



Identity and Access Management

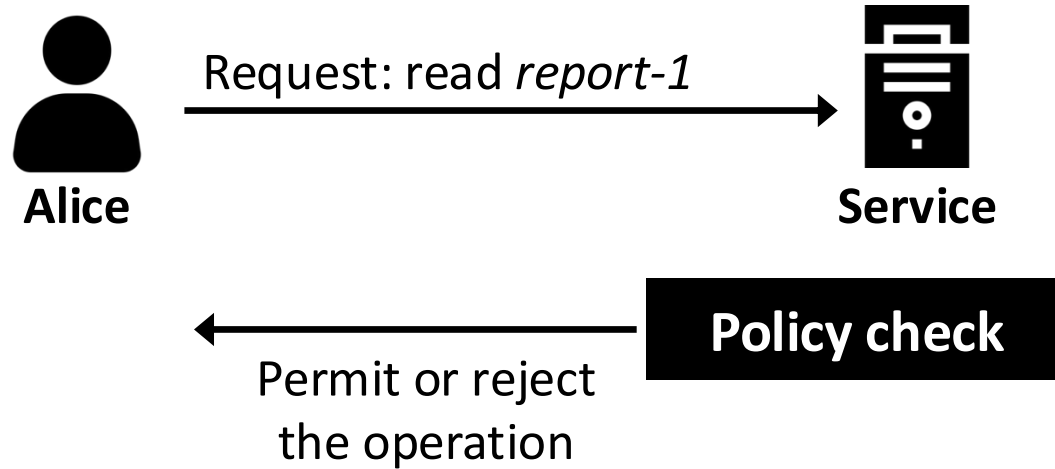
Cloud Security

CSE4059

Access Control

- Access control determines who or what can access a resource and what operations they are allowed to perform
 - It enforces rules about whether a principal may perform an action on a resource
- Principal
 - The entity whose access is being evaluated, such as a user or a running program
- Permission
 - Allows a specific action on a specific resource
- Access decisions are made by evaluating access requests against access-control policy

Access Request



- An access request identifies
 - **Principal**: who or what is making the request
 - **Action**: what operation is requested
 - **Resource**: what the operation targets
- Context may provide additional information
 - Such as request time or authentication strength
- The request is evaluated against access-control policy before the operation is performed

Identity and Credentials

- Identity
 - A representation of an entity within a system
 - Ex) Alice's account
- Identifier
 - A name or value used to refer to an identity
 - Ex) a username
- Credential
 - Evidence used to prove control of an identity
 - Ex) a password
- Example
 - Alice changes her password, but her account and permissions remain the same
 - A credential can change while the underlying identity and its permissions remain the same



Alice (*identity*)

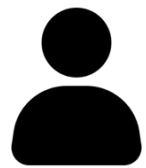
Account: alice1 (*identifier*)

Permissions: Read *report-1*

Password:
abcd (*credential*) → **Password:**
efgh (*credential*)

Authentication

- Authentication verifies evidence supporting a claimed identity
 - TLS also performs authentication – typically, the client authenticates the server
 - Here, we focus on authenticating users and software



Alice



Verifier

Identity claim: *I'm Alice*
Evidence: *credential*



Authenticated identity,
Session established



- The verifier checks Alice's credential to determine whether the claim should be accepted
 - If successful, the system establishes an authenticated session associated with Alice's account
- Authentication establishes the identity represented by the requester
 - Does not authorize a specific resource operation

Authorization

- Authorization decides whether a principal may perform an action on a resource
 - Authentication = who are you?
 - Authorization = what are you allowed to do?
- The same authenticated identity may receive different authorization decisions for different actions
- Example: Alice is already authenticated, and both requests target the same resource (e.g., report-1)

Identity	Requested action	Authorization result
Alice	Read	Allow
Alice	Delete	Deny

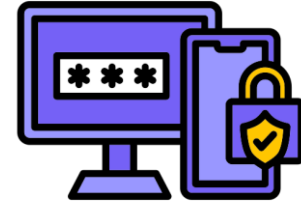
Authentication Factors

- An authentication factor is a category of evidence used to verify an identity
- Knowledge
 - Something you know
 - ex) a password
- Possession
 - Something you control
 - ex) a security key or authenticator app
- Inherence
 - Something you are
 - ex) a fingerprint
- A username is an identifier, not an authentication factor
- Two passwords are still one factor category: knowledge.

Multi-Factor Authentication

- Multi-factor authentication (MFA) requires evidence from at least two distinct factor categories

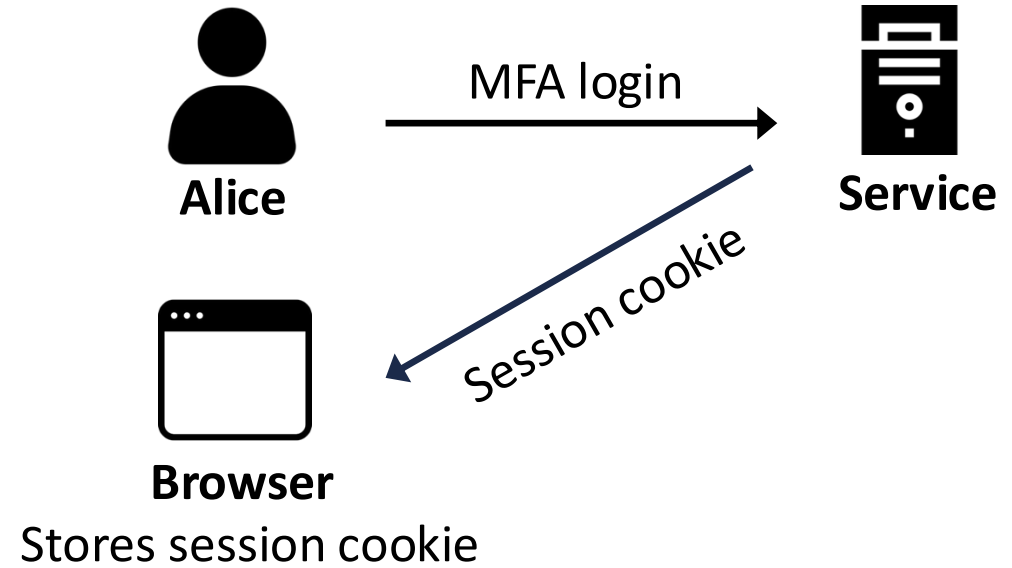
- Example: a password + a code from an authenticator app



- A stolen password alone is not enough if the attacker cannot provide the second factor
- MFA **strengthens** authentication
 - It does not grant additional permissions
- After login, session credentials must also be protected

Session Theft

- Session theft occurs when an attacker steals and reuses a session credential to act as an authenticated user
- Example: stealing a session cookie
- Alice completes MFA, and the service issues a session cookie to her browser
 - Session cookie? a value the browser stores and sends with later requests to represent an authenticated session



Session Theft (cont'd)

- Malware on Alice's computer steals the cookie and reuses it to access the service
 - Assume malware can read cookie and session is active
- If the service accepts the stolen cookie, the attacker can read *report-1* without repeating Alice's MFA
- The attacker can use only the permissions available to that session
- TLS protects network traffic
 - But it does not protect session data already exposed on a compromised endpoint



Least Privilege

- Least privilege grants only the access needed for an assigned task
- Permissions should be limited by both action and resource
 - ex) a report reader may read *report-1*, but does not need permission to delete reports or modify access control
- Different tasks may require different permission sets
 - ex) a publisher may need additional permission to publish reports
- If a credential is stolen, least privilege limits what the attacker can do
 - The attacker can still perform operations that were legitimately permitted

Permission Matrix

- A permission matrix shows which permissions each job requires
 - Each row: a job or responsibility / Each column: a permission
- This helps define task-specific permissions before assigning them to users through roles
- When many users perform the same job, roles simplify permission management

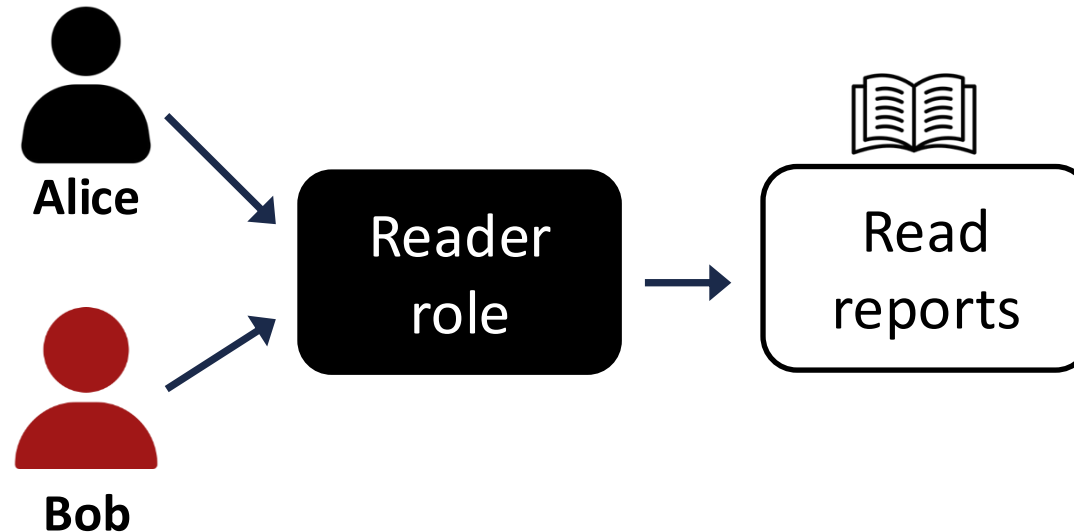
Job	Read reports	Publish reports	Read audit records	Delete inputs
Reader	O	—	—	—
Publisher	O	O	—	—
Auditor	—	-	O	—

O: permission is granted / — : no grant

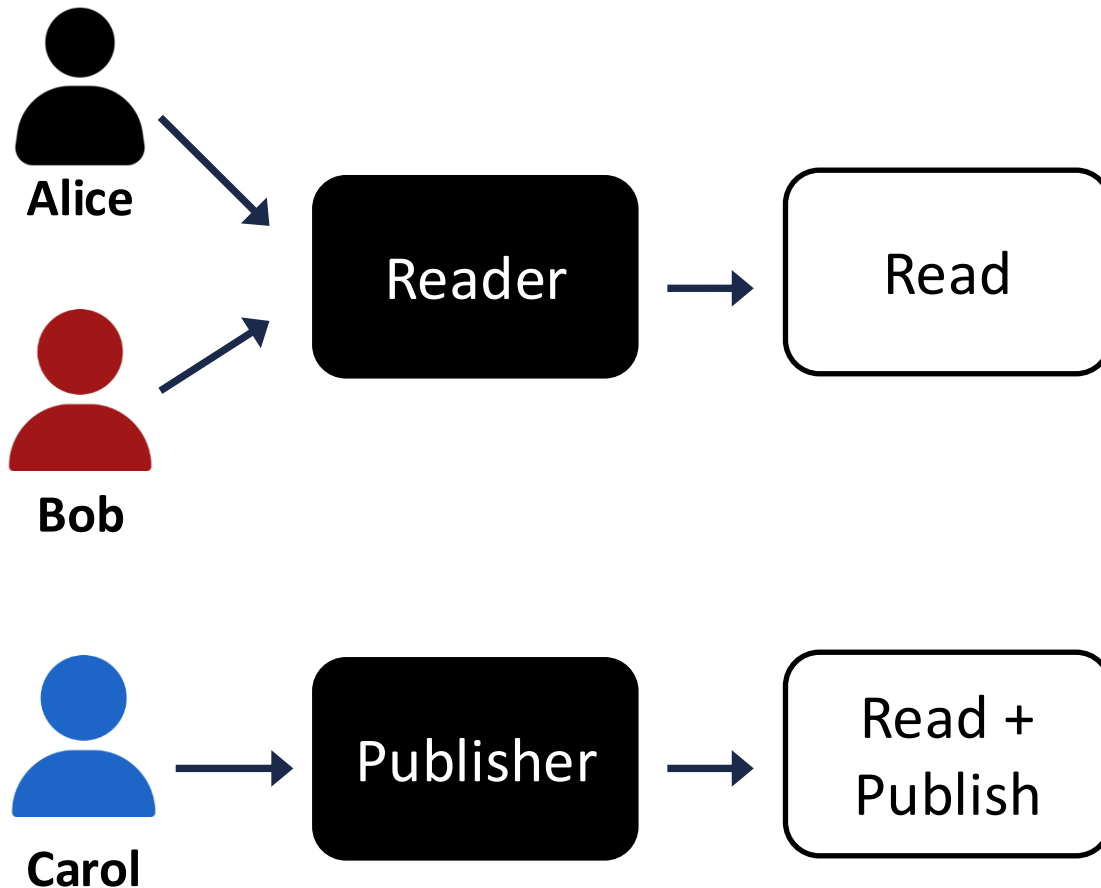
Role-Based Access Control

- Role-Based Access Control (RBAC) assigns permissions to roles and assigns users to those roles
- A role groups permissions needed for a responsibility
 - ex) the Reader role includes permission to read reports
- Changing the role's permissions changes what all users assigned to that role can do

Alice and Bob can share the Reader role while keeping separate identities and credentials



Role Assignment



- Reader permits report reads
 - Alice and Bob are assigned the Reader role
- Publisher permits report reads and publication
 - Carol is assigned the Publisher role
- For the same publish request, Alice is denied while Carol is allowed
- Changing a role assignment changes a user's access

Privileged Access

- Privileged access is access that can change security controls, identities, permissions, or broad sets of resources
 - Ex) an administrator who changes role assignments can change other users' access
- Routine work should be separated from administrative work
 - Grant elevated access only for the required task and, where practical, for a limited time
- Temporary privileged access can still make lasting changes
 - Expiring the session does not undo policy or configuration changes already made

Human and Workload Identities

- In cloud systems, software also accesses resources
 - So far, our examples used human users
- A workload is software that performs a task, often without a human actively signed in
 - ex) scheduled jobs, backend services, batch processes
- Human and workload identities should be separate
 - Alice and a scheduled reader may need the same permission, but they are different principals

Entity	Example	Required access
Human 	Alice	Read reports
Workload 	Scheduled reader	Read reports

Human and Workload Identities (cont'd)

- Like a human user, a workload also needs credentials to authenticate when accessing resources



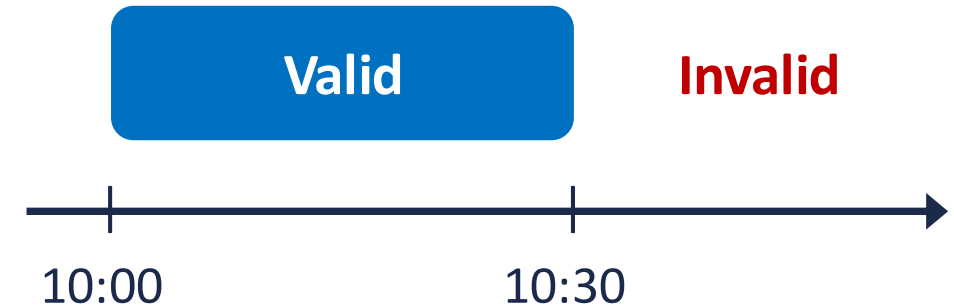
- A workload should not use a human user's personal credential
 - Embedding Alice's credential makes the job act as Alice and couples the job to her credential lifecycle
- Give the workload its own identity so its permissions and credentials can be managed independently

Long-Lived Credentials

- A **long-lived** credential remains valid for a long period
 - Until it is disabled, rotated, or expires
- Embedding a credential in application code can spread copies into repositories, builds, and deployments
 - Removing the credential from the code does not invalidate copies that were already leaked
- If a credential is exposed, disable or rotate it
- A safer design is to let authorized workloads obtain **short-lived** credentials when needed

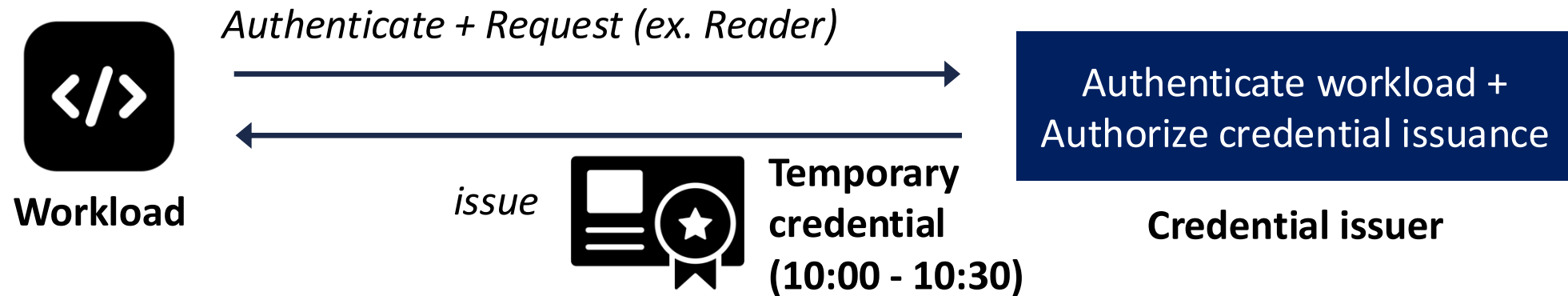
Temporary Credentials

- Temporary credentials are authentication material valid for a limited lifetime
- An authorized workload obtains them when needed
 - Instead of storing a permanent secret in its code
- Short-lived does not mean low-privilege
 - Permission scope and credential lifetime are separate
- The workload must still authenticate before credentials can be issued



ex) This credential is valid
from 10:00 to 10:30

Credential Issuance



- The workload authenticates to the credential issuer and requests temporary access
- The issuer checks whether the workload is authorized for the Reader role
- If approved, the issuer issues temporary credentials with a limited lifetime
- The report service later authorizes each resource request separately

Using Temporary Credentials

Time	Action	Credential	Permission	Result
10:10	Read	Valid	Permitted	Accept
10:10	Delete	Valid	Not permitted	Reject (permission)
10:31	Read	Expired	Permitted	Reject (expired)

- Assume the same workload, Reader role, and *report-1* in all rows
- A request succeeds only if the credential is valid and the requested action is permitted
- Credential validity and authorization are separate checks

Credential Compromise

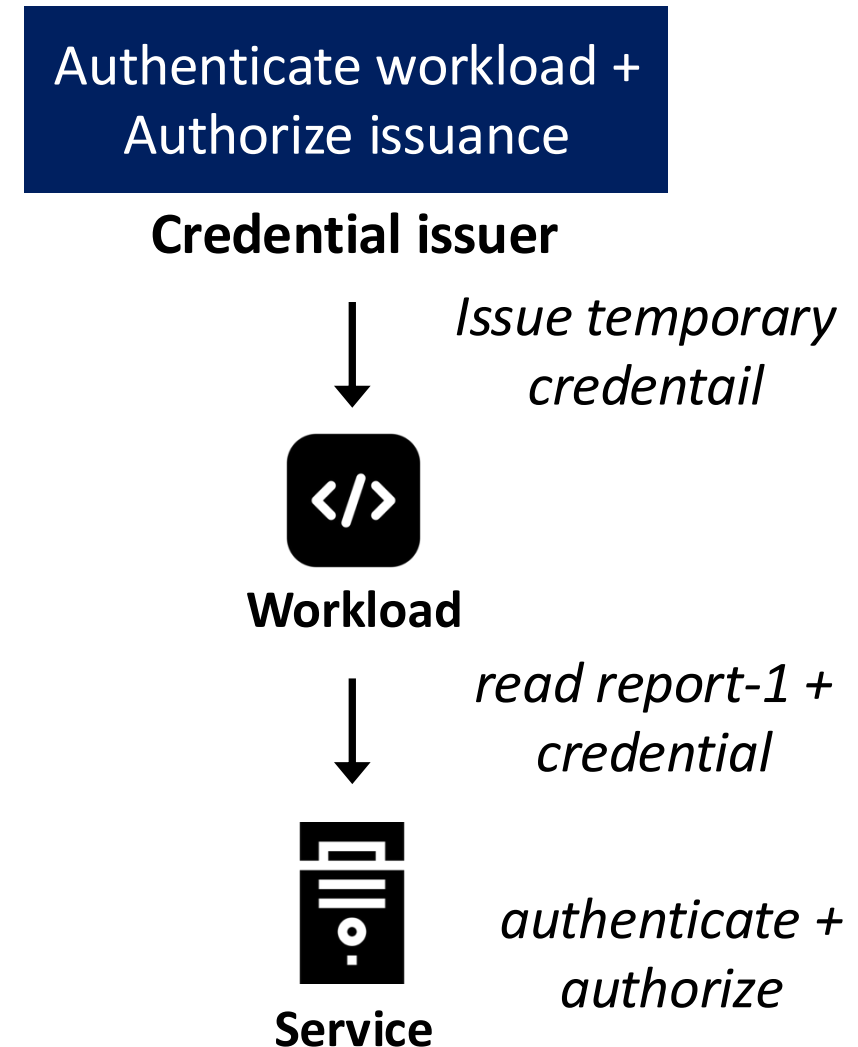
- With only a copied credential, the attacker is limited by its remaining lifetime and associated permissions
- If the workload *itself* is compromised, the attacker may obtain fresh credentials
 - Expiring one set does not end the access
- Contain the workload and stop further credential issuance
 - Waiting for expiration is not enough
- Also need to review actions already completed before containment
 - Expiration does not undo them

Accountability

- Accountability is the ability to trace actions and outcomes to the identity or session that performed them
 - Links actions and outcomes to the identity or session that performed them
- This requires logs of relevant actions, such as principal, action, resource, time, and result
 - A log is a record of system events or actions
- Shared accounts weaken accountability
 - If two employees share report-admin, the log cannot tell which person performed the action
- Logs show which account or session was used, not necessarily who physically performed the action

Putting It All Together: A Report-Reader Design

- **Task:** scheduled workload reads *report-1*
- **Identity:** workload has its own identity
- **Credentials:** obtains temporary credentials from a trusted issuer
- **Authorization:** report service permits only read *report-1*
- **Accountability:** log the identity, action, resource, time, and result
- **Residual risk:** a compromised workload can still use its legitimate read permission



Summary

- Authentication verifies identity evidence
- Authorization decides whether a requested operation is permitted
- Keep human and workload identities separate
- MFA strengthens authentication
- Define task-specific permissions, organize them through roles, and protect role assignment and other administrative changes
- Temporary credentials limit lifetime, not permission scope
- Protect credential issuance and maintain records for accountability