Sécurité dans les environnements infonuagiques

Module 2 : Gestion des identitiés et des accès (Partie 1)

Armstrong Foundjem

Polytechnique Montréal

Automne 2023

Plan

1. Concepts
2. Definition
3. IAM Architecture
4. Vulnerabilities

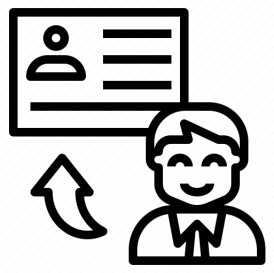
1. Concepts

2. Definition & Architecture

3. IAM vulnerabilities

Entity, Identity

- **Entity**
  - can be real: employees, contractors, customers, business partners, external parties
  - can be virtual: service, application
- **Identity**
  - an entity can have several identities
  - can be username, social security number, passport id, birthdate



Attribute, Resource, Trust

- **Attribute**
  - Each identity has many attributes that an entity can claim
  - can be biometric, location, role as an employee, age, and sex
- **Resource**
  - a resource is any object in the cloud owned by an organization
  - it can be files, S3 buckets, Serveless functions, EC2 instances
- **Trust**
  - It is the relationship $\mathcal{T}$ between the system S and entities E such that $S \times E \subseteq \mathcal{T}$

Sondage ☺: `https://www.wooclap.com/HTOMML`.

wooclap

Security Policy

1. A set of rules that must be satisfied by the system
2. Let $r_1, r_2, ..., r_n$ a finite set of condition rules that must be satisfied to achieved the policy $\mathcal{P}$

$$\frac{r_1, .., r_n}{P}$$

3. The system S satisfied the policy $\mathcal{P}$ when the

$$S \models \mathcal{P}$$

4. Given a Virtual Private Cloud (VPC) Network with an access control list ACL_{vpc} that accepts all ports. The policy

$$VPC \models (S{=}\{443/tcp, 22/tcp\} \lhd ACL_{vpc}) \land (ACL_{vpc} = S)$$

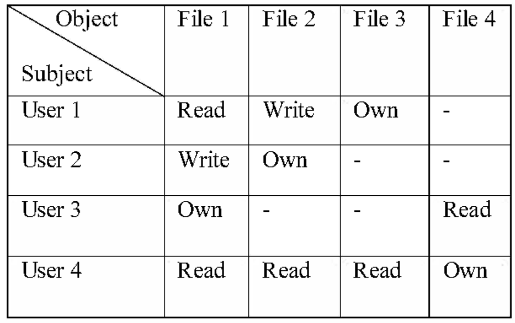
restricts network ports to HTTPS and SSH.

Access control

1. Entities: E
2. Resources/Objects: O
3. Right access: $f : E \times O \rightarrow P$
4. P is a set of permissions (e.g., $\{$read, write, owner$\}$)
5. Access control matrix: $(f_{e,o})_{(e,o)\in E\times O}$, with $f_{e,o} \subseteq P$
6. The number of associations to relate entities to permissions is $|E| \times |P|$
7. The number of entities and permission associations to authorize each entity in E for each permission in P is $|E| + |P|$
8. The access control is enforced when $|E| + |P| < |E|.|P|$, with $|P| > 2$

Access control

1. What are the values of $f_{user3,file4}$?



Object Subject	File 1	File 2	File 3	File 4
User 1	Read	Write	Own	-
User 2	Write	Own	-	-
User 3	Own	-	-	Read
User 4	Read	Read	Read	Own

2. Does File 3 is accessible by User 2 ?
3. Given a set of two files S={File 2, File 3}, what is the value of $f_{user2,S}$?

What are the types of access control ?

Sondage ☺: `https://www.wooclap.com/DQLZQE`.

