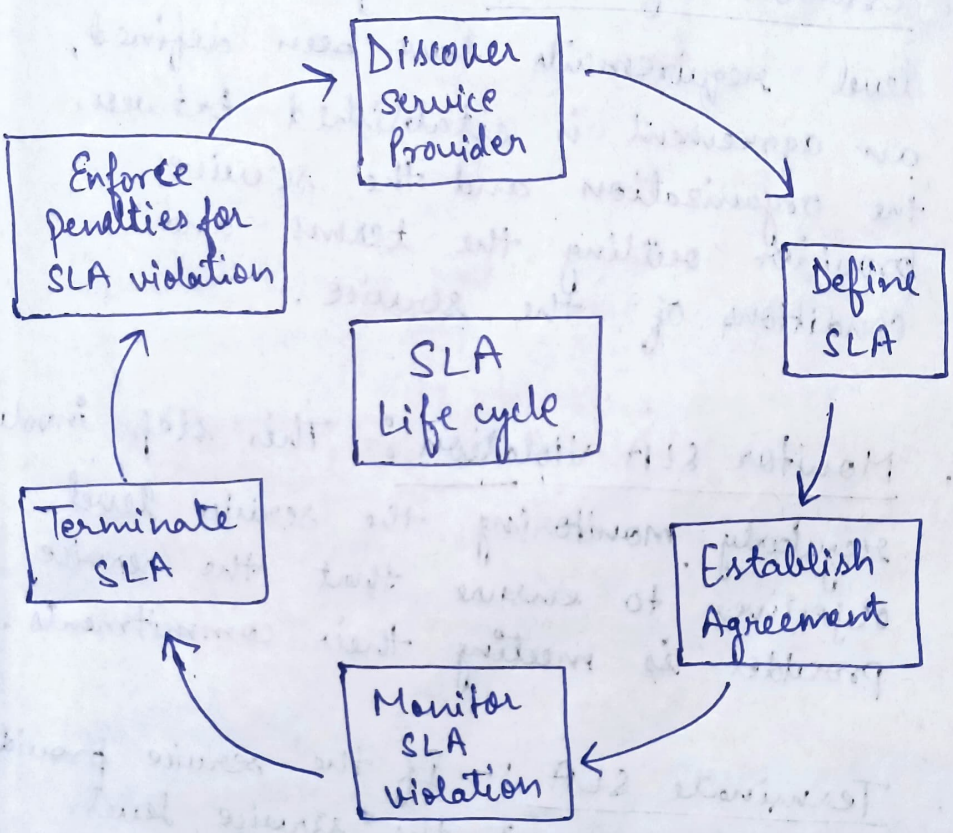
Tut - 1

1.) Draw and explain the diagram representing the SLA life cycle. Compare SLA management strategies of AWS, Azure and Google Cloud.

SLA Life Cycle



1. **Discover Service Provider** :- This step involves identifying a service provider that can meet the needs of the organization and has the capability to provide the required service. This can be done through research, requesting proposals, or reaching out to vendors.

2. Define SLA :- In this step, the service level requirement are defined and agreed upon between the service provider and the organization. This includes defining the service level objectives, metrics, and targets that will be used to measure the performance of the service provider.

3. Establish Agreement :- After the service level requirements have been defined, an agreement is established between the organization and the service provider outling the terms and conditions of the service.

4. Monitor SLA Violation :- This step involves regularly monitoring the service level objectives to ensure that the service provider is meeting their commitments.

5. Terminate SLA :- If the service provider is unable to meet the service level objectives, or if the organization is not satisfied with the service provided, the SLA can be terminated.

6. Enforce Penalties for SLA voilation :- If the service provider is found to be in voilation of the SLA, penalities can be imposed as outlined in the agreement.

Comparison of SLA Management Strategies

(1) Amazon AWS

→ Defines SLA for each service (eg:- EC2, S3, RDS) seperately.

→ Monitoring tools used AWS cloud watch, Cloud Trail, and Trusted Advisor.

→ Provides extensive API's and event triggers via AWS Lambda and SNS for automated SLA breach ~~alters~~ alerts.

(2) Microsoft Azure

→ Provides service based SLA with uptime varying by redundancy (eg:- 99.9%, 99.95%).

→ Monitoring tools used Azure monitor, Applications insights, and service health for SLA.

→ Integrates with Azure automation and Logic Apps for auto remediation workflows.

(3) Google Cloud

→ Offers detailed SLA's per service (compute engine, cloud storage) with region based differentation.

→ Monitoring tools used cloud monitoring, and cloud operations suite for SLA visibility.

→ Supports automated responses through cloud functions and Pub/Sub integration.

2.) How do intrusion detection system (IDS) and intrusion prevention systems (IPS) work? Discuss how configuration drift can affect host security.

Intrusion detection systems (IDS)

An IDS is a passive security mechanism that monitors network or system traffic for suspecious behaviour and alerts administrators of potential threats.

Working Mechanism

→ IDS typically operates out-of-band, meaning it observes data without interfering with normal traffic flow :-

Types of IDS :-

→ Network based IDS (NIDS)
→ Host based IDS (HIDS)

Detection Methods :-

→ Signature based detection.
→ Anamaly based detection.
→ Hybrid detection.

Intrusion Prevention Systems (IPS)

An IPS is an active protection mechanism that not only detects but also prevents or blocks malicious activities in real time.

Working Mechanism

→ IPS is placed inline (directly in the traffic path).

→ Upon identifying malicious behaviour, it can drop packets, reset connections, or quarantine hosts automatically.

→ Configuration drift refers to the gradual deviation of a system's configuration from its baseline or desired state over time.

→ This occurs when unauthorized, manual, or untracked changes are made to system parameters, software versions, firewalls rules, or policies.

Causes of Configuration drift :-

→ Manual configuration changes by administrator.

→ Inconsistent patch management or updates.

→ Misalignment between production and test environments.

→ Software version mismatches due to automation failures.

Tut - 2

1.) Discuss the key data privacy and security issues in cloud computing. Discuss how public key infrastructure (PKI) supports authentication.

Key data privacy and security issues in cloud computing are :-

1) Data Confidentiality.
2) Data Integrity.
3) Data Availability
4) Multi - Tenancy
5) Data Location and Jurisdiction.
6) Insecure API's and Interfaces.
7) Lack of Visibility and control
8) Insider Threats.
9) Virtualization Vulnerabilities.
10) Data Remanence

How PKI Supports Authentication :

1. **Identity Verification :-**
   → When a user or system requests access to a cloud service, it presents a digital certificate issued by a trusted CA.

2. **Digital Signature :-**
   → PKI allows user to digitally sign documents or transactions.

3. **Mutual Authentication :-**
   → Both client and server authenticate each other (eg:- SSL/TLS handshake in HTTPS).
   → Prevents Man-In-The-Middle Attack.

4. **Secure Key Exchange :-**
   → PKI faciliates secure exchange of symmetric keys used for encryption in communication sessions.

5. **Cloud Usecase Example :-**
   → When a user accesses AWS Management Console :-
      → AWS uses TLS certificate (PKI-based) to authenticate its server to the user.
      → User can use X.509 certificates for API authentication in AWS IOT or mutual TLS setups.

2.) What is risk based authentication? What are the differences between authentication, authorization and accounting in IAM?

Risk based authentication (RBA) also known as adaptive authentication is an intelligent security mechanism that dynamically adjusts the level of authentication required based on the perceived risk of a login attempt or transaction.

Unlike static multi-factor-authentication (MFA), which always requires fixed factors (eg:- password + OTP), RBA evaluates contextual parameters in real time to decide whether additional verification is needed.

(1) Authentication :-

Authentication is the process of verifying the identity of a user, device, or process attempting to access a system.

Purpose :-

To ensure that an entity is who they claim to be.

Methods :-

→ Knowledge - based :- Passwords, