expose measured events; system calls/APIs and log parsers are ways scripts can collect them.

Example

Worked microexample

- Transferring 80 MB in 10 s gives 8 MB/s≈64 Mb/s before overhead. A 95%-busy CPU with a growing runnable queue suggests CPU pressure; a 95%-full disk requires capacity action even if its latency is low.

Data Communication and Computer Networking module

Network and System Administration module

NSA23 | Monitoring, capture and assessment tools

Task Manager shows running processes/resource use; Resource Monitor/Performance Monitor give more detailed counters.

Wireshark captures and dissects packets.

Nmap discovers hosts/ports/services.

Nessus assesses vulnerabilities/configuration.

Snort can detect suspicious traffic and alert, and can prevent traffic in suitable inline deployments; it also has sniffing/logging roles.

Thus packet-capture capability is not exclusive to one product, although Wireshark is the conventional analyzer choice.

An IDS detects/alerts; an IPS can block inline.

Scanning identifies potential weaknesses rather than proving successful exploitation.

A captured packet, open port or alert needs context.

Monitoring network usage is not the same as configuring access control or encrypting communications.

Data Communication and Computer Networking module

Network and System Administration module

Wireshark User Guide, Introduction

Nmap Reference Guide

Snort project

Tenable Nessus User Guide

NSA24 | Backups, restore and continuity

Backups recover previous data after deletion, compromise, corruption, failed upgrades/migrations or hardware loss.

Authorization controls current access, a different function.

A mirror/replica also copies deletions or corruption and does not replace versioned independent backups.

Keep protected copies in separate failure domains, define retention and test restores.

A full backup copies the chosen dataset; incremental copies changes since the last backup; differential copies changes since the last full backup.

Restoring incrementals generally needs the full plus each subsequent incremental, whereas differential needs the full plus latest differential.

RPO is tolerable lost-data time and RTO tolerable restoration time.

Compression reduces storage, scheduling runs the operation and restore testing establishes whether recovery actually works.

Data Communication and Computer Networking module

Network and System Administration module

NSA25 | RAID layouts, capacity and selection

RAID0 stripes for capacity/throughput with no disk-failure tolerance.

RAID1 mirrors; a two-disk mirror has one disk's usable capacity and survives one disk failure.

RAID2 uses bit striping/Hamming-code redundancy, RAID3 byte striping with dedicated parity, RAID4 block striping with dedicated parity and RAID5 block striping with distributed parity.

RAID6 distributes dual parity and tolerates two disk failures.

RAID10 stripes mirrored pairs; survival depends on which disks fail.

With N equal disks of capacity D, RAID0 usable=ND, paired RAID1/10≈ND/2, RAID5=(N−1)D and RAID6=(N−2)D. Hardware RAID may offload work to controller resources/cache but does not universally outperform software RAID.

Rebuilds consume bandwidth; parity writes have overhead.

A large university/bank alone does not select a level.

Striping plus parity is not uniquely RAID5 unless distributed block parity is specified.

Example

Worked microexample

- Four 2 TB disks give 8 TB RAID0, 4 TB RAID10, 6 TB RAID5 or4 TB RAID6, ignoring metadata. Choose using failure tolerance, workload, capacity and recovery requirements; none of these layouts replaces backups.

Data Communication and Computer Networking module

Network and System Administration module

Intel server storage options

NSA26 | Security policy development and purpose

A security policy states required behavior, scope, accountable roles and enforcement.

Begin with assets/service objectives, assess threats and weaknesses, choose controls, define exceptions/incident procedures, approve/communicate, monitor and periodically review.

CIA means confidentiality, integrity and availability; neither sensitive-data protection nor availability alone is the universal primary goal.

A policy can require password controls, approved encryption, restricted administrative access, backups and patch deadlines; firewall rules implement part of it.

Open unrestricted wireless access conflicts with controlled internal access unless intentionally isolated as a guest service. “Use the system passively” is not a purpose of security policy.

Policies coordinate action and reduce risk; they do not guarantee every attack will fail.

Data Communication and Computer Networking module

Network and System Administration module

NSA27 | Hardening, controls and vulnerabilities

Hardening reduces exposure through unnecessary-service removal, supported patches, restricted accounts/permissions and appropriate port filtering.

Blocking an unused known port can be hardening; blocking every useful port merely destroys availability.

Weak passwords and unpatched exploitable software are vulnerabilities; attacks exploit weaknesses, and risk combines likelihood with impact.