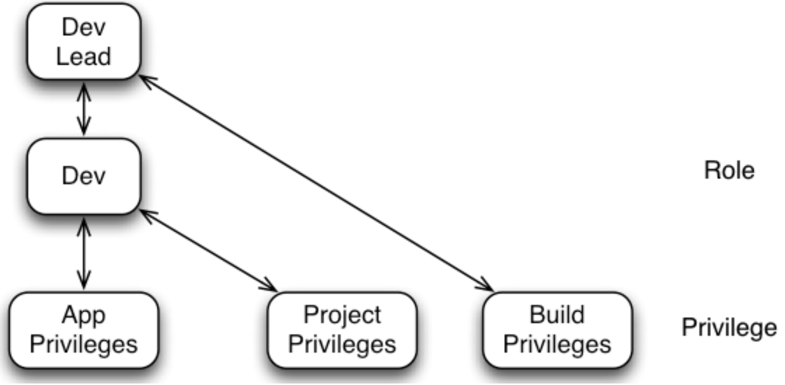
wooclap

Role-based Access Control (RBAC)

1. RBAC associates roles to entities and roles to permissions.
2. A permission can be assigned to a **group** of entities having the same role
3. Let R be the set of all roles in the organisation
4. The numbers of associations to relate roles to entities and roles to permissions are respectively $|R|.|E|$ and $|R|.|P|$
5. The number of entities and permission associations to authorize each entity in E for each permission in P and for each role in R is $|R|.(|E|+|P|)$
6. For a number of roles $n_r = |R|$ in the organization, the access control is enforced when

$$\sum_{i=1}^{n_r}(|E_i|+|P_i|) < \sum_{i=1}^{n_r}|E_i|.|P_i|$$

Role-based Access Control (RBAC)



Least privilege

- The principle allows only a minimum level of privilege or permissions required to do a given task
- The Least Privilege is the function : $E \times \mathcal{T} \rightarrow P$
  - that associates a **minimum set of permissions** $P_{min}^{Ext} \subseteq P$ to external entities $\text{Ext} \subseteq E$ that have the potential to compromise the system they extend.
  - Entities Ext have a set of tasks $T_{\text{Ext}} \subseteq \mathcal{T}$ to be done in the system.
- **Advantages**
  - limits the damage that can result from an accident or error
  - reduces the number of potential interactions among privileged programs
  - to the minimum for correct operation, so that unintentional, unwanted, or improper uses of privilege are less likely to occur

What are the limitations of Least Privilege ?

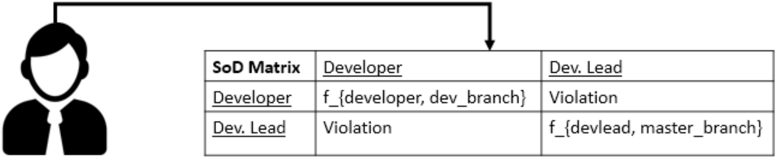
Sondage ☺: `https://www.wooclap.com/UYZADN`.

Separation of Duty (SoD)

- enforce policies to fix conflicting roles
- when an entity e is authorized with a given role r, the entity must be forbidden to get another role r'
- the role for which the entity is authorized is not mutually exclusive with any other role owned by the entity

$$\forall e \in E\ \forall r, r' \in R\ e \in f_{e,r} \wedge e \in f_{e,r'} \implies (r, r') \notin ME$$

- with $ME \subseteq R \times R$, a set of mutual exclusive role pairs



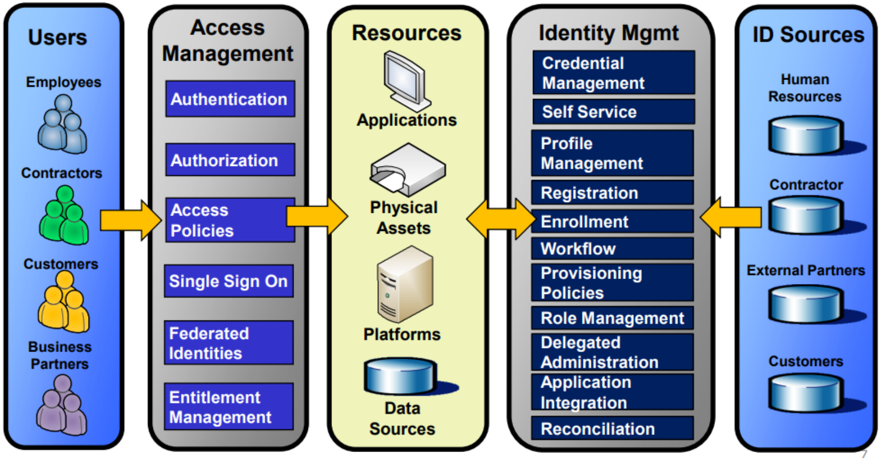
SoD Matrix	Developer	Dev. Lead
Developer	f_{developer, dev_branch}	Violation
Dev. Lead	Violation	f_{devlead, master_branch}

1 Concepts

2 Definition & Architecture

3 IAM vulnerabilities

What is Identity and Access Management ?



Credits: Crowe

What is Identity and Access Management ?

- **Provisioning/Deprovisioning** of User Identities
  - manage user accounts (creation, modification, revocation) following security policies
  - Deactivate user accounts when access to resources is revoked or no longer applicable
- **Authentication**
  - what you know, e.g., password, passphrase
  - what you have, e.g. token
  - what you are, fingerprint, location
  - both, with multi-factor authentication
  - federated authentications (e.g., Single-Sign-On)
- **Authorization**
  - the strategy to allow specific actions to be execute by entities
  - can be security policies to grant access to resources (e.g., S3, EC2 Instance, VPC)
  - or assigning roles to entities that grant the permissions

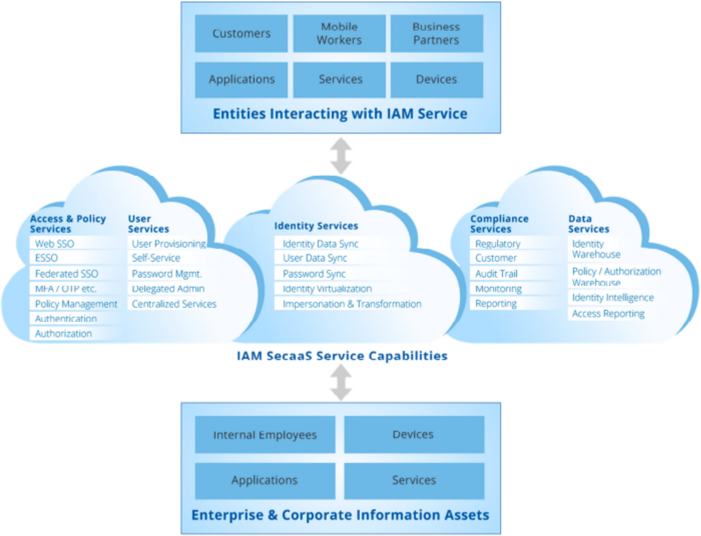
What is Identity and Access Management ?

- API access via authorization key (e.g., Bearer/Basic token) and a private key

- **Access policies**
  - use Role-based Access Control (RBAC)
  - enforce access control rules using standards such as XACML
- **Federated Single Sign On (SSO)**
  - allow access of multiple applications requiring authentication by passing a single credential
  - based on standards such as Open ID, WS-Fed, SAML
  - enable to federate identities between entities, identity providers, and service providers
  - are of different types: internal SSO on-prem, inbound SSO for service providers, and outbound SSO for external partners
- **Entitlement Management**
  - Directory services based using LDAP protocol for user authen.
  - Audit and Reporting (e.g., Tamper proof, logging)

Sondage ☺: `https://www.wooclap.com/KYFTOE`.

IAM Architecture: Cloud Security Alliance (CSA)



Customers

Mobile Workers

Business Partners

Applications

Services

Devices

Entities Interacting with IAM Service

Access & Policy Services
- Web SSO
- ESSO
- Federated SSO
- MFA / OTP etc.
- Policy Management
- Authentication
- Authorization

User Services
- User Provisioning
- Self-Service
- Password Mgmt.
- Delegated Admin
- Centralized Services

Identity Services
- Identity Data Sync
- User Data Sync
- Password Sync
- Identity Virtualization
- Impersonation & Transformation

Compliance Services
- Regulatory
- Customer
- Audit Trail
- Monitoring
- Reporting

Data Services
- Identity Warehouse
- Policy / Authorization Warehouse
- Identity Intelligence
- Access Reporting

IAM SecaaS Service Capabilities

Internal Employees

Devices

Applications

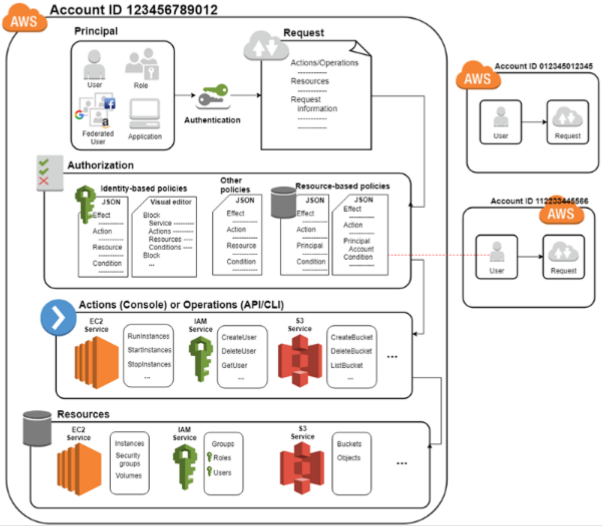
Services

Enterprise & Corporate Information Assets

IAM Architecture: Cloud Security Alliance (CSA)

- **What is the difference between Web SSO and Federated SSO ?**
- **What are the common centralized services to manage user accesses ?**
- CSA architecture supports *Identity virtualization* that
  - allows abstraction of multiple identity services (e.g., LDAP services, federated identities)
  - allows a local and global view of aggregated/correlated identities
  - often coupled with contexts such as geographic location for data enrichment
  - contexts improve identity queries by selecting the identity provider nearest to the user/entity

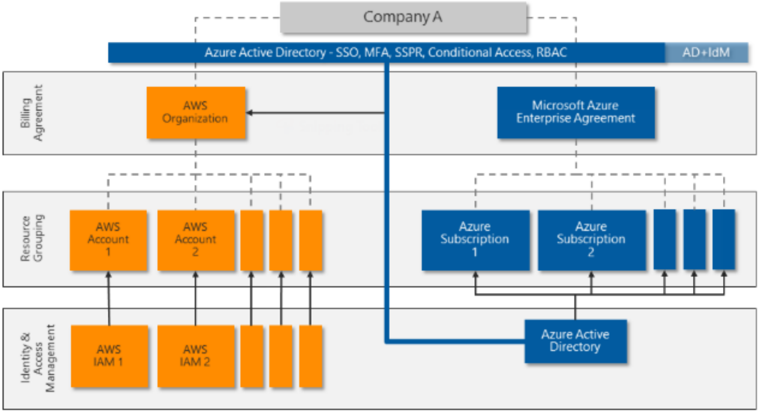
IAM Architecture: AWS



IAM Architecture: AWS

- In AWS, authorization to access resources is managed using security policies
- Security policies contains the following components:
  - **Principal**. Account ID(s) or name(s) of the user(s) authorized to access resources e.g., root, 1224455994
  - **Effect**. Permissions to access resources: *Allow*, *Deny*
  - **Action**. The allowed (resp. denied) operation(s) that user(s) can execute (resp. can not execute). Operations can be *CreateBucket, DeleteUser, or RunInstances*
  - **Resource**. The service(s) targeted by users e.g. S3 service, EC2 service, VPC service, IAM service
  - **Conditions**. They are used to apply more restrictions on users and resources e.g. *aws:username* = *test*, *aws:ResourceTag:EC2* = *testserver*
- **What are the different types of IAM policies available in AWS ?**

IAM Architecture: Multi-cloud



Company A

Azure Active Directory - SSO, MFA, SSPR, Conditional Access, RBAC

AD+IdM

Billing Agreement

AWS Organization

Microsoft Azure Enterprise Agreement

Resource Grouping

AWS Account 1

AWS Account 2

Azure Subscription 1

Azure Subscription 2

Identity & Access Management

AWS IAM 1

AWS IAM 2

Azure Active Directory

Credits: Microsoft

IAM Architecture: Multi-cloud

- Frontline
  - Conditional access is similar to conditions in security policies for strict resource and user management
  - Self-service Password Reset (SSPR) allows users to modify their password
  - SSO, Multi factor authentication, RBAC
- Federated SSO and LDAP Directory services allow centralized management of user identities from different service providers
  - avoiding manage multiple identities and passwords
  - across multiple organizations and from different locations
- Multi-cloud IAM controls role delegation for **Just-in-Time access** to specific resources across different service providers
  - For example, an *Admin* role is assigned to an employee with role *User* to do a specific task
  - the *Admin* role is automatically revoked after task is done
  - and the role *User* is reassigned to the employee

Sondage ☺: `https://www.wooclap.com/GNZPBU`.

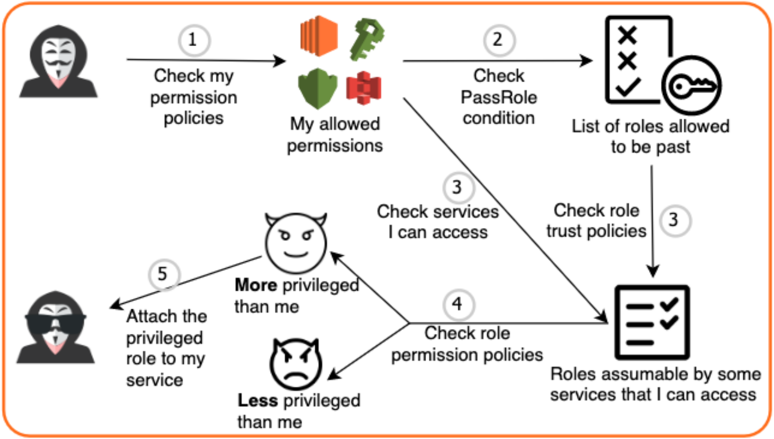
wooclap

1. Concepts
2. Definition & Architecture
3. IAM vulnerabilities

IAM vulnerabilities: OWASP Cloud

- **Broken Access Control**
  - access to API with missing access controls for POST, PUT and DELETE
  - privilege elevation by acting as administrator with a user role
  - modification of the URL parameters or force browsing, the HTML page, or injecting API requests to bypass access controls
  - no deny by default or violation of the principle of least privilege
  - manipulation of metadata such as cookie, access token
- **Identification and Authentication failures**
  - brute-force/automated attacks is allowed during logging/authentication
  - authentication with default, weak, or well-known passwords
  - missing or ineffective multi-factor authentication
  - session identifier exposed via URL or reused after login
  - manipulation of metadata such as cookie, access token

Broken Access Control: Palo Alto Net. Unit42 Case Study

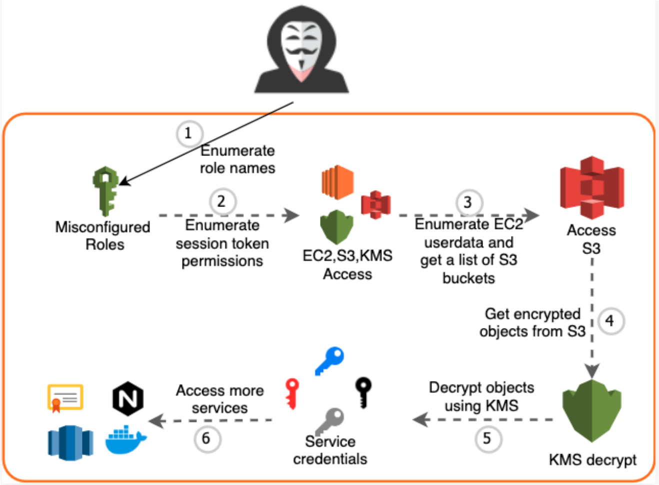

1
Check my permission policies
My allowed permissions
2
Check PassRole condition
List of roles allowed to be past
3
Check services I can access
Check role trust policies
3
Roles assumable by some services that I can access
4
Check role permission policies
More privileged than me
Less privileged than me
5
Attach the privileged role to my service

Credits: Palo Alto Networks

IAM vulnerabilities: OWASP Cloud

- **Insecure configuration**
  - improper permissions set on resources (buckets, EC2 instance, VPC, ...)
  - principal or resource fields in security policies are configured with "*" to grant access to any user or any resource
  - root profile is used by default
  - over-permissive IAM role configuration
- **Over-permissive/insecure network policies**
  - access control list accepts all the inbound traffic
  - missing or mis-configured policies to restrict access to subnets
- **Ineffective logging & monitoring**
  - missing or mis-configured policies to allow cloud trails or logging for IAM changes
  - no container or instance process activity monitoring

Insecure Misconfig.: Palo Alto Net. Unit42 Case Study


1
Enumerate role names
Misconfigured Roles
2
Enumerate session token permissions
EC2,S3,KMS Access
3
Enumerate EC2 userdata and get a list of S3 buckets
Access S3
Get encrypted objects from S3
4
KMS decrypt
Decrypt objects using KMS
5
Service credentials
Access more services
6

Credits: Palo Alto Networks

Sondage ☺: `https://www.wooclap.com/CKKEBP`.

wooclap