Cloud Security

Lecture 7

Disclaimer

- All figures in this presentation are taken from Cloud Computing by Thomas Erl, Zaigham Mahmood, and Ricardo Puttini, (ISBN: 0133387526) Copyright © 2013 Arcitura Education, Inc. All rights reserved.
- AWS Academy : AWS Cloud Foundation

Learning Objectives

- Basic Terms and Concepts
- Cloud Security Threats
- Cloud Security Mechanisms
- AWS Identity and Access Management (IAM)

Basic Terms and Concepts

- Confidentiality
  - A characteristic of something is being made accessible only to authorized parties.
- Integrity
  - A characteristic of not having been altered by an unauthorized party.
- Authenticity
  - A characteristic of something having been provided by an authorized source.
- Availability
  - A characteristic of being accessible and usable during a specified time period.

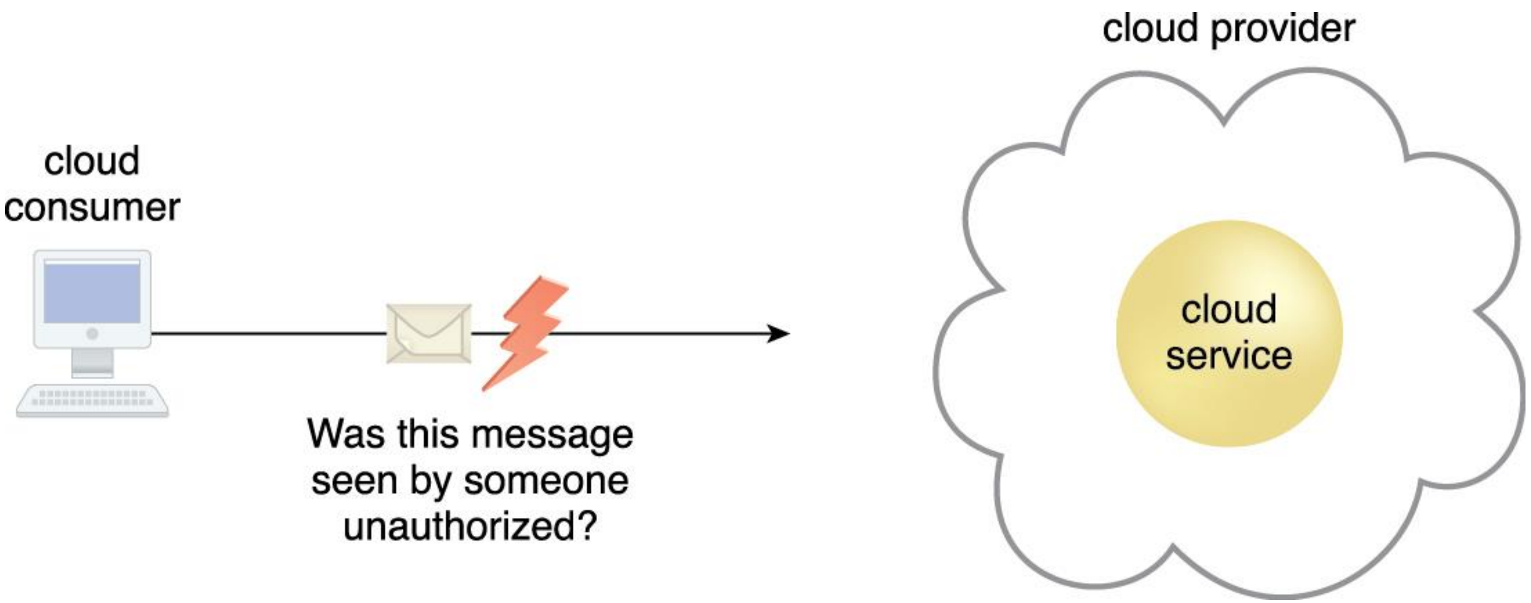


Figure 6.1 The message issued by the cloud consumer to the cloud service is considered confidential only if it is not accessed or read by an unauthorized party.

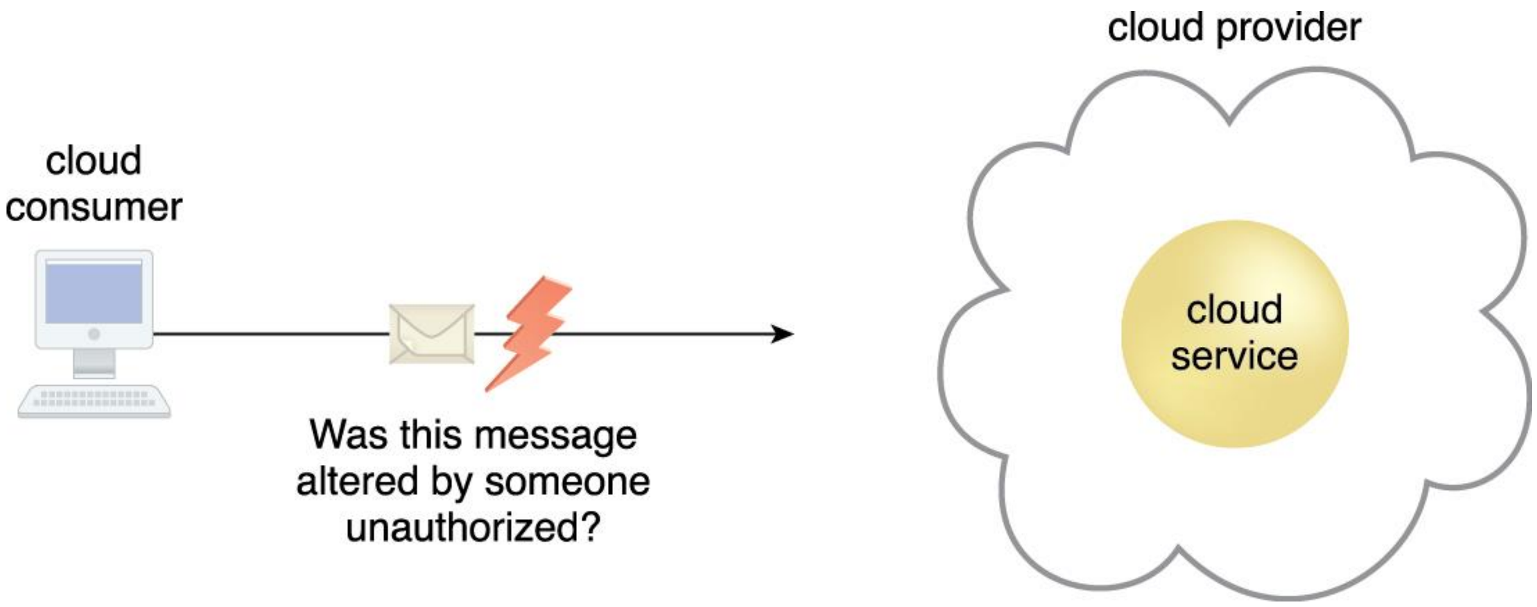


Figure 6.2 The message issued by the cloud consumer to the cloud service is considered to have integrity if it has not been altered.

Basic Terms and Concepts (cont.)

- Threat
  - A potential security violation that can challenge defenses in an attempt to breach privacy and/or cause harm.
- Vulnerability
  - A weakness that can be exploited either because it is protected by insufficient security controls, or because existing security controls are overcome by an attack.
- Risk
  - A possibility of loss or harm arising from performing an activity.

Basic Terms and Concepts (cont.)

- Security Controls
  - Countermeasures used to prevent or respond to security threats and to reduce or avoid risk.
- Security Mechanisms
  - Components comprising a defensive framework that protects IT resources, information, and services.
- Security Policies
  - A security policy establishes a set of security rules and regulations.

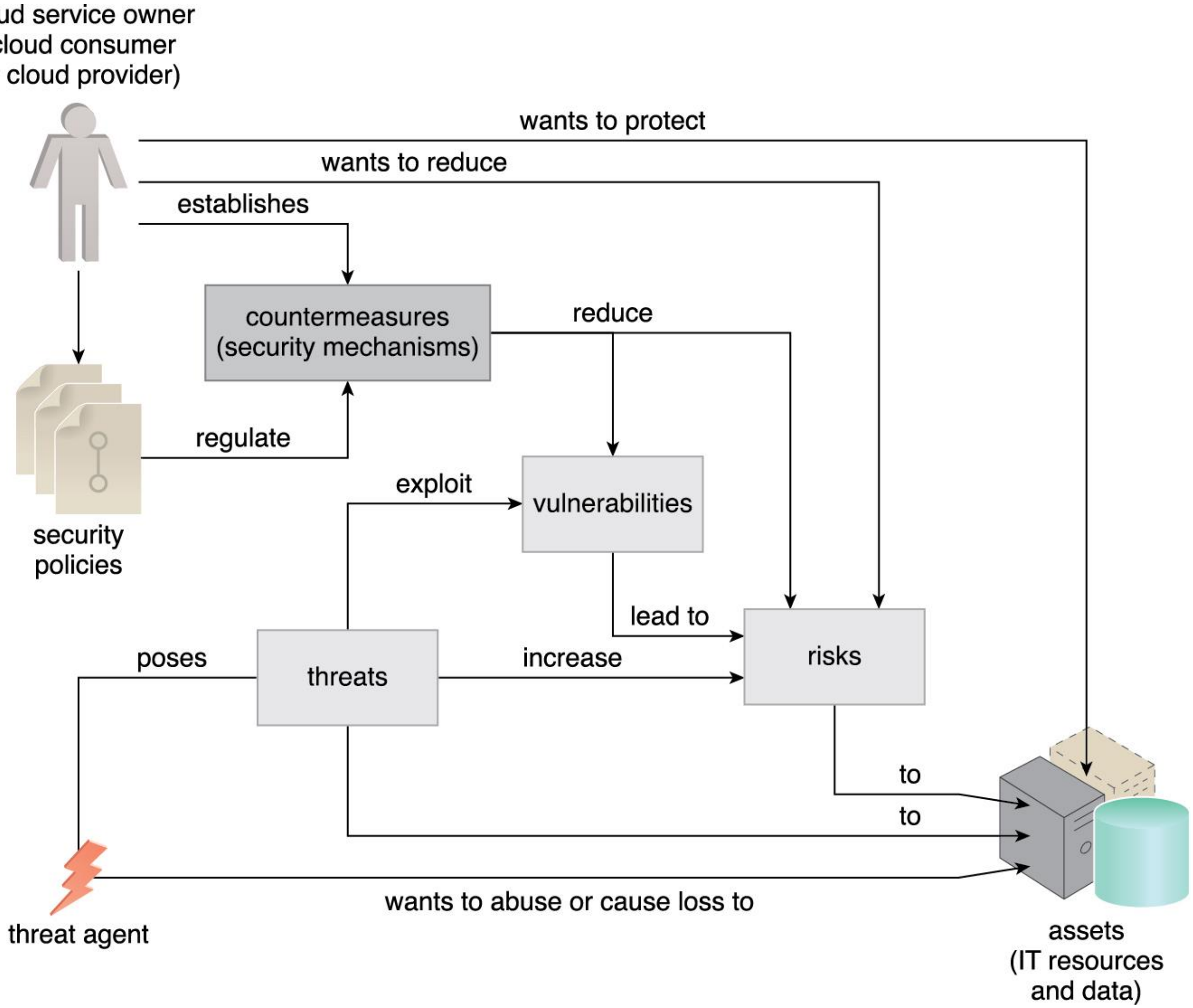
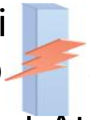


Figure 6.3 How security policies and security mechanisms are used to counter threats, vulnerabilities, and risks caused by threat agents.

Threat Agents

- Definition
  - A threat agent is an entity that poses a threat because it is capable of carrying out an attack.
  - Could be internal or external, from humans or software programs.
- Types of threat agents
  - Anonymous Attacker - a non-trusted cloud service consumer without permissions in the cloud; i.e. external software that launches network-level attachs through public networks.
  - Malicious Service Agent - intercept and forward network traffic that flows within a cloud; service agent with compromise or malicious logic or external program and intercept and corrupt message contents.
  - Trusted Attacker - trusted cloud consumer and exploits legitimate credentials to target cloud providers and other tenants with whom they share IT resources (i.e. malicious tenants)
  - Malicious Insider - human threat agents acting on behalf or in relation to the cloud provider



Cloud Security Threats

- Traffic Eavesdropping
  - Occurs when data being transferred to or within a cloud (usually from the cloud consumer to the cloud provider) is passively intercepted by a malicious service agent for illegitimate information gathering purposes. .
- Malicious Intermediary
  - Messages are intercepted and altered by a malicious service agent, thereby potentially compromising the message's confidentiality and/or integrity.

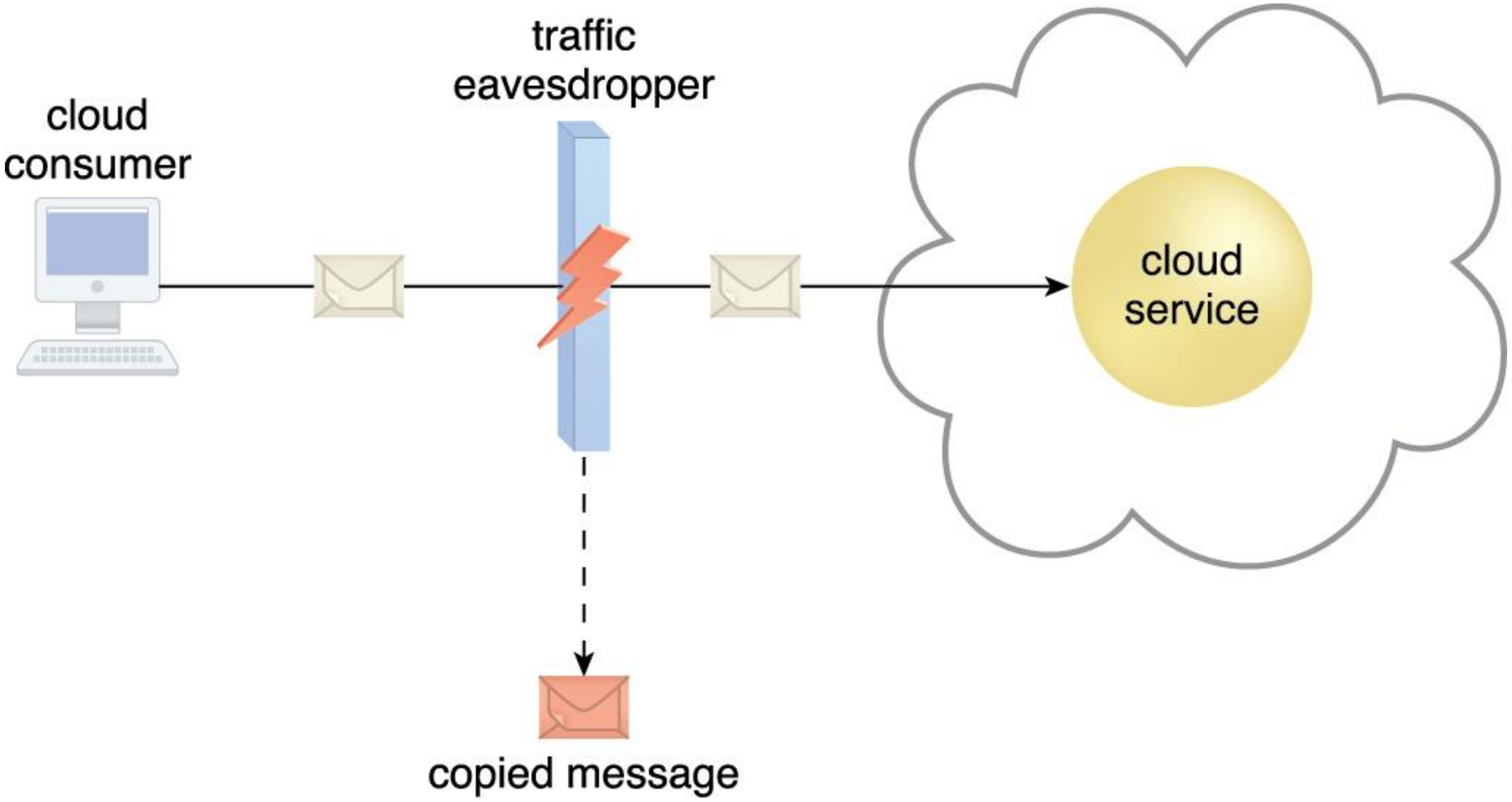


Figure 6.8 An externally positioned malicious service agent carries out a traffic eavesdropping attack by intercepting a message sent by the cloud service consumer to the cloud service. The service agent makes an unauthorized copy of the message before it is sent along its original path to the cloud service.