Preventive controls restrict harmful actions, detective controls expose them and recovery controls restore service.

Account lifecycle, backup procedures, device configuration and traffic analysis can all be security controls under a stated objective.

A vulnerability scan checks known weakness patterns, not bandwidth.

Test/stage patches, prepare rollback, deploy and verify; testing and installation are both legitimate administrator duties.

Data Communication and Computer Networking module

Network and System Administration module

NSA28 | Encryption, hashes and script secrets

Encryption provides confidentiality with correct key/nonce use; authenticated encryption adds integrity/authenticity.

Unspecified encryption alone does not guarantee both properties.

A password hash supports verification without recovering the original password; it is not a reversible credential store.

Encryption can protect a recoverable credential only if its key is separately protected.

Keep reusable secrets out of source code and routine logs; obtain them from an appropriately protected credential service or restricted runtime source.

Environment variables and configuration files are transport/storage choices, not automatic encryption.

A script that must authenticate to another service often needs the actual credential/token, so replacing it with a one-way hash will not work.

State whether the question is about password verification or recoverable service credentials.

Data Communication and Computer Networking module

Network and System Administration module

RFC 5116 Authenticated Encryption

NSA29 | Incidents, evidence and reports

Incident response prepares, detects/analyzes, contains, removes the cause, restores service and learns from the event.

Investigation is reactive response; routine patching and backup are normally preventive/preparatory operations.

Document timestamps, affected systems, evidence, impact, actions, responsible people and recovery results.

An incident report may record policy violations, but its scope is broader than that alone.

DoS harms service availability; DDoS uses distributed sources.

Phishing deceives users, brute force tries credentials and an interception attack reads/alters communications.

Correlate logs using time and identity/context.

Preserve relevant evidence before resetting systems.

A suspicious event is not automatically confirmed compromise; record facts and uncertainty.

Data Communication and Computer Networking module

Network and System Administration module

NSA30 | Segmentation, NAC and remote administration

Segmentation separates groups of systems and enforces allowed flows with routing/filtering policies; VLAN membership alone does not establish complete security.

NAC assesses device/user admission and can reject or quarantine unauthorized/noncompliant devices.

Authentication proves an identity, while access rules decide which resources it may use.

Use authenticated protected remote administration such as SSH or properly secured management services, with restricted source access and auditing.

VPNs provide protected connectivity under their defined security model; they are not packet analyzers.

Firewall filtering permits/blocks traffic according to policy; it is not inherently encryption.

Losing remote connectivity while changing a management route/firewall is an operational risk to plan for.

Data Communication and Computer Networking module

Network and System Administration module

Cisco, Network Access Control

NSA31 | Scripting: inputs, control flow and reusable work

Scripts automate repeatable administration; they are not inherently faster than compiled programs or automatically secure.

Python, shells and PowerShell are common choices; HTML is markup.

Define inputs and outputs, use variables for state, if/elif/else for decisions, loops for repeated work and functions for reusable operations.

Python indentation defines blocks; == compares while = assigns.

Make reruns predictable: compute the desired state, compare with current state and change only what differs.

Validate a small dry run before mutation.

Avoid requiring privilege for read-only reporting.

A good script reports failures and has a meaningful exit status rather than printing success unconditionally.

Example

Worked microexample

- def warn_percent(values, limit): for name, percent in values: if percent >= limit: print(name, "high", percent) warn_percent([("root", 92), ("data", 40)], 90) # Output: root high 92 (with spaces from print). Only the first pair meets the threshold.

Python tutorial, control flow

NSA32 | Launching commands with Python subprocess

Python subprocess is a module, not a universal shell command. run launches a child program and returns its result; capture_output collects stdout/stderr, text=True decodes text, timeout bounds waiting and check=True raises for nonzero exit.

Pass an argument list to avoid accidental shell interpretation; quote handling differs from composing a shell string.

Echo prints, grep searches text and ipconfig/ifconfig reports/configures networking; none is the general Python child-process abstraction.

Specify the scripting language before selecting subprocess for process execution.

Example

Worked microexample

- import subprocess result = subprocess.run(["ip", "addr", "show"], text=True, capture_output=True, check=True, timeout=5) print(result.stdout) # Linux with iproute2 installed. Missing program raises OSError; # nonzero status raises CalledProcessError; timeout raises TimeoutExpired.

Python subprocess documentation

NSA33 | File input, configuration parsing and updates

File I/O obtains/persists configuration; string operations parse values; regex can recognize patterns.

File I/O, string operations and regex can all help manipulate configuration; choose according to the task.

Use the format's parser for structured JSON/INI rather than blindly replacing every matching substring.

Open text with an explicit encoding and use with so files close even when processing fails.

For changes, validate the desired output, preserve a known-good copy, write to a temporary file in the same filesystem, then atomically replace the target where supported.

Check permissions and configuration syntax, reload the affected service, verify and retain a rollback.

Merely appending the same setting on every run is not idempotent.

Example

Worked microexample

- import json with open("settings.json", encoding="utf-8") as handle: config = json.load(handle) port = config["port"] if not isinstance(port, int) or not 1 <= port <= 65535: raise ValueError("port must be 1..65535") print("validated port", port) # This example validates/reports and does not alter a system file.

Python tutorial, input/output

NSA34 | Arguments and targeted error handling

Command-line parsing converts user inputs into named, typed values and displays usage.

In Python argparse, positional arguments supply required inputs and flags enable options.

Validate ranges and paths before work; do not let unchecked input become executable shell text.

Example

Worked microexample

- import argparse from pathlib import Path p = argparse.ArgumentParser() p.add_argument("logfile", type=Path) p.add_argument("--limit", type=int, default=10) a = p.parse_args() if a.limit <1: p.error("limit must be positive") try: print(a.logfile.read_text(encoding="utf-8")[:a.limit]) except (OSError, UnicodeError) as exc: p.exit(1, f"read failed: {exc}\n") # Input limit controls characters here. Exceptions skip the rest of try; # catching specific failures makes unexpected programming errors visible.

Python argparse documentation

Python errors and exceptions tutorial

NSA35 | Regex log analysis and rotation

Regex recognizes textual patterns; it does not encrypt, compress or rotate logs. search finds a match anywhere, while fullmatch requires the entire string to fit.

Use raw Python strings for backslash-heavy patterns, capture only the needed fields and treat unmatched/malformed lines explicitly.

Log rotation bounds file size and preserves historical segments; it is complementary to parsing.

Count the event actually requested, use timestamps/hosts to correlate, and separate absence of matches from failure to read the log.

Logs support security and performance analysis, so neither purpose is universally exclusive.

Example

Worked microexample

- import re pattern = re.compile(r"status=(\d{3})\b") counts = {} for line in ["status=200 ok", "status=503 unavailable", "noise"]: match = pattern.search(line) if match: code = match.group(1) counts[code] = counts.get(code,0)+1 print(counts) # {"200":1,"503":1}; noise is deliberately ignored.

Python re documentation

NSA36 | Collecting CPU and memory measurements

A cumulative CPU counter must be sampled twice to calculate interval use; a single lifetime total is not current utilization.

On Linux /proc/stat exposes cumulative CPU categories; define which categories count as idle and avoid double-counting guest fields.

A simple convention is busy fraction=1−Δidle/Δtotal using idle+iowait as the chosen nonbusy categories.

Iowait interpretation has limitations.

/proc/meminfo exposes MemAvailable, an estimate of memory available without swapping; MemFree alone excludes reclaimable memory.

Windows Performance Counters and OS APIs expose analogous measurements through different interfaces.

System calls, counters and log parsing can all contribute; selecting one as the only monitoring technique is unjustified without a platform/API.

Example

Worked microexample

- If total CPU counters grow by 1000 ticks and chosen idle counters by 250, measured busy≈75%. If MemTotal=8192 MiB and MemAvailable=2048 MiB, approximate unavailable/used fraction is75%, not a prediction that an allocation of exactly 2048 MiB must succeed.

Data Communication and Computer Networking module

Network and System Administration module

Linux proc_stat(5)

Linux proc_meminfo(5)

NSA37 | Scheduling a backup and checking restoration

Schedule recurring work using cron/systemd timers on suitable Unix systems or Task Scheduler on Windows.

A user crontab uses minute,hour,day-of-month,month,day-of-week then command.

Use absolute paths, known environment, protected inputs, logs, bounded runtime and a lock when overlap is unsafe.

Do not assume an interactive PATH, current directory or stored credentials exist in a scheduled job.

Compression packages data; scheduling chooses when the operation runs.

Test exit status and the archive contents, then restore a sample into an isolated destination.

A successful command alone does not prove a usable backup.

Version control of scripts helps review but does not replace data backup.

Example

Worked microexample

- One user-crontab entry (keep the command on one line): 0 2 * * * /usr/bin/python3 /opt/backup.py /srv /bak >>/var/log/bak.log 2>&1 This runs daily at 02:00 in the scheduler's timezone as the crontab owner. The script takes its configured source and destination arguments. Paths, permissions, logging and the backup program must already be configured. Standard user crontabs omit a username field; system crontabs may include one.

Data Communication and Computer Networking module

Network and System Administration module

crontab(5) manual

NSA38 | Construct a bounded backup script

Separate argument parsing, preconditions, copying/archive creation and result reporting.

Require a source directory, avoid putting the backup inside that source, create unique output names and ensure a failed partial file is not mistaken for a completed backup.

Real backup policies also need application consistency, capacity checks, retention and independent protected copies.

The following teaching script packages a directory; it does not silently install a scheduled job.

Example

Worked microexample

- import argparse, shutil from pathlib import Path from datetime import datetime, timezone p = argparse.ArgumentParser() p.add_argument("source", type=Path) p.add_argument("destination", type=Path) a = p.parse_args() src, dst = a.source.resolve(), a.destination.resolve() if not src.is_dir() or not dst.is_dir(): p.error("source and destination must be existing directories") if dst == src or src in dst.parents: p.error("destination must be outside source") name = datetime.now(timezone.utc).strftime("copy-%Y%m%dT%H%M%S%fZ") base = dst /name try: archive = shutil.make_archive(str(base), "gztar", root_dir=src) except OSError as exc: base.with_suffix(".tar.gz").unlink(missing_ok=True) p.exit(1, f"backup failed: {exc}\n") print(archive) # Run against a teaching directory, verify archive members, then restore # into a different empty directory and compare selected files.

Python tutorial, control flow

Python tutorial, input/output

Python errors and exceptions tutorial

Python shutil documentation

NSA-S01 | Operations principles versus development and delivery models

Least privilege limits an account or process to the rights its task needs. Agile development instead organizes frequent delivery and adaptation of software. Object-oriented programming organizes software around objects with state and behavior. Cloud computing supplies on-demand access to a shared pool of configurable computing resources. A development approach or hosting model does not itself define an access-rights principle.

Network and System Administration module · page 43

Agile Manifesto - Principles

Oracle - Object-oriented programming concepts: objects

NIST SP 800-145 - The NIST Definition of Cloud Computing

NSA-S02 | Disaster recovery as a planning objective

Disaster recovery planning prepares for restoring systems and data after disruptive loss. Regular protected backups and tested restoration are part of that plan. Continuing operation despite a component failure is a fault-tolerance objective; possessing extra components alone does not establish that older valid data can be recovered.

Network and System Administration module · page 30

NIST SP 800-34 Revision 1 - Contingency planning

NSA-S03 | Initialization, shutdown and filesystem-check tools

In a traditional SysV-style system, init starts and supervises configured processes; /etc/inittab is its configuration file. The halt command stops the system through the supported shutdown procedure. Neither initialization, shutdown nor their configuration file is a filesystem-consistency check; fsck supplies that separate checking and optional repair role.

Debian sysvinit manual - init

Debian sysvinit manual - halt

NSA-S04 | What an ICMP reachability probe actually tests

Ping uses ICMP echo messages to test IP reachability. A reply shows that the addressed host and path support that exchange; no reply alone does not prove the host is disconnected, because filtering or loss can suppress responses. Traceroute instead discovers responding intermediate hops through packet-lifetime expiry; it is a path-discovery diagnostic, not the same question as an end-to-end echo test.

Network and System Administration module · page 27

NSA-S05 | Distinguish scripting features by their responsibility

Exception handling transfers control from a failing operation to a matching handler. A try block contains the attempted work; an except clause handles specified raised exceptions. A loop repeats operations, a conditional chooses an ordinary branch, and a function packages reusable work; these constructs do not by themselves intercept exceptions.

A module groups definitions and statements that can be imported. A class bundles data with functionality for its objects. These organize a program, while command-line parsing interprets the arguments and flags supplied at invocation. Their implementation can use functions or classes, but organization alone does not parse arguments.

A signal handler responds when a process receives a specified asynchronous event notification. Handling such a notification is distinct from reading CPU or memory measurements through an OS interface, a performance counter or a log that actually records the metric.

Python tutorial - Errors and exceptions

Python tutorial - Modules

Python tutorial - Classes

Python library - Signal handlers for asynchronous events