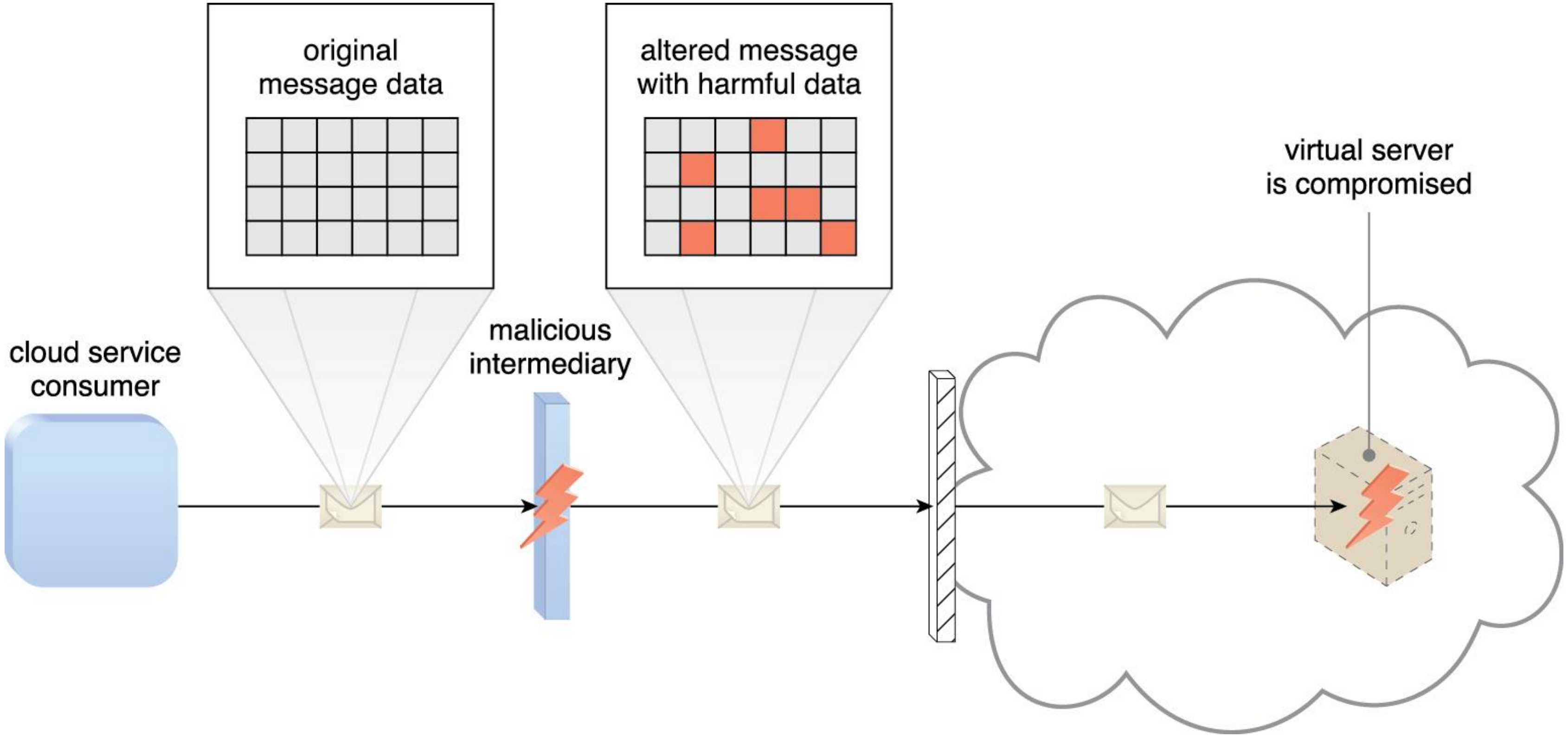


Figure 6.9 The malicious service agent intercepts and modifies a message sent by a cloud service consumer to a cloud service (not shown) being hosted on a virtual server. Because harmful data is packaged into the message, the virtual server is compromised.

Cloud Security Threats (cont.)

- Denial of Service
  - To overload IT resources to the point where they cannot function properly.
  - Three common ways:
    - Workload on cloud artificially increased with imitation messages.
    - Network overloaded with traffic to reduce its responsiveness.
    - Multiple cloud service requests are sent to consume excessive memory and processing resources.
- Insufficient Authorization
  - Access is granted to an attacker erroneously or too broadly.

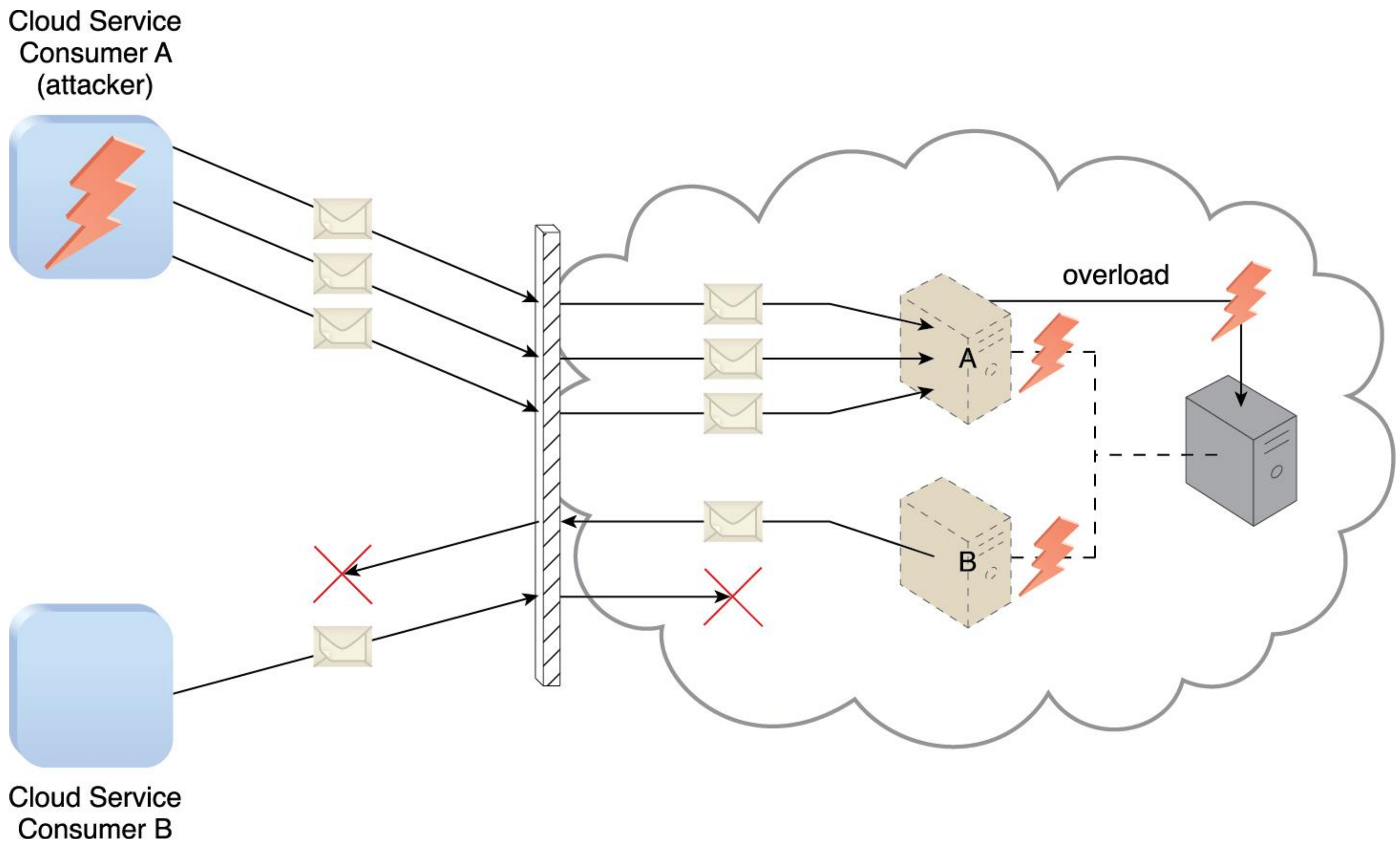


Figure 6.10 Cloud Service Consumer A sends multiple messages to a cloud service (not shown) hosted on Virtual Server A. This overloads the capacity of the underlying physical server, which causes outages with Virtual Servers A and B. As a result, legitimate cloud service consumers, such as Cloud Service Consumer B, become unable to communicate with any cloud services hosted on Virtual Servers A and B.

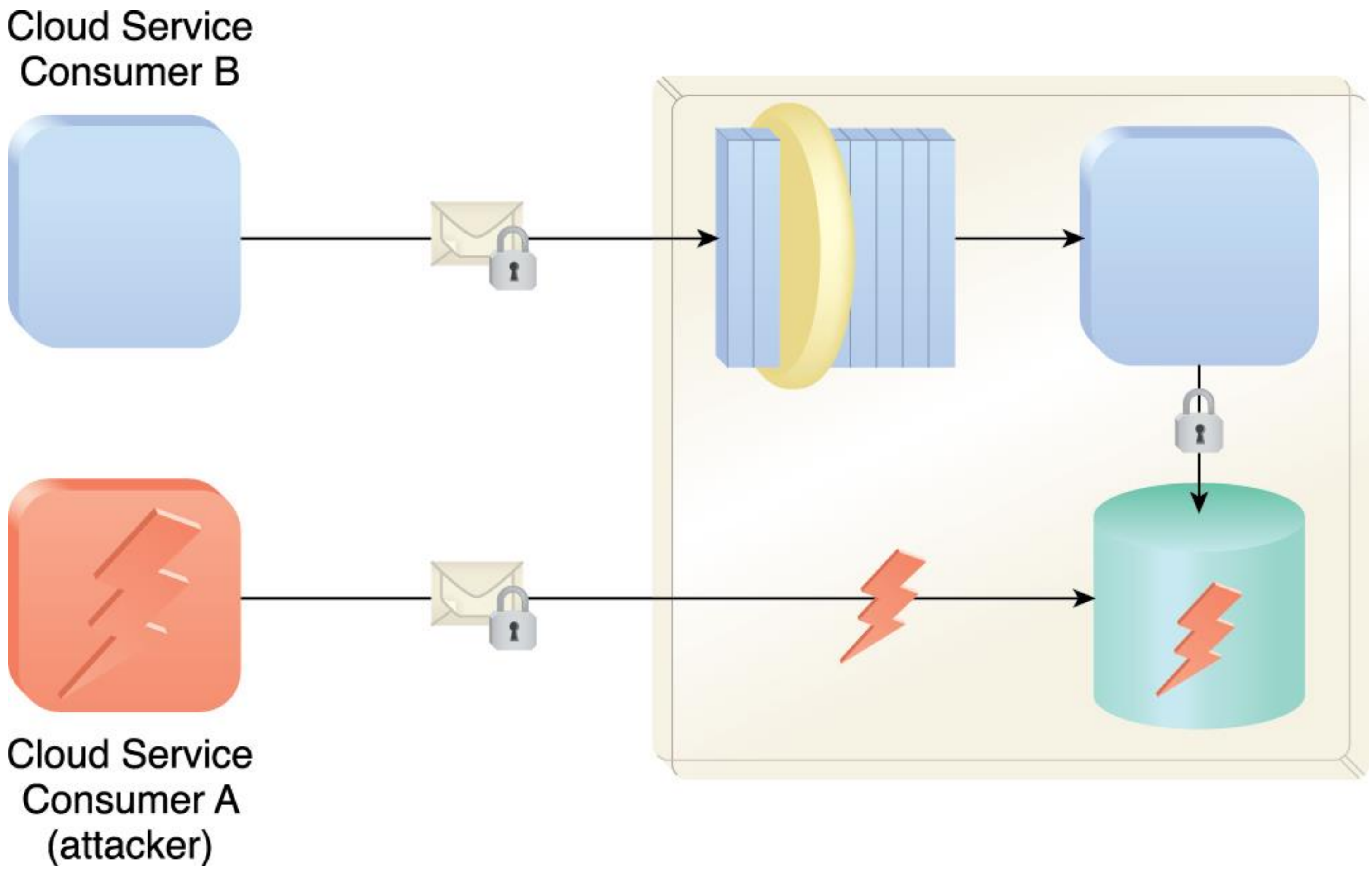


Figure 6.11 Cloud Service Consumer A gains access to a database that was implemented under the assumption that it would only be accessed through a Web service with a published service contract (as per Cloud Service Consumer B).

Cloud Security Threats (cont.)

- Virtualization Attack
  - Exploits vulnerabilities in the virtualization platform to jeopardize its confidentiality, integrity, and/or availability.
  - Compromising physical IT resources through virtualization platform.
- Overlapping Trust Boundaries
  - Malicious cloud service consumers can target shared IT resources with the intention of compromising cloud consumers or other IT resources that share the same trust boundary.

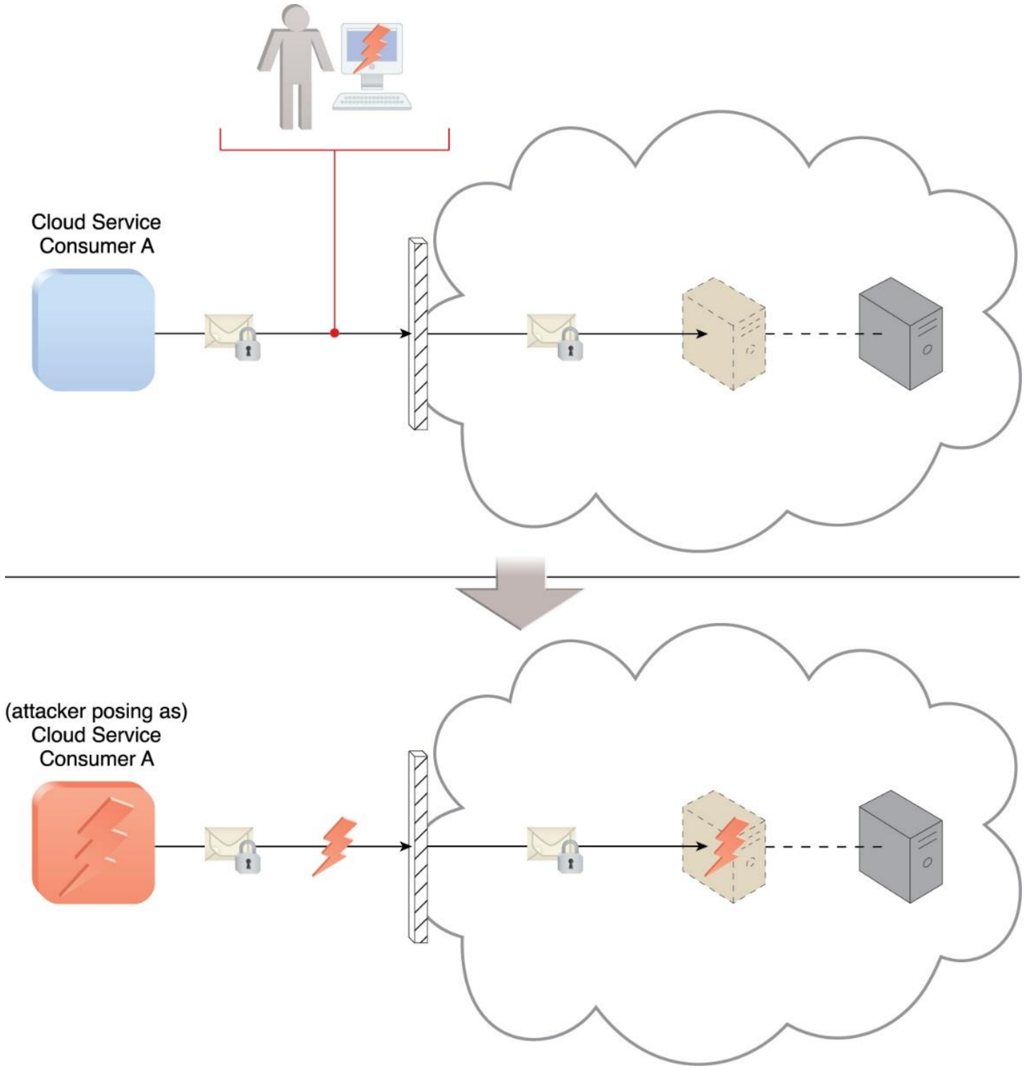


Figure 6.12 An attacker has cracked a weak password used by Cloud Service Consumer A. As a result, a malicious cloud service consumer (owned by the attacker) is designed to pose as Cloud Service Consumer A in order to gain access to the cloud-based virtual server.

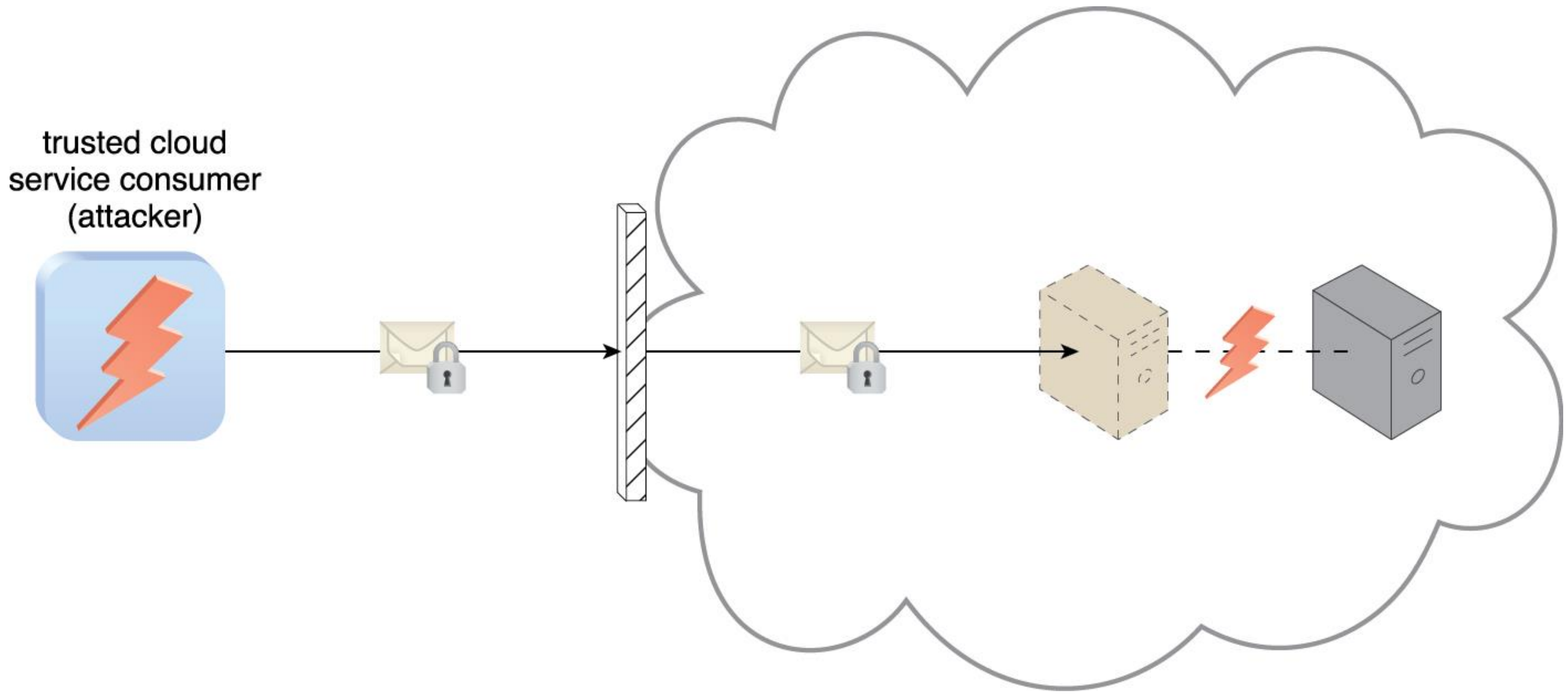


Figure 6.13 An authorized cloud service consumer carries out a virtualization attack by abusing its administrative access to a virtual server to exploit the underlying hardware.

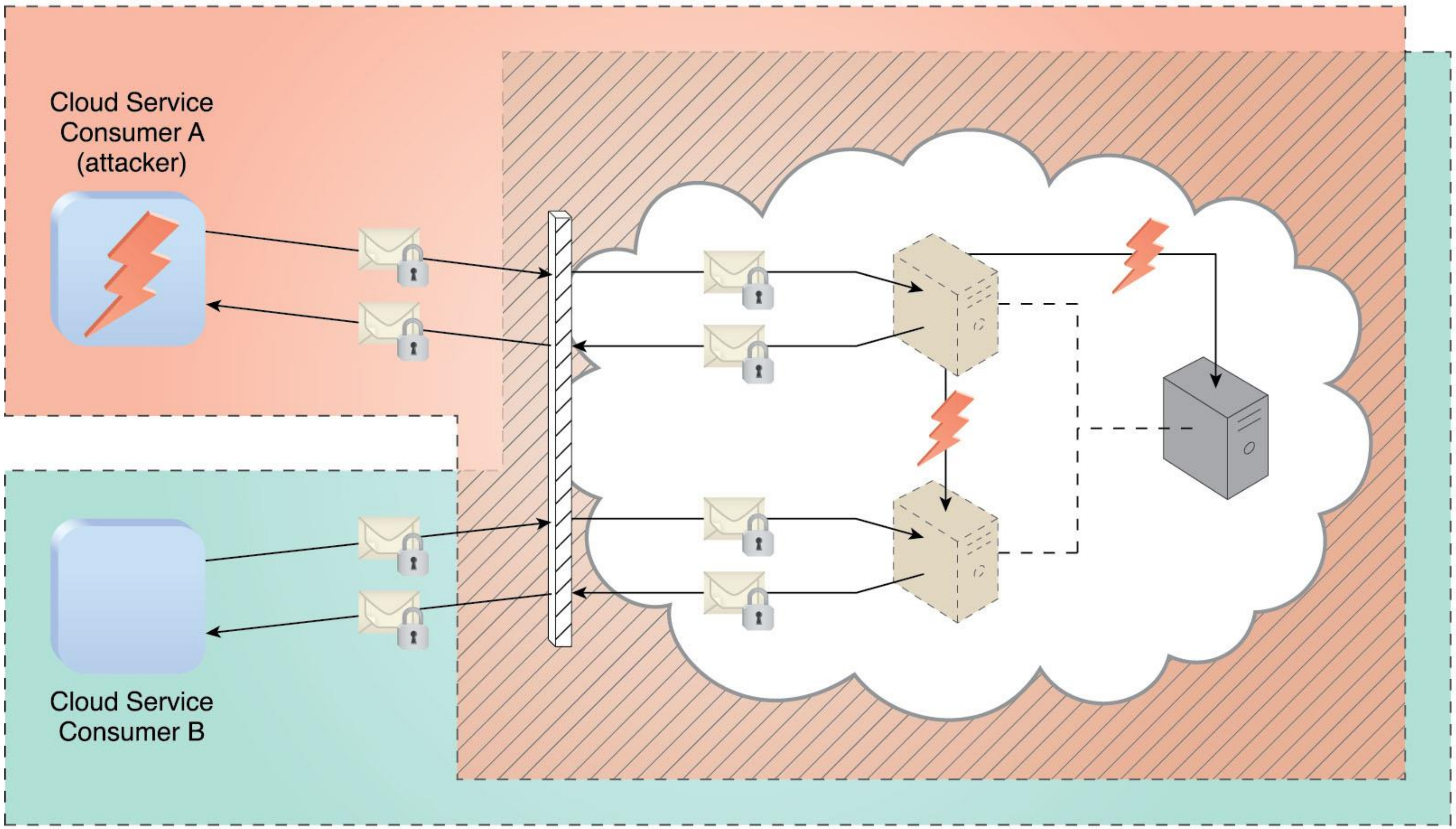


Figure 6.14 Cloud Service Consumer A is trusted by the cloud and therefore gains access to a virtual server, which it then attacks with the intention of attacking the underlying physical server and the virtual server used by Cloud Service Consumer B.

Some Other Considerations

- Flawed Implementations
  - Deploying flawed cloud-based solutions
- Security Policy Disparity
  - Understanding how a cloud provider defines and imposes proprietary
- Contracts
  - Clearly define liability, indemnity, and blame for potential security breaches between cloud consumers and cloud provider.
- Risk Management
  - Perform corresponding assessment of the identified risks.

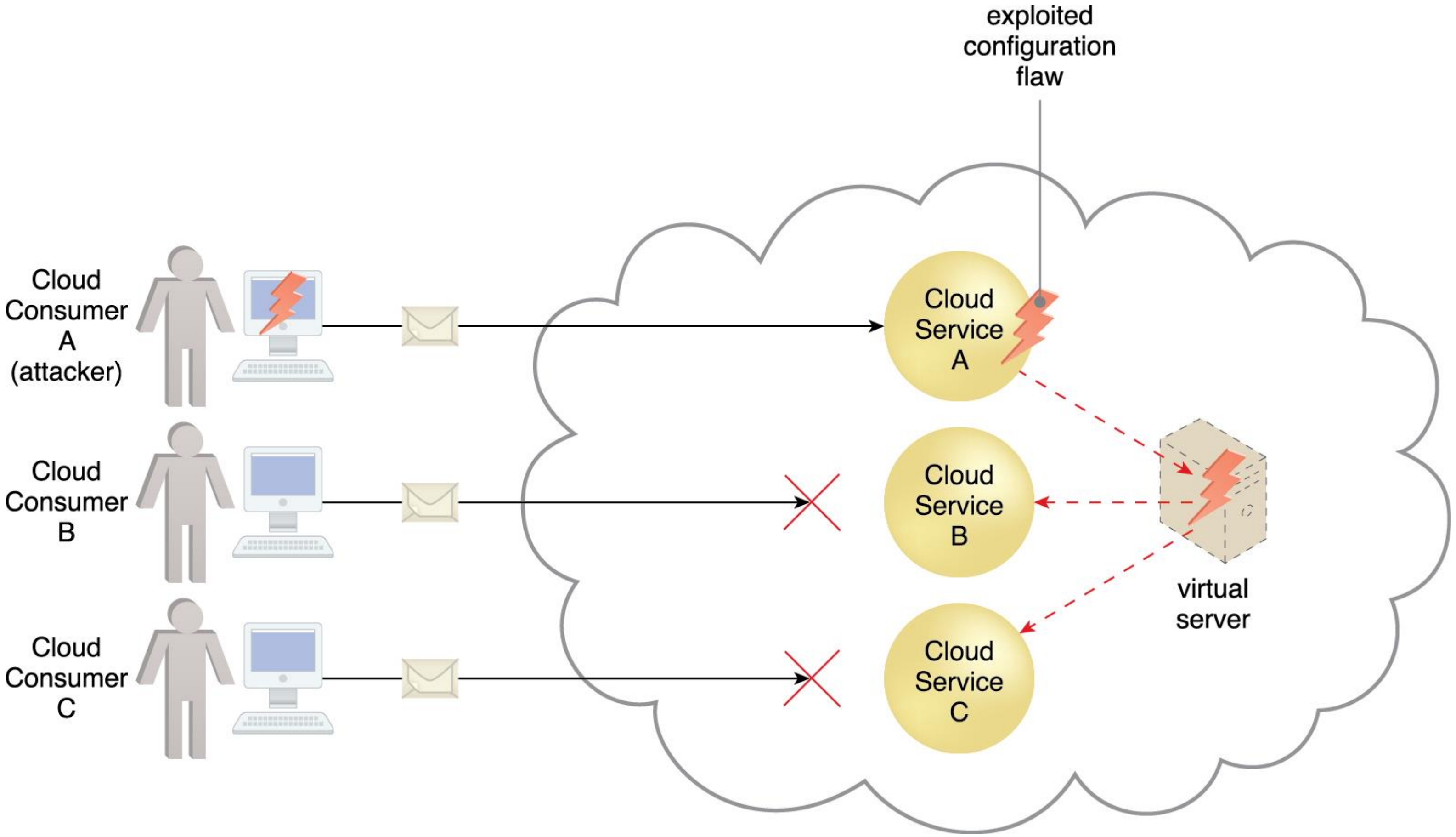


Figure 6.15 Cloud Service Consumer A’s message triggers a configuration flaw in Cloud Service A, which in turn causes the virtual server that is also hosting Cloud Services B and C to crash.

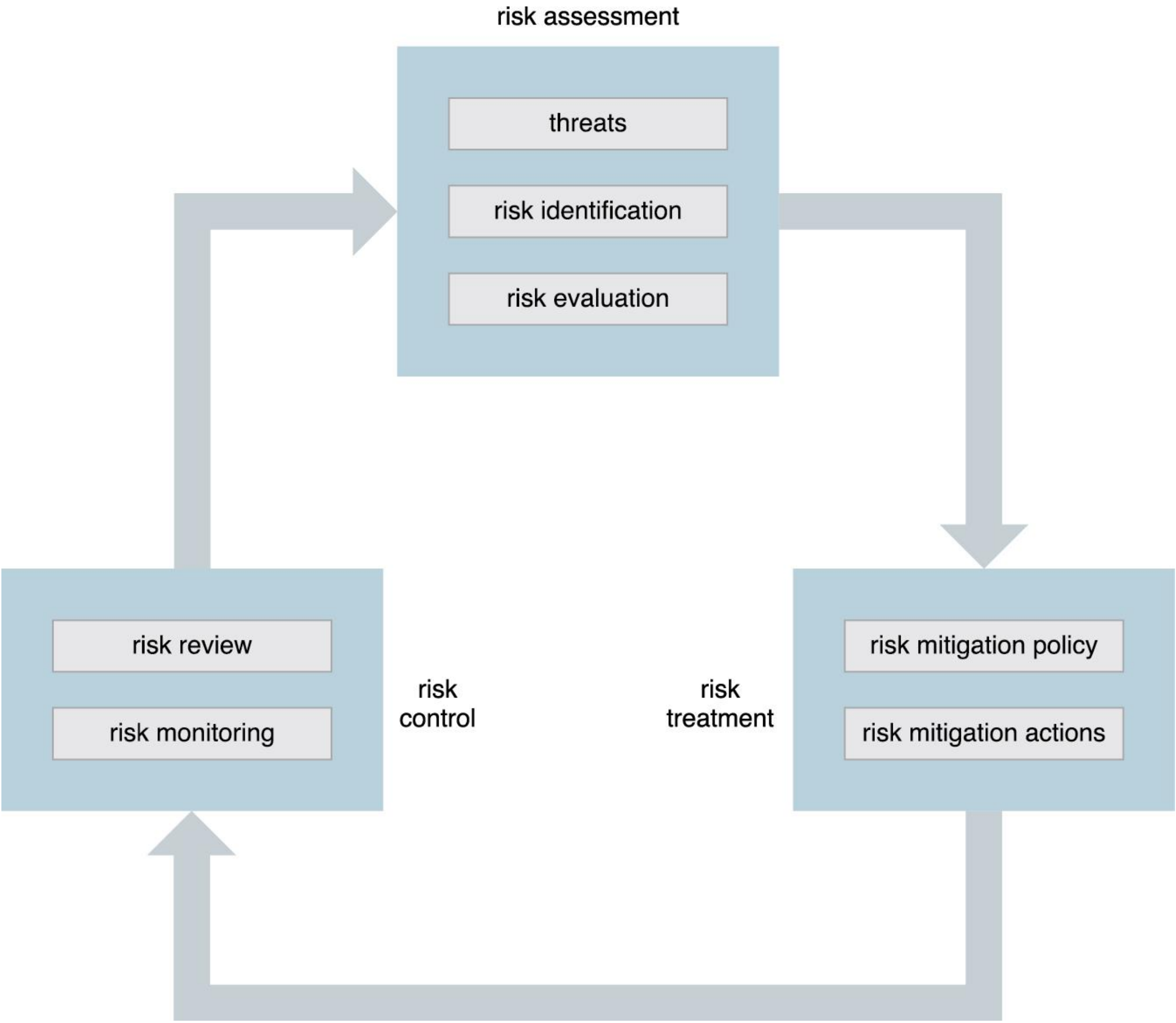


Figure 6.16 The on-going risk management process, which can be initiated from any of the three stages.

Cloud Security Mechanisms

Encryption

- Data normally coded as plaintext.
- During transmission, plaintext is vulnerable to unauthorized and potentially malicious access.
- Encryption:
  - A digital coding system dedicated to preserving the confidentiality and integrity of data.
  - Encoding plaintext into protected and unreadable format, i.e. ciphertext.
  - When encryption is applied to plaintext, the data is paired with a string of characters called encryption key, a secret message that is established by and shared among authorized parties.
  - Encryption key is used to decrypt the ciphertext into its original plaintext.

Encryption (cont.)

- Encryption mechanism reduces the risk of the following threats:
  - Traffic eavesdropping
  - Malicious intermediary
  - Insufficient authorization
  - Overlapping trust boundaries
- Two forms of encryption:
  - Symmetric: same key for both encryption and decryption.
  - Asymmetric: Relies on two different keys: public and private keys.

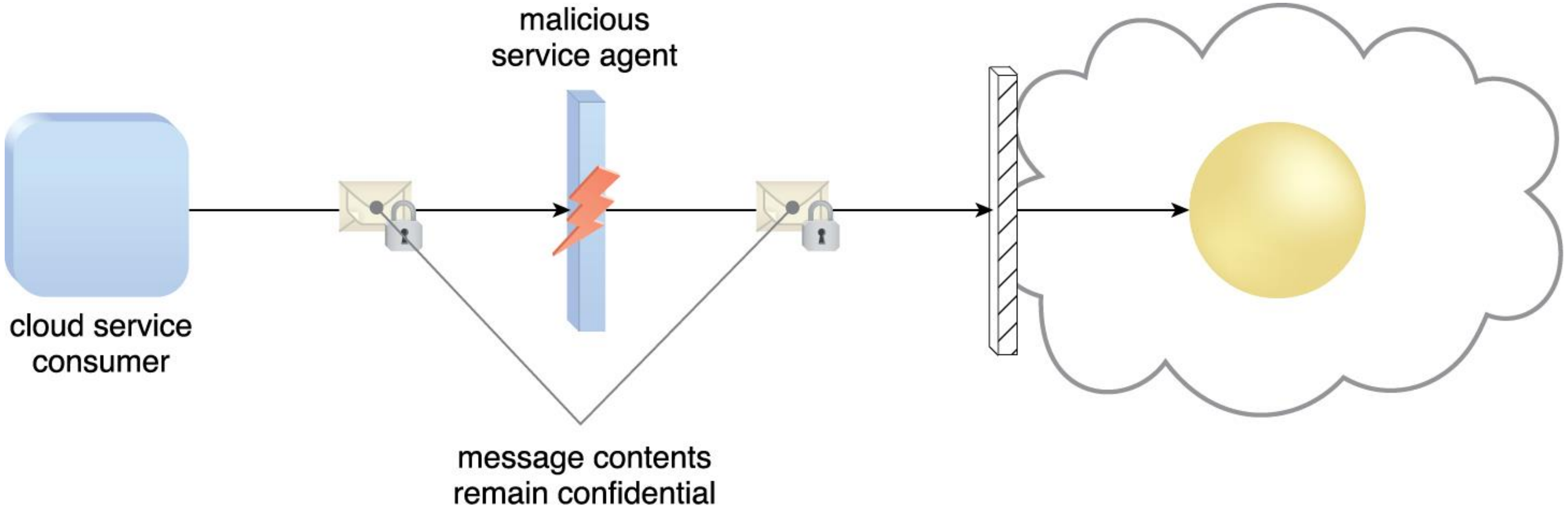
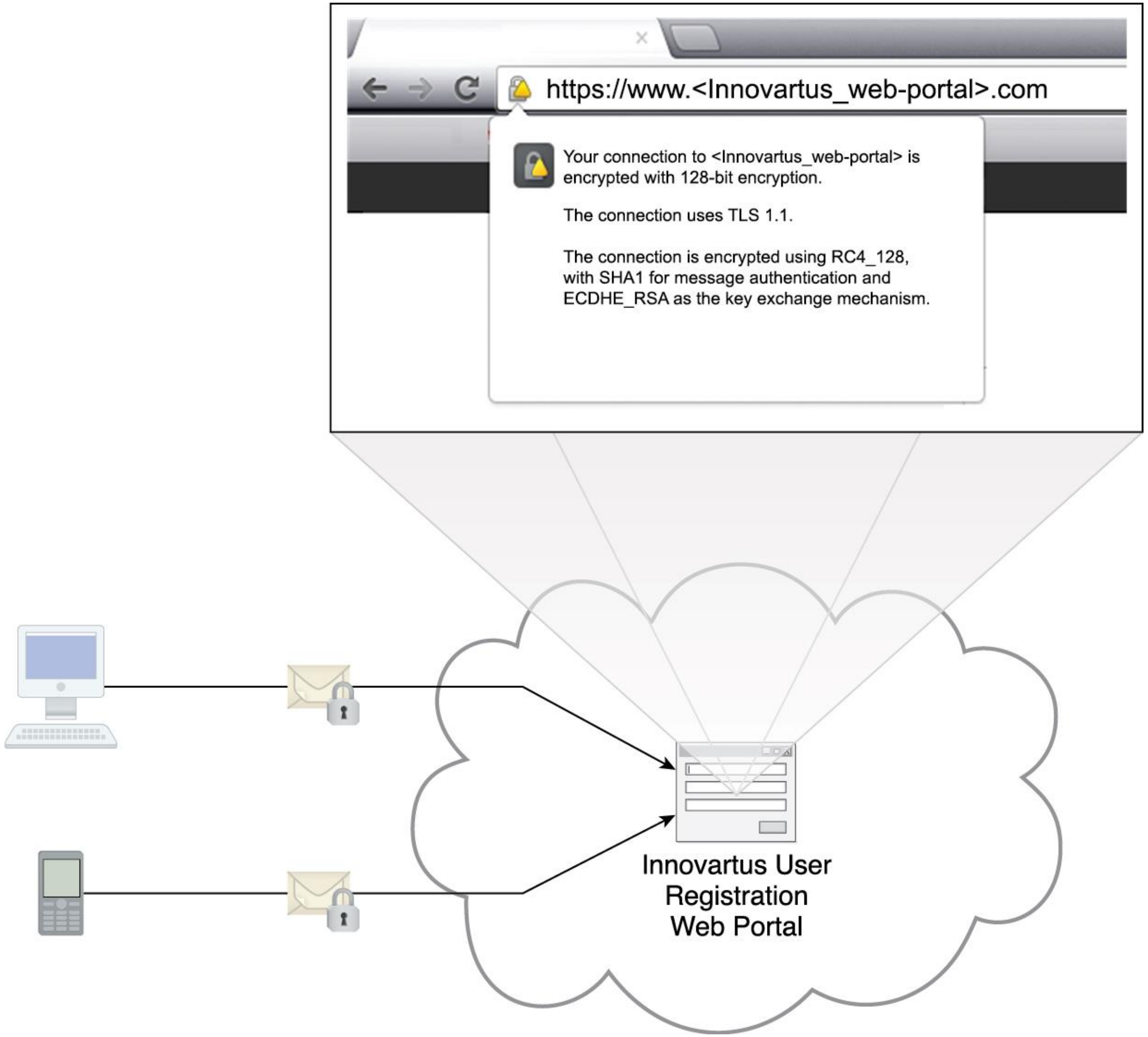


Figure 10.1 A malicious service agent is unable to retrieve data from an encrypted message. The retrieval attempt may furthermore be revealed to the cloud service consumer. (Note the use of the lock symbol to indicate that a security mechanism has been applied to the message contents.)



https://www.<Innovartus_web-portal>.com

Your connection to <Innovartus_web-portal> is encrypted with 128-bit encryption.

The connection uses TLS 1.1.

The connection is encrypted using RC4_128, with SHA1 for message authentication and ECDHE_RSA as the key exchange mechanism.

Figure 10.2 The encryption mechanism is added to the communication channel between outside users and Innovartus’ User Registration Portal. This safeguards message confidentiality via the use of HTTPS.

Hashing

- It is used when a one-way, non-reversible form of data protection is required.
- Once hashing is applied to a message, it is locked and no key is provided for the message to be unlocked.
- A common application: storage of passwords.
- Can be used to derive a hashing code or message digest from a message, which is often of a fixed length and smaller than the original message.
- Used for message authentication.
    - A potential security violation that can challenge defenses in an attempt to breach privacy and/or cause harm.

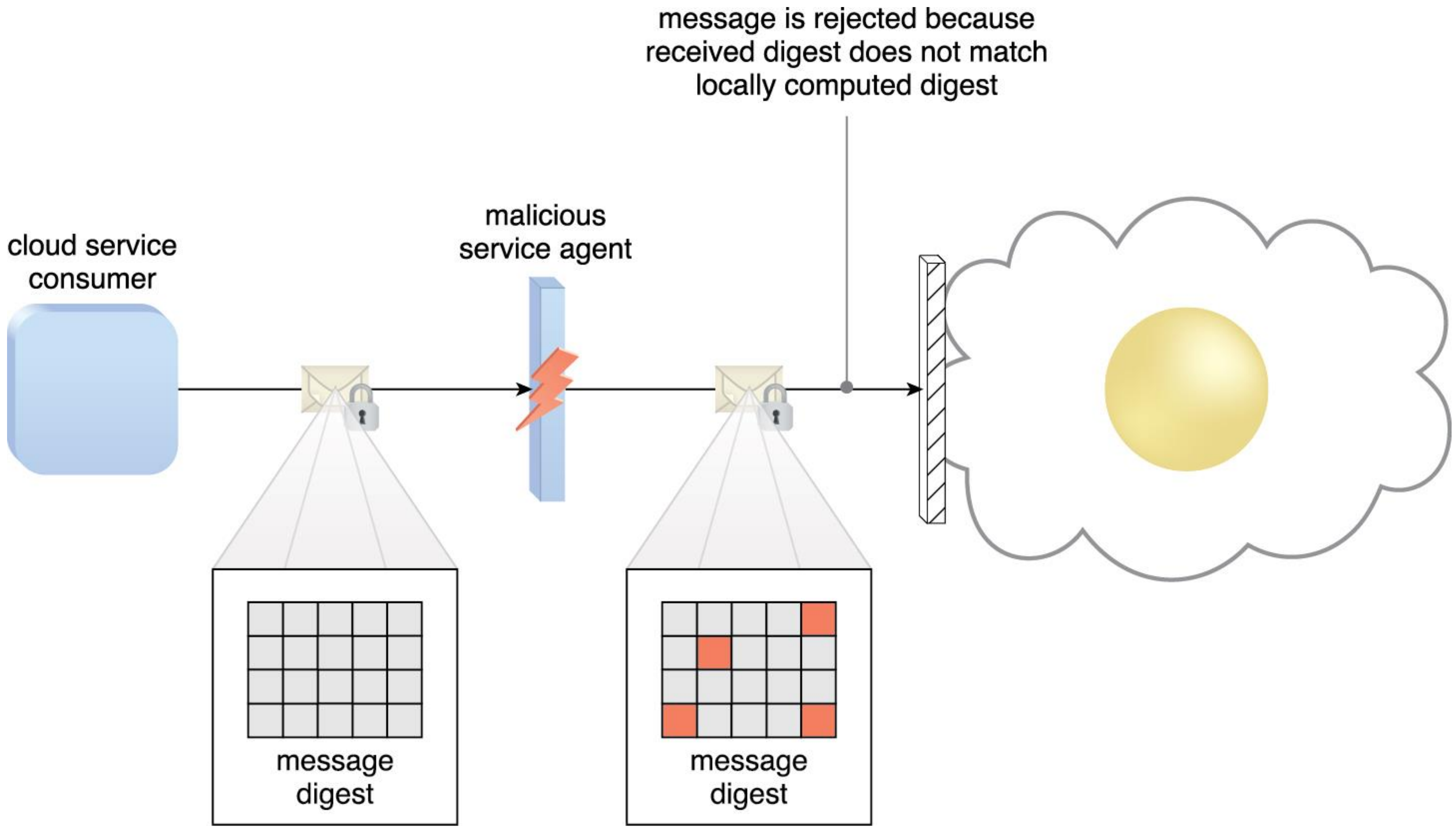
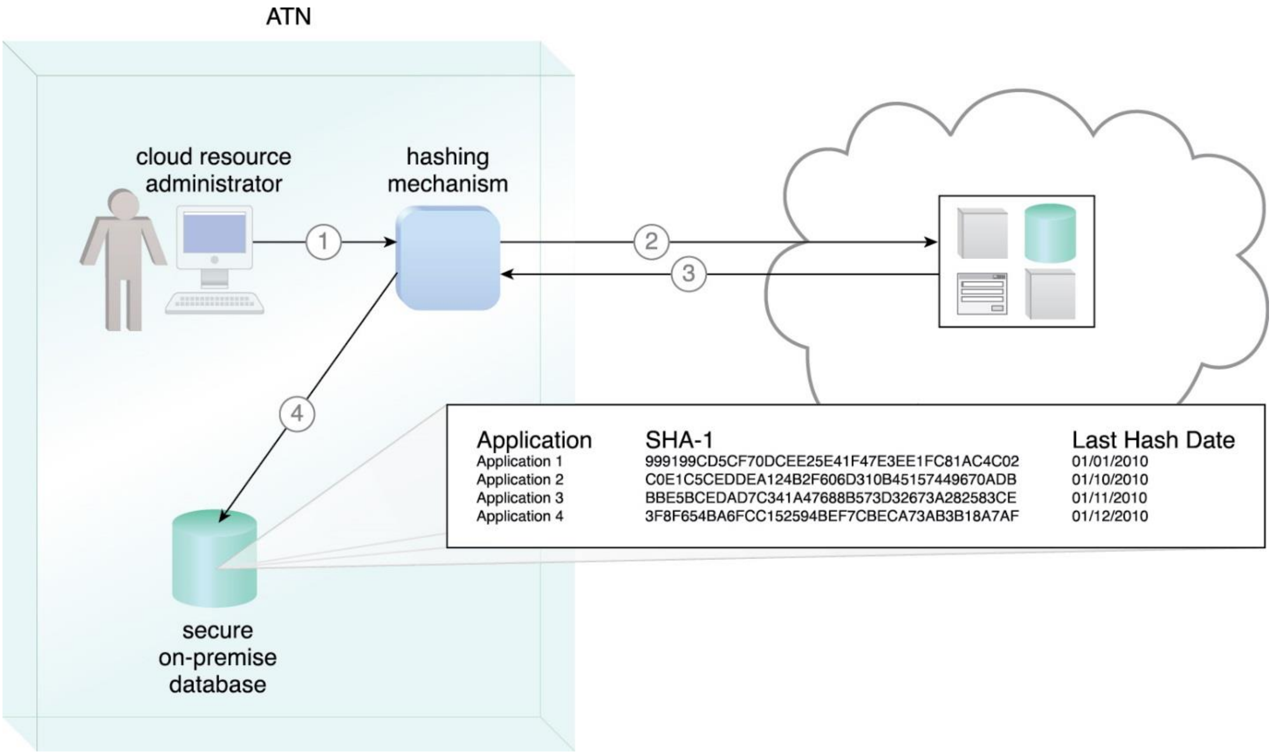

message is rejected because received digest does not match locally computed digest
malicious service agent
cloud service consumer
message digest
message digest

Figure 10.3 A hashing function is applied to protect the integrity of a message that is intercepted and altered by a malicious service agent, before it is forwarded. The firewall can be configured to determine that the message has been altered, thereby enabling it to reject the message before it can proceed to the cloud service.

Hashing: Case Study Example



Application	SHA-1	Last Hash Date
Application 1	999199CD5CF70DCEE25E41F47E3EE1FC81AC4C02	01/01/2010
Application 2	C0E1C5CEDDEA124B2F606D310B45157449670ADB	01/10/2010
Application 3	BBE5BCEDAD7C341A47688B573D32673A282583CE	01/11/2010
Application 4	3F8F654BA6FCC152594BEF7CBECA73AB3B18A7AF	01/12/2010

Figure 10.4 A hashing procedure is invoked when the PaaS environment is accessed (1). The applications that were ported to this environment are checked (2) and their message digests are calculated (3). The message digests are stored in a secure on-premise database (4), and a notification is issued if any of their values are not identical to the ones in storage.

Digital Signature

- A means of providing data authenticity and integrity through authentication and non-repudiation.
- It provides evidence that the message received is the same as the one created by its rightful sender.
- Both hashing and asymmetrical encryption are involved in the creation of a digital signature.
- Helps mitigate the malicious intermediary, insufficient authorization, and overlapping trust boundary security threats.

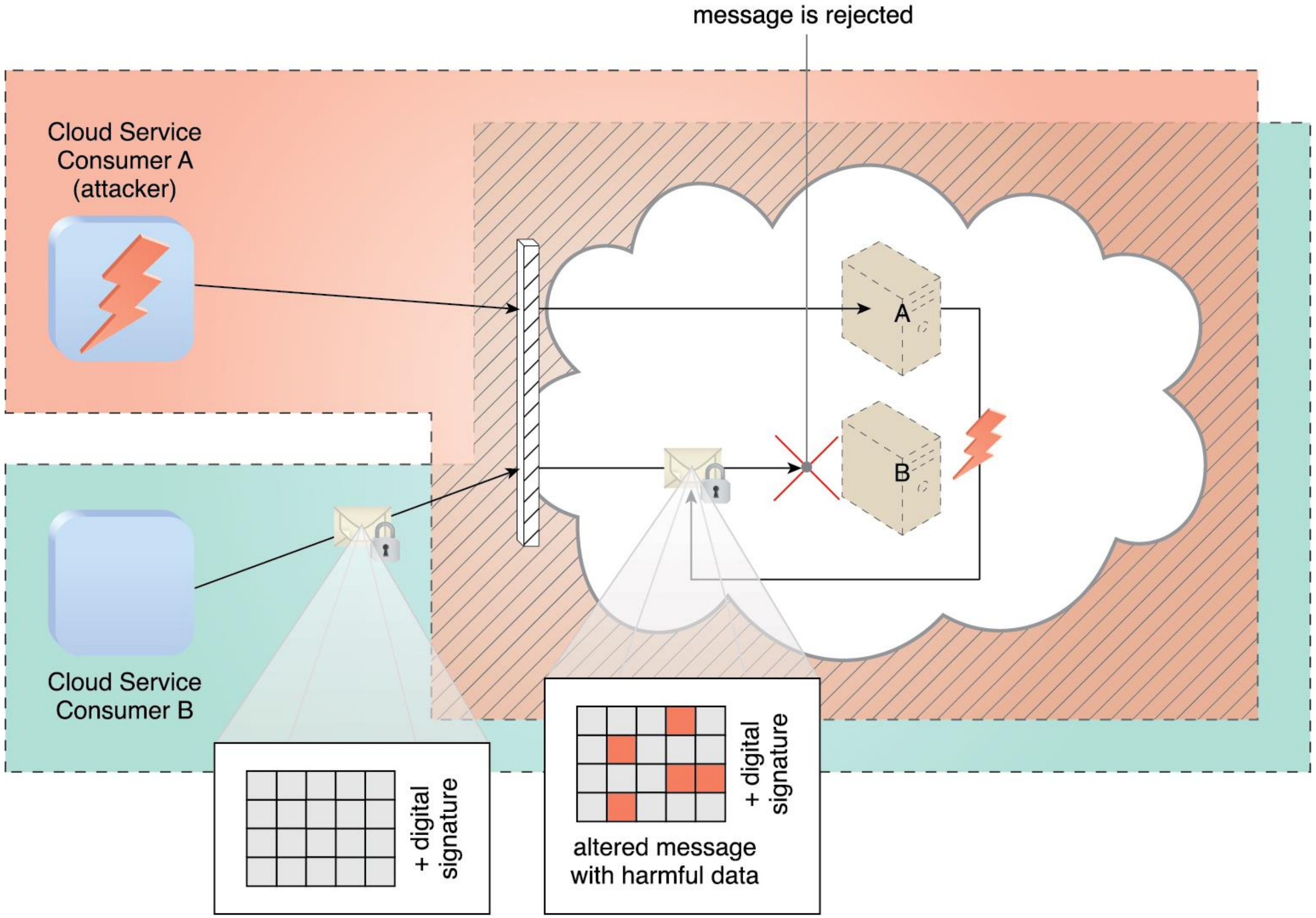
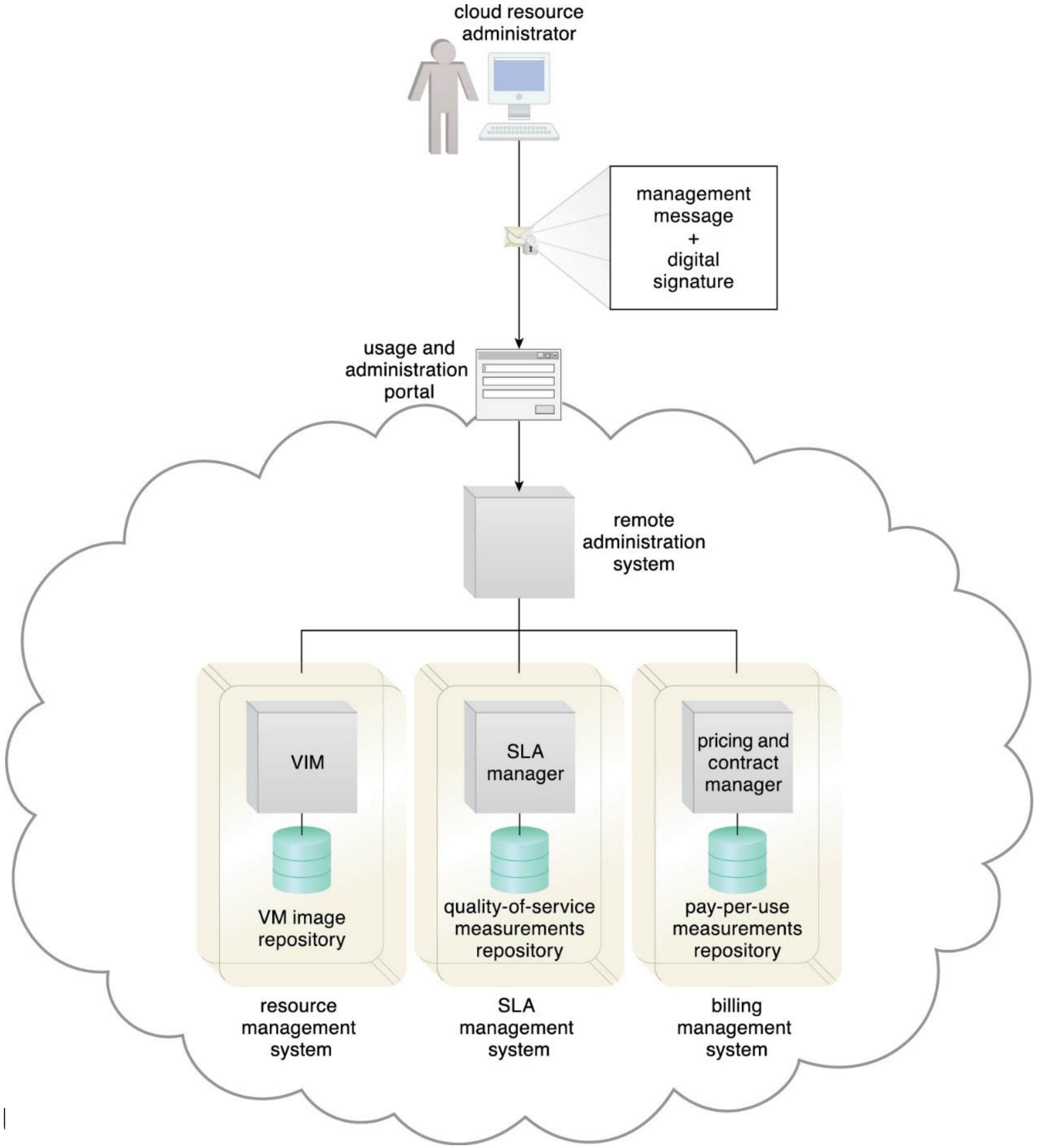


Figure 10.5 Cloud Service Consumer B sends a message that was digitally signed but was altered by trusted attacker Cloud Service Consumer A. Virtual Server B is configured to verify digital signatures before processing incoming messages even if they are within its trust boundary. The message is revealed as illegitimate due to its invalid digital signature, and is therefore rejected by Virtual Server B.

DigSig: Case Study Example

Figure 10.6 Whenever a cloud consumer performs a management action that is related to IT resources provisioned by DTGOV, the cloud service consumer program must include a digital signature in the message request to prove the legitimacy of its user.


cloud resource administrator
management message + digital signature
usage and administration portal
remote administration system
VIM
VM image repository
SLA manager
quality-of-service measurements repository
pricing and contract manager
pay-per-use measurements repository
resource management system
SLA management system
billing management system

Public Key Infrastructure (PKI)

- A system of protocols, data formats, rules, and practices that enable large-scale systems to securely use public key cryptography.
- This system is used to associate public keys with their corresponding key owners (known as *public key identification*).
- It relies on the use of digital certificates: digitally-signed data structures that bind public keys to certificate owner identities.
- Digital certificates are usually digitally signed by a third-party certificate authority (CA).
- The PKI mechanism is primarily used to counter the insufficient authorization threat.

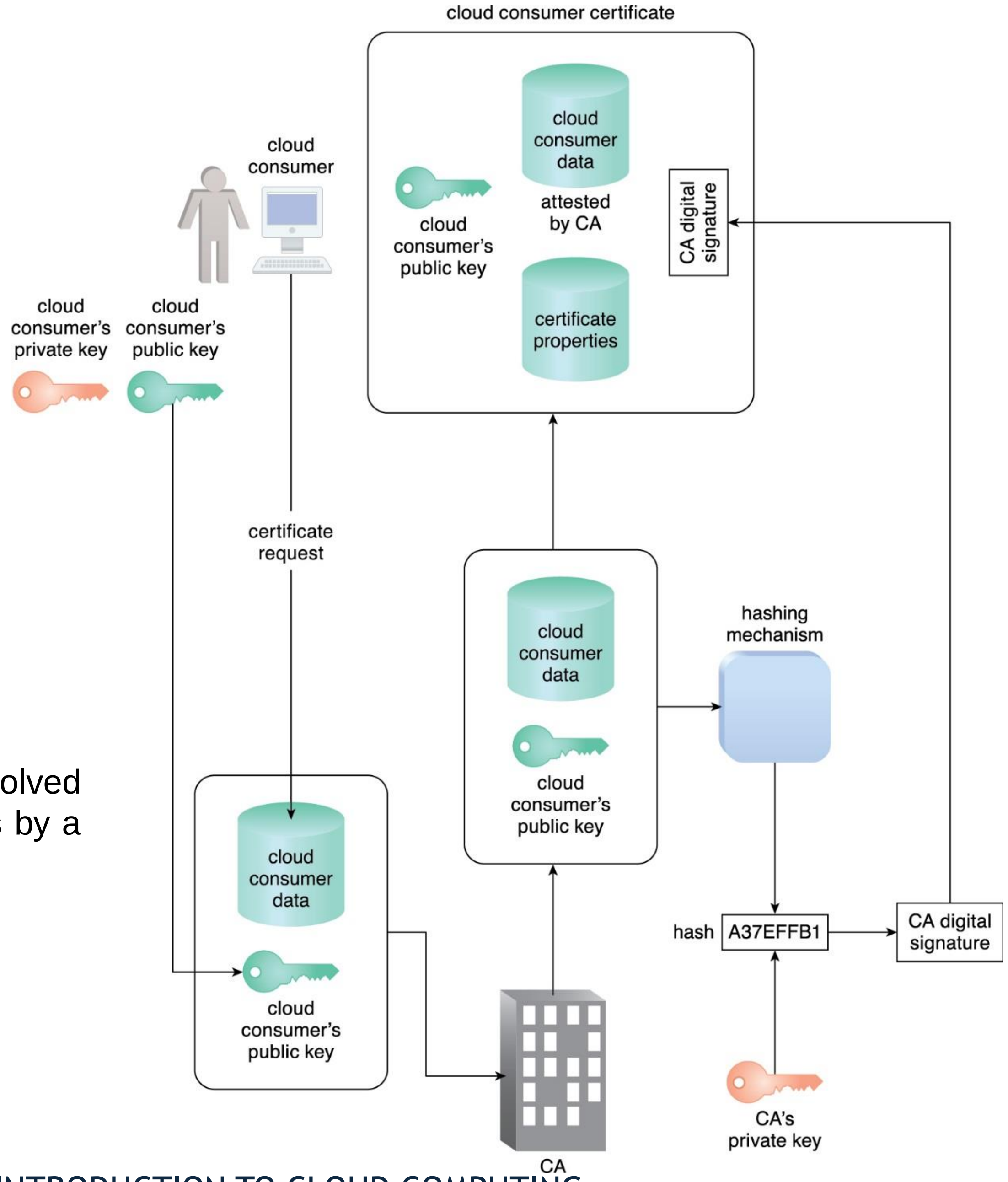
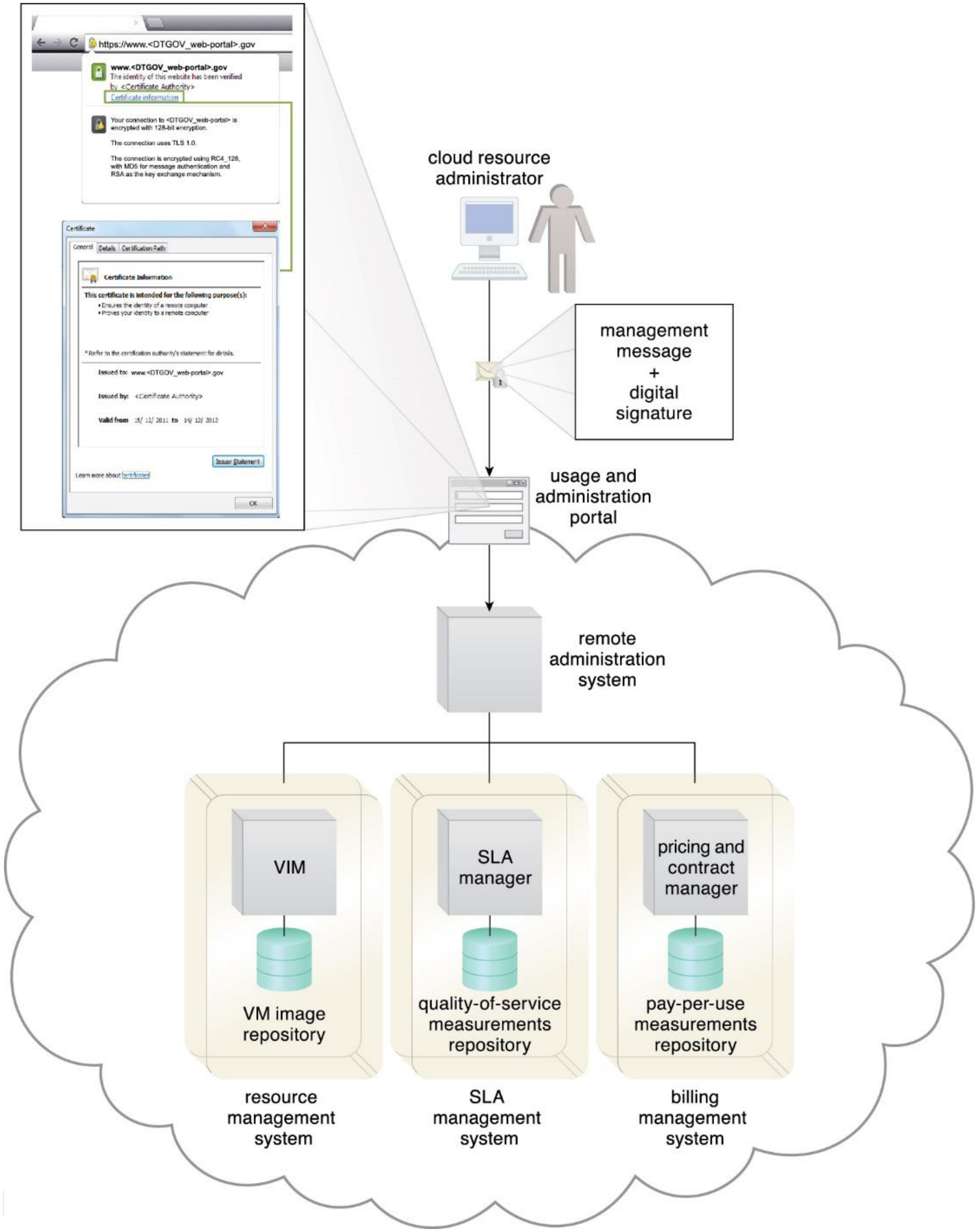


Figure 10.7 The common steps involved during the generation of certificates by a certificate authority.

PKI: Case Study Example



https://www.<DTGOV_web-portal>.gov

www.<DTGOV_web-portal>.gov
The identity of this website has been verified by <Certificate Authority>
Certificate information

Your connection to <DTGOV_web-portal> is encrypted with 128-bit encryption.

The connection uses TLS 1.0.

The connection is encrypted using RC4_128, with MD5 for message authentication and RSA as the key exchange mechanism.

Certificate

General | Details | Certification Path

Certificate Information

This certificate is intended for the following purpose(s):
- Ensures the identity of a remote computer
- Proves your identity to a remote computer

* Refer to the certification authority's statement for details.

Issued to: www.<DTGOV_web-portal>.gov

Issued by: <Certificate Authority>

Valid from 15/ 12/ 2011 **to** 14/ 12/ 2012

Issuer Statement

Learn more about certificates

OK

cloud resource administrator
management message + digital signature
usage and administration portal
remote administration system
VIM
VM image repository
resource management system
SLA manager
quality-of-service measurements repository
SLA management system
pricing and contract manager
pay-per-use measurements repository
billing management system

Figure 10.8 An external cloud resource administrator uses a digital certificate to access the Web- based management environment. DTGOV's digital certificate is used in the HTTPS connection and then signed by a trusted CA.

Identity & Access Management (IAM)

- A collection of components and policies necessary to control and track user identities and access privileges for IT resources, environment, and systems.
- IAM mechanisms exist as systems comprised of four main components:
  - Authentication: Username & password, DigSig, DigCert, biometric hardware, specialized software.
  - Authorization: access controls.
  - User Management: Creating new user identities, access groups, password resets, password policies, privilege management.
  - Credential Management: establishes identities and access control rules for defined user accounts.
- IAM is primarily used to counter the insufficient authorization, DoS, and overlapping trust boundaries threats.

Single Sign-On (SSO)

- Enables one cloud service consumer to be authenticated by a security broker, which establishes a security context that is persisted while the cloud service consumer accesses other cloud services or cloud-based IT resources.
- It primarily enhances the usability of cloud-based environments for access and management of distributed IT resources and solutions.

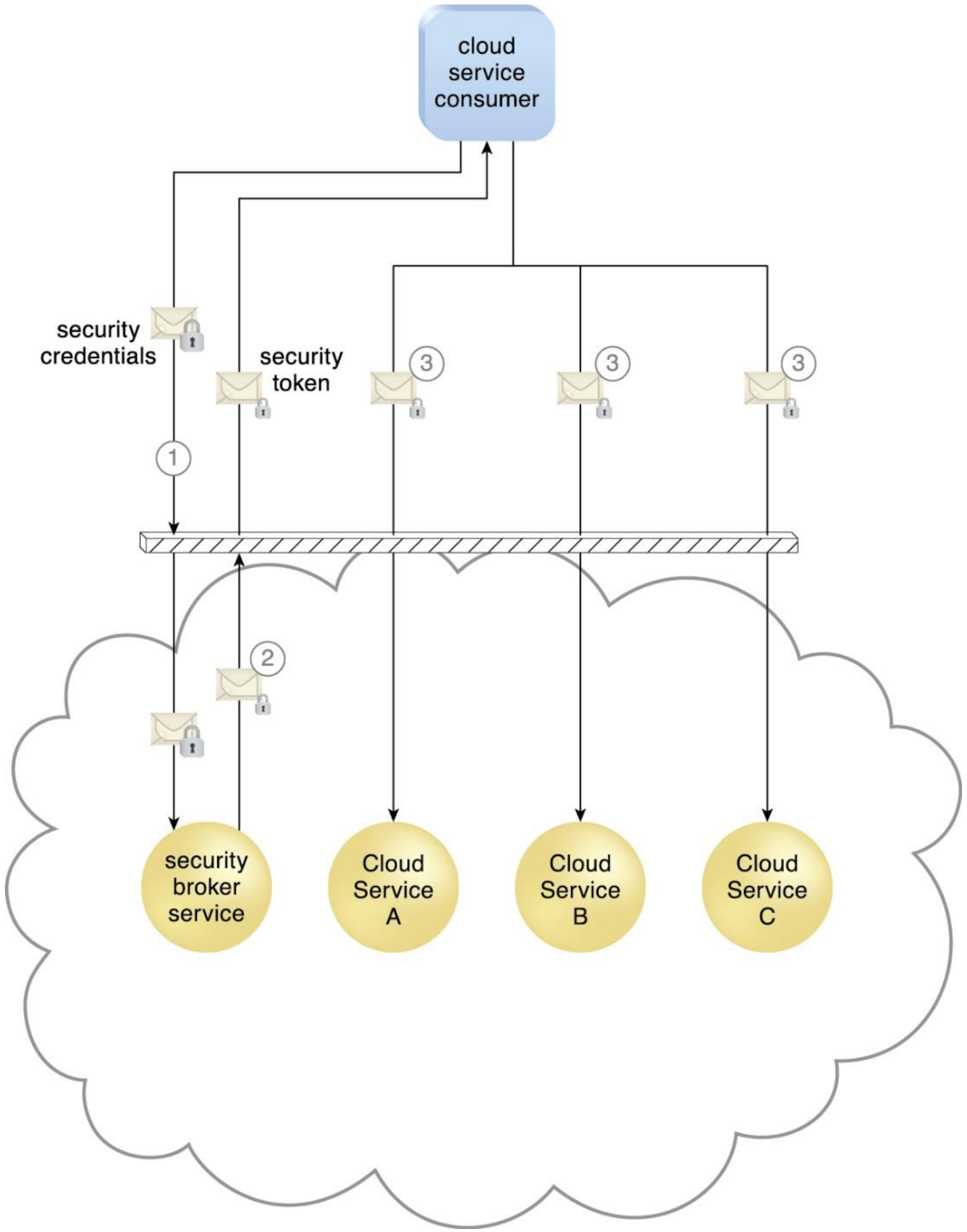


Figure 10.9 A cloud service consumer provides the security broker with login credentials (1). The security broker responds with an authentication token (message with small lock symbol) upon successful authentication, which contains cloud service consumer identity information (2) that is used to automatically authenticate the cloud service consumer across cloud Services A, B, and C (3).

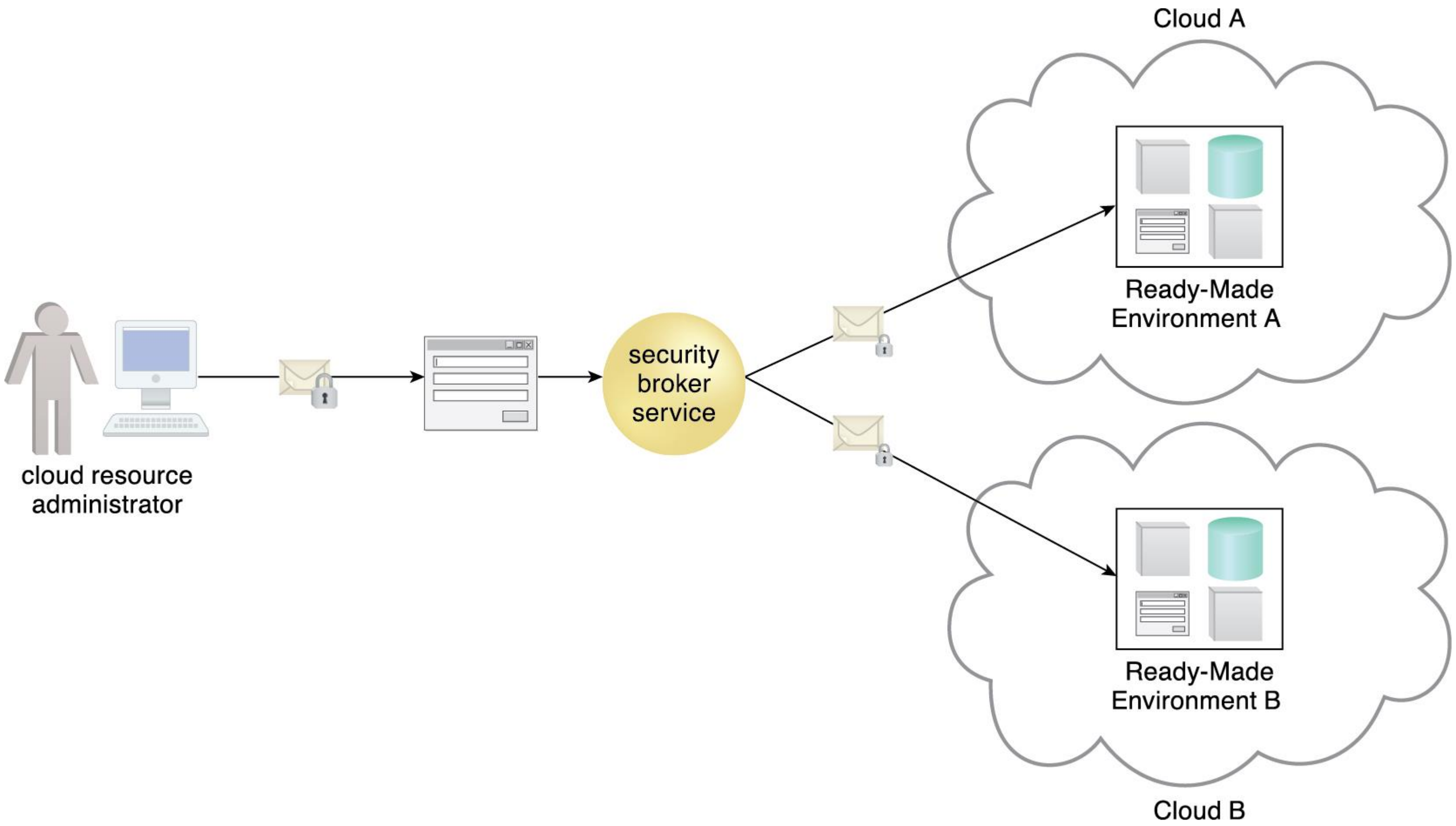


Figure 10.10 The credentials received by the security broker are propagated to ready-made environments across two different clouds. The security broker is responsible for selecting the appropriate security procedure with which to contact each cloud.

Cloud-Based Security Groups

- Cloud resource segmentation is a process by which separate physical and virtual IT environments are created for different users and groups.
- Resource segmentation is used to enable virtualization by allocating a variety of physical IT resources to VMs.
- Networks are segmented into logical cloud-based security groups that form logical network parameters.
- Cloud-based security groups delineate areas where different security measures can be applied.
- Can be used to help counter DoS, insufficient authorization, and overlapping trust boundaries threats.

Figure 10.11 Cloud-Based Security Group A encompasses Virtual Servers A and D and is assigned to Cloud Consumer A. Cloud-Based Security Group B is comprised of Virtual Servers B, C, and E and is assigned to Cloud Consumer B. If Cloud Service Consumer A's credentials are compromised, the attacker would only be able to access and damage the virtual servers in Cloud-Based Security Group A, thereby protecting Virtual Servers B, C, and E.

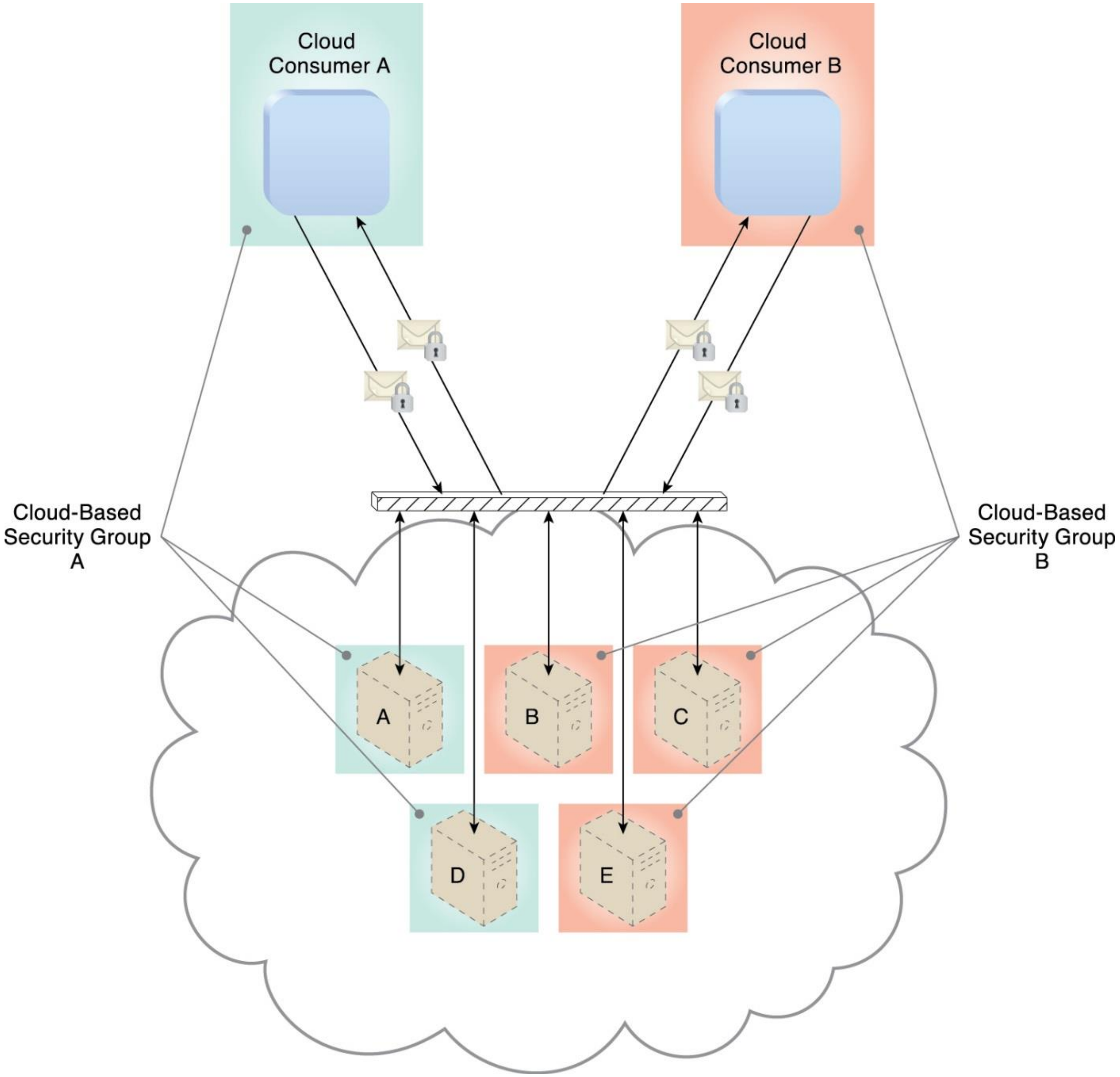
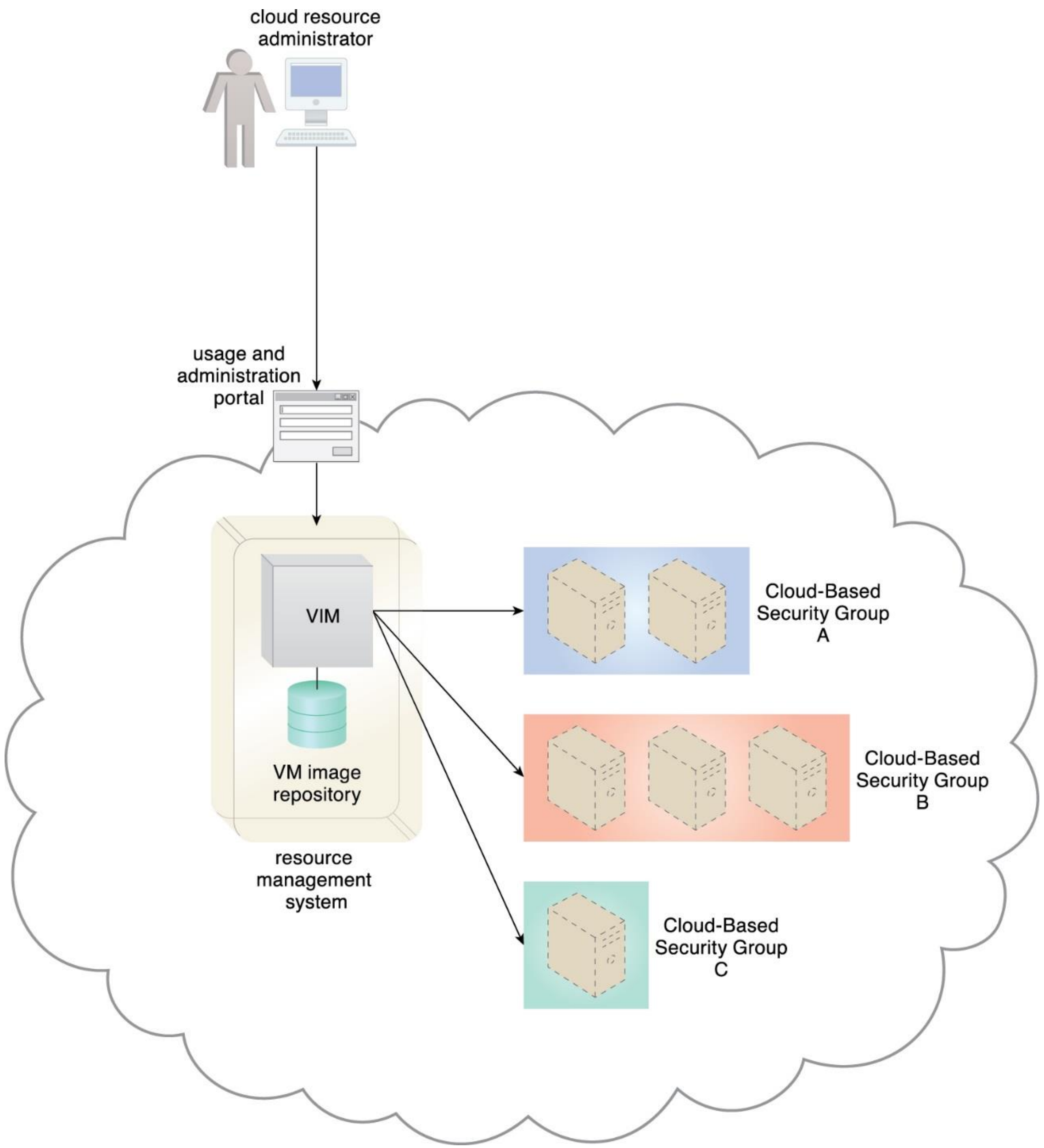


Figure 10.12 When an external cloud resource administrator accesses the Web portal to allocate a virtual server, the requested security credentials are assessed and mapped to an internal security policy that assigns a corresponding cloud-based security group to the new virtual server.


cloud resource administrator
usage and administration portal
VIM
VM image repository
resource management system
Cloud-Based Security Group A
Cloud-Based Security Group B
Cloud-Based Security Group C

Hardened Virtual Server Images

- Hardening is the process of stripping unnecessary software from a system to limit potential vulnerabilities that can be exploited by attackers.
- E.g.: Removing redundant programs, closing unnecessary ports, disabling unused services, internal root accounts, guest access.
- A hardened virtual server image is a template for virtual service instance creation that has been subjected to a hardening process.
- It helps to counter the DoS, insufficient authorization, and overlapping trust boundaries threats.

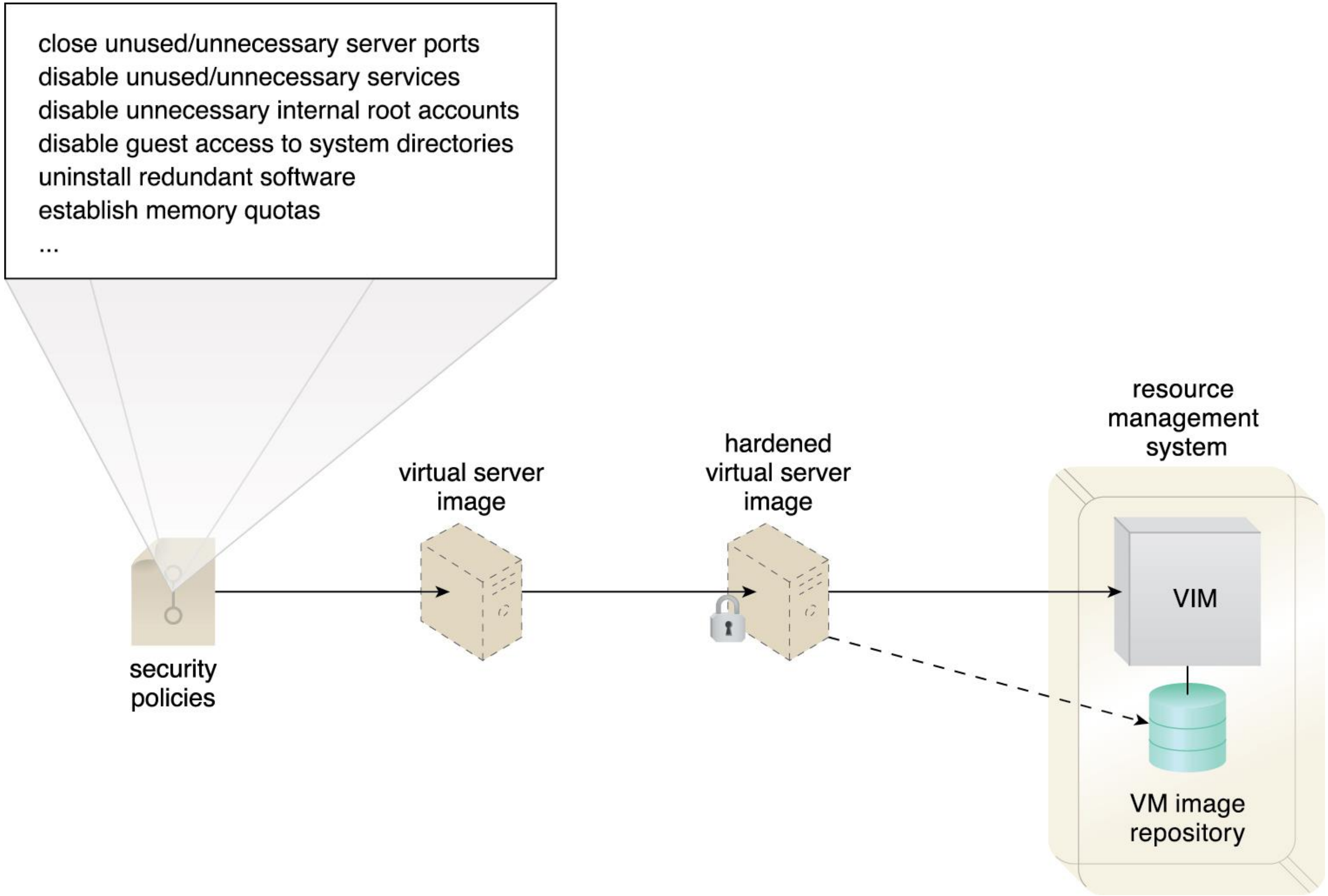
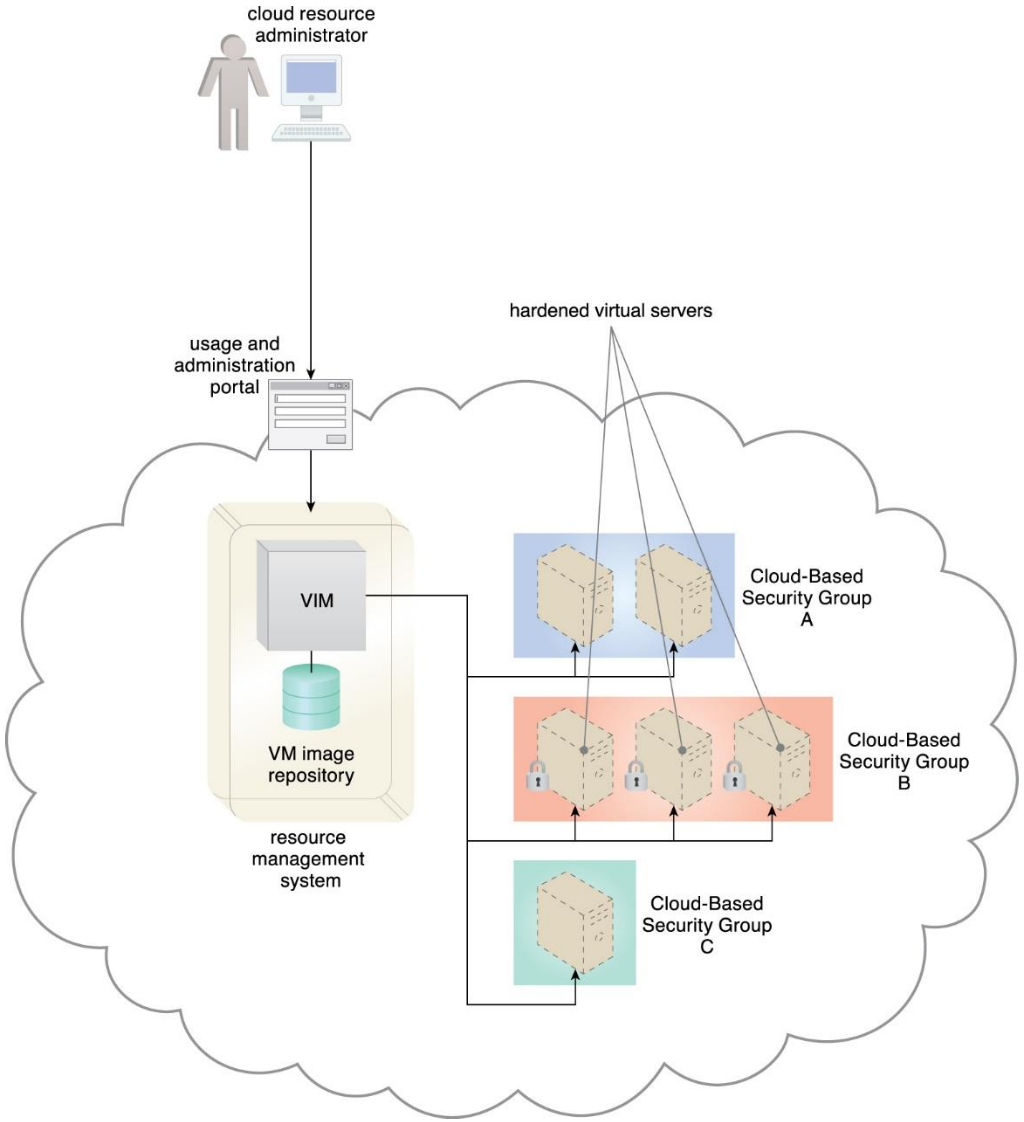

close unused/unnecessary server ports
disable unused/unnecessary services
disable unnecessary internal root accounts
disable guest access to system directories
uninstall redundant software
establish memory quotas
...
security policies
virtual server image
hardened virtual server image
resource management system
VIM
VM image repository

Figure 10.13 A cloud provider applies its security policies to harden its standard virtual server images. The hardened image template is saved in the VM images repository as part of a resource management system.

Hardened VM: Case Study Example

Figure 10.14 The cloud resource administrator chooses the hardened virtual server image option for the virtual servers provisioned for Cloud-Based Security Group B.


cloud resource administrator
usage and administration portal
hardened virtual servers
VIM
VM image repository
resource management system
Cloud-Based Security Group A
Cloud-Based Security Group B
Cloud-Based Security Group C

AWS Identity and Access Management (IAM)

AWS Identity and Access Management (IAM)

- Use **IAM** to manage access to **AWS resources** –
  - A resource is an entity in an AWS account that you can work with
  - Example resources; An Amazon EC2 instance or an Amazon S3 bucket

- *Example* – Control who can terminate Amazon EC2 instances

- Define fine-grained access rights –
  - **Who** can access the resource
  - **Which** resources can be accessed and what can the user do to the resource
  - **How** resources can be accessed

- IAM is a no-cost AWS account feature

AWS Identity and Access
Management
(IAM)

IAM: Essential components

IAM user

A **person** *or* **application** that can authenticate with an AWS account.

IAM group

A **collection of IAM users** that are granted identical authorization.

IAM policy

The document that defines **which resources can be accessed** and the **level of access** to each resource.

IAM role

Useful mechanism to grant a set of permissions for making AWS service requests.

Authenticate as an IAM user to gain access

When you define an **IAM user**, you select what ***types of access*** the user is permitted to use.

Programmatic access

- Authenticate using:
  - Access key ID
  - Secret access key
- Provides AWS CLI and AWS SDK access

AWS CLI

AWS Tools and SDKs

AWS Management Console access

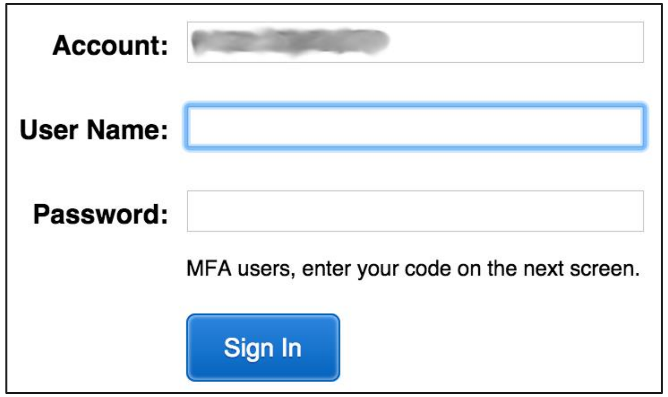
- Authenticate using:
  - 12-digit Account ID *or* alias
  - IAM user name
  - IAM password
- If enabled, **multi-factor authentication (MFA)** prompts for an authentication code.

AWS Management Console

IAM MFA

- MFA provides increased security.

- In addition to **username** and **password**, MFA requires a unique authentication code to access AWS services.

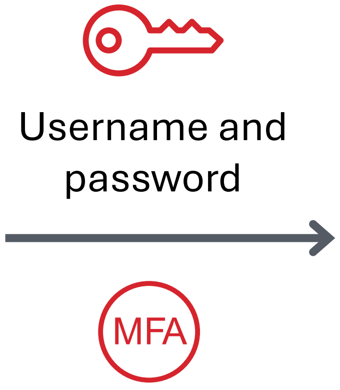


Account: [redacted]

User Name:

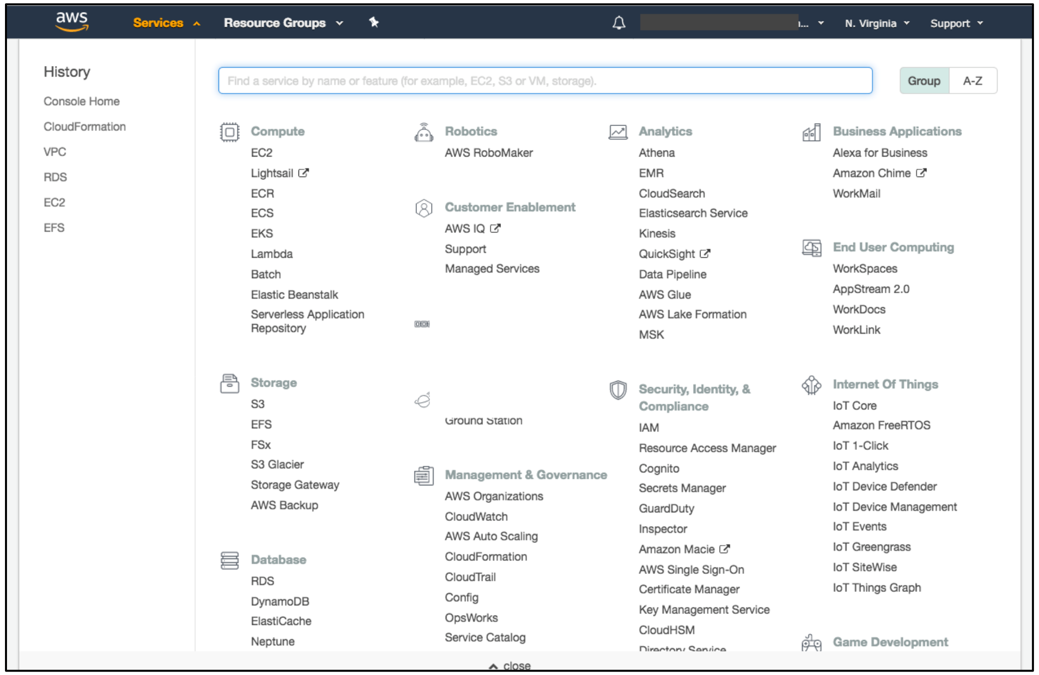
Password:

MFA users, enter your code on the next screen.



Username and password

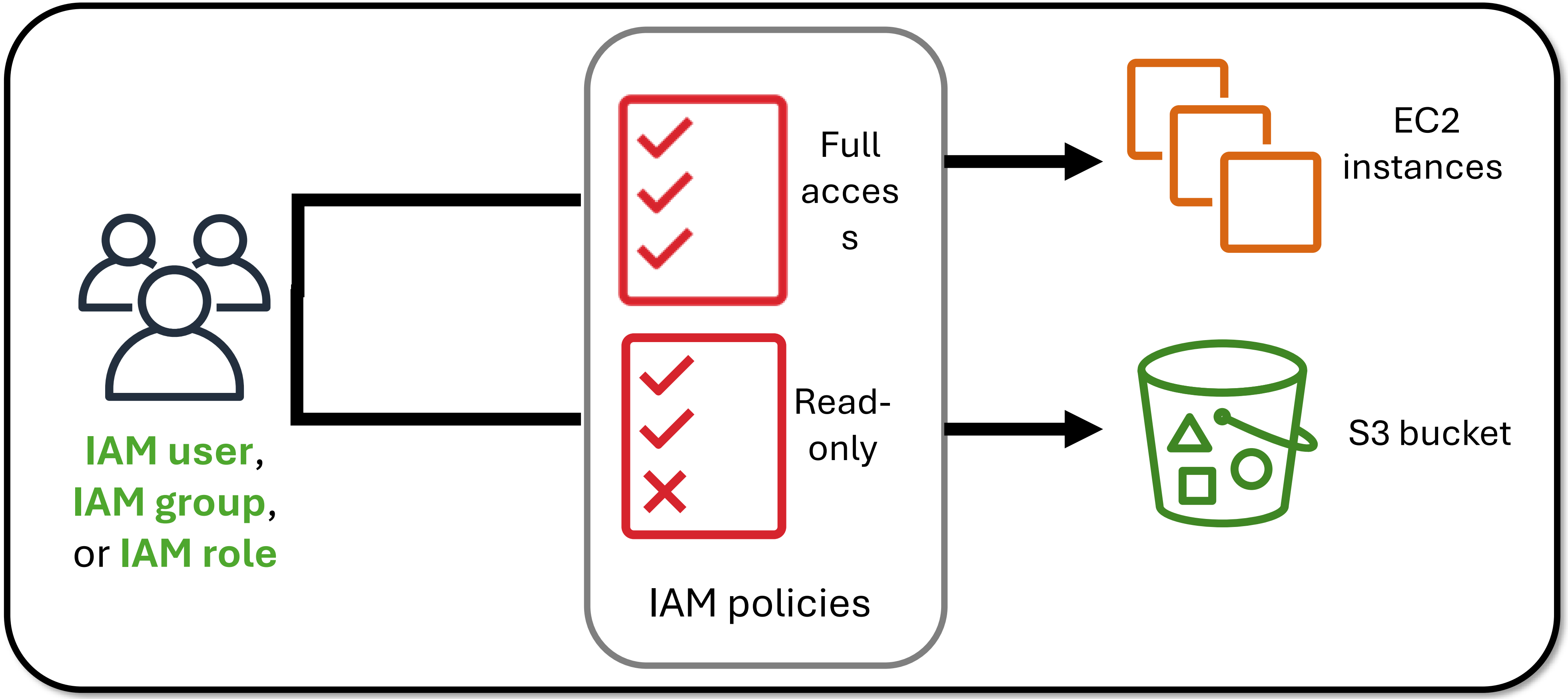
MFA token



AWS Management Console

Authorization: What actions are permitted

After the user or application is connected to the AWS account, what are they allowed to do?



IAM: Authorization

- Assign permissions by creating an IAM policy.

- Permissions determine **which resources and operations** are allowed:
  - All permissions are implicitly denied by default.
  - If something is explicitly denied, it is never allowed.

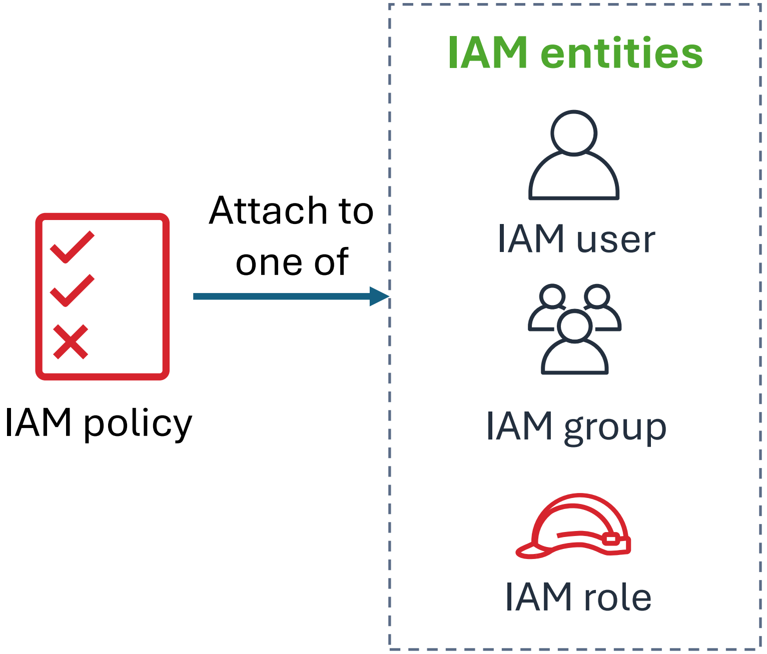
Best practice: Follow the **principle of least privilege**.

IAM permissions

Note: The scope of IAM service configurations is global. Settings apply across all AWS Regions.

IAM policies

- **An IAM policy is a document that defines permissions**
  - Enables fine-grained access control
- Two types of policies – *identity-based* and *resource-based*
- **Identity-based** policies –
  - Attach a policy to any IAM entity
    - An IAM user, an IAM group, or an IAM role
  - Policies specify:
    - Actions that ***may*** be performed by the entity
    - Actions that ***may not*** be performed by the entity
  - A single *policy* can be attached to multiple *entities*
  - A single *entity* can have multiple *policies* attached to it
- **Resource-based** policies
  - Attached to a resource (such as an S3 bucket)



IAM policy example

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect":"Allow",
    "Action":["DynamoDB:*","s3:*"],
    "Resource":[
      "arn:aws:dynamodb:region:account-number-without-hyphens:table/table-name",
      "arn:aws:s3:::bucket-name",
      "arn:aws:s3:::bucket-name/*"]
    },
    {
    "Effect":"Deny",
    "Action":["dynamodb:*","s3:*"],
    "NotResource":["arn:aws:dynamodb:region:account-number-without-hyphens:table/table-name”,
      "arn:aws:s3:::bucket-name",
      "arn:aws:s3:::bucket-name/*"]
    }
  ]
}
```

Explicit allow gives users access to a specific DynamoDB table and…

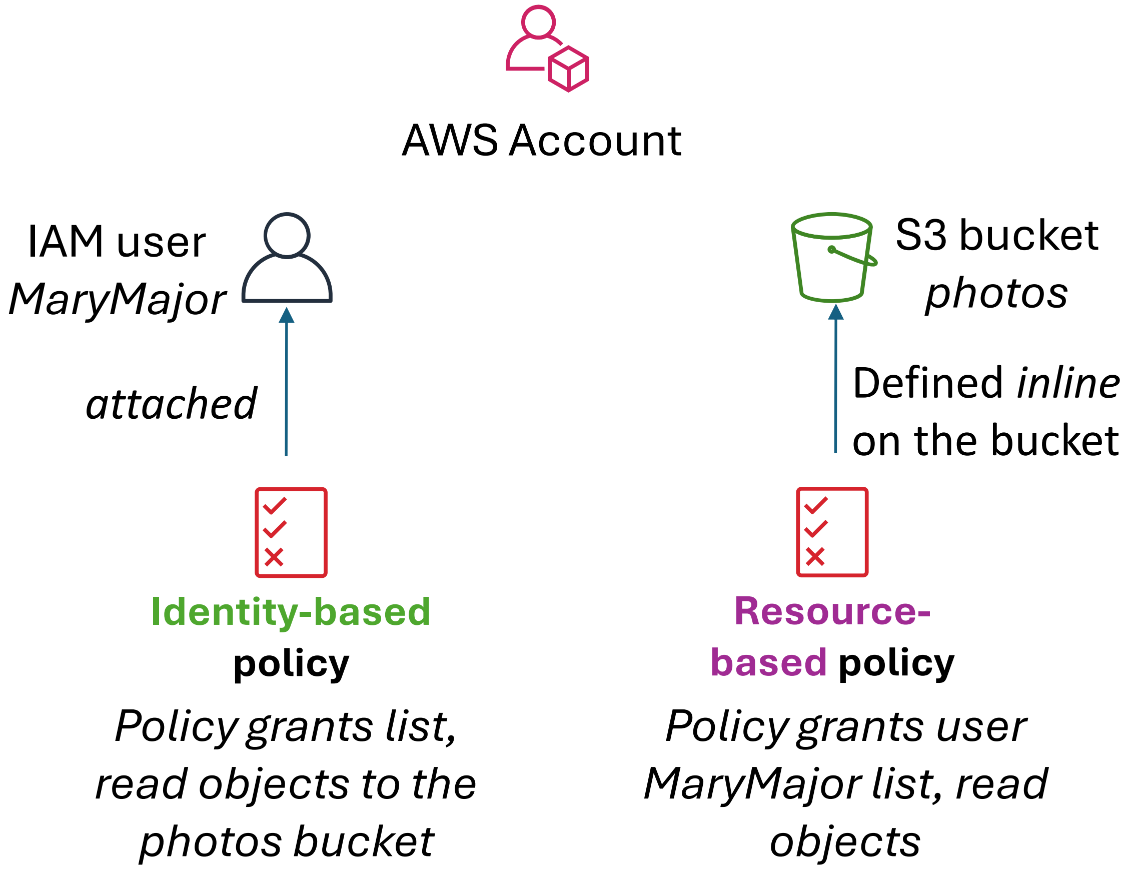
…Amazon S3 buckets.

Explicit deny ensures that the users cannot use any other AWS actions or resources other than that table and those buckets.

An explicit deny statement **takes precedence** over an allow statement.

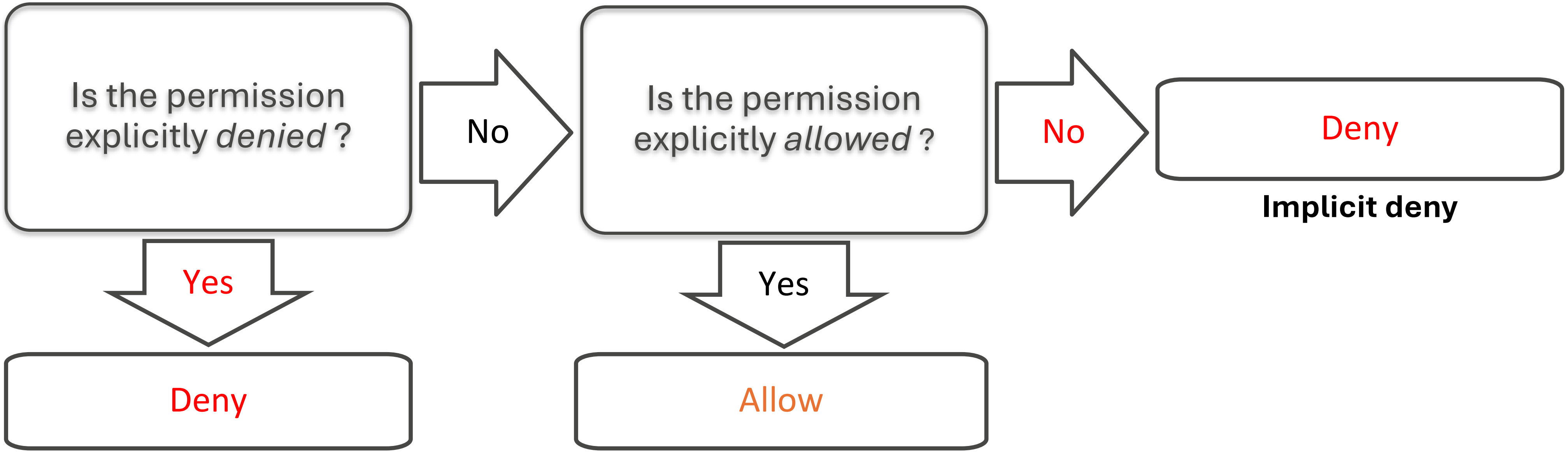
Resource-based policies

- *Identity-based policies* are attached to a user, group, or role
- **Resource-based policies** are attached to a resource (*not* to a user, group or role)
- Characteristics of resource-based policies –
  - Specifies who has access to the resource and what actions they can perform on it
  - The policies are *inline* only, not managed
- Resource-based policies are supported only by some AWS services


AWS Account
IAM user
MaryMajor
attached
S3 bucket
photos
Defined *inline* on the bucket
Identity-based policy
Policy grants list, read objects to the photos bucket
Resource-based policy
Policy grants user MaryMajor list, read objects

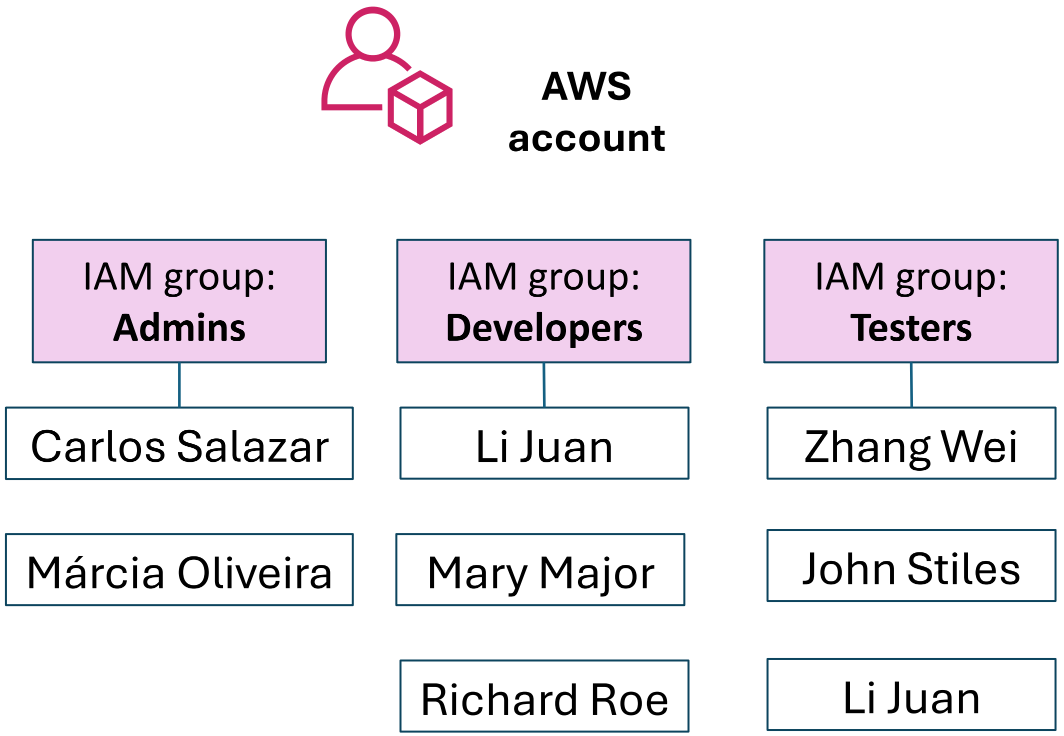
IAM permissions

How IAM determines permissions:


Is the permission explicitly *denied* ?
No
Is the permission explicitly *allowed* ?
No
Deny
Implicit deny
Yes
Deny
Yes
Allow

IAM groups

- An **IAM group** is a collection of IAM users
- A group is used to grant the same permissions to multiple users
  - Permissions granted by attaching IAM *policy* or policies to the group
- A user can belong to multiple groups
- There is no default group
- Groups cannot be nested



AWS account

IAM group: **Admins**

Carlos Salazar

Márcia Oliveira

IAM group: **Developers**

Li Juan

Mary Major

Richard Roe

IAM group: **Testers**

Zhang Wei

John Stiles

Li Juan

IAM roles

- An **IAM role** is an IAM identity with specific permissions
- Similar to an IAM user
  - Attach permissions policies to it
- Different from an IAM user
  - Not uniquely associated with one person
  - Intended to be *assumable* by a **person**, **application**, or **service**
- Role provides *temporary* security credentials
- Examples of how IAM roles are used to **delegate** access –
  - Used by an IAM user in the same AWS account as the role
  - Used by an AWS service—such as Amazon EC2—in the same account as the role
  - Used by an IAM user in a different AWS account than the role

IAM role

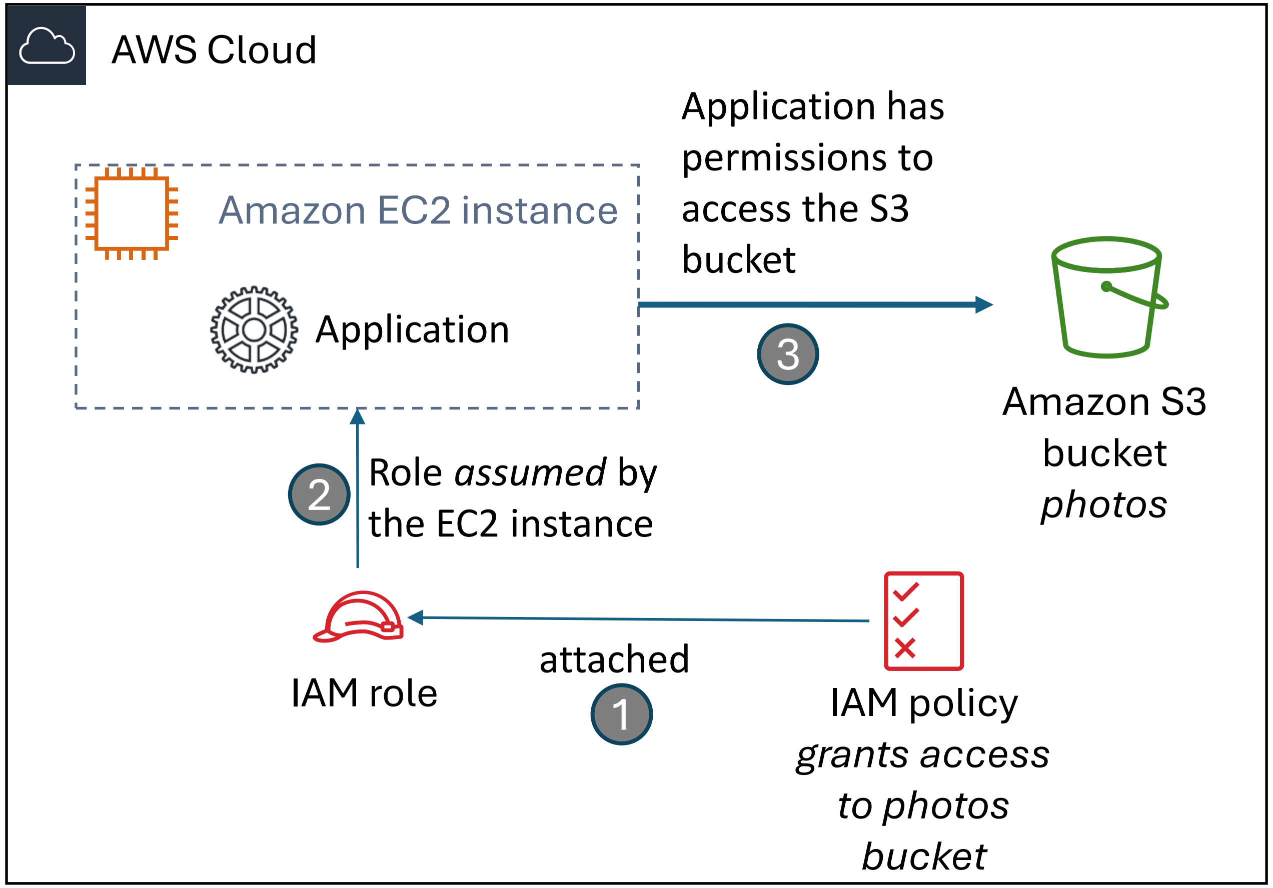
Example use of an IAM role

Scenario:

- An application that runs on an EC2 instance needs access to an S3 bucket

Solution:

- Define an IAM policy that grants access to the S3 bucket.
- Attach the policy to a role
- Allow the EC2 instance to assume the role


AWS Cloud
Amazon EC2 instance
Application
Application has permissions to access the S3 bucket
3
Amazon S3 bucket *photos*
2
Role *assumed* by the EC2 instance
IAM role
attached
1
IAM policy *grants access to photos bucket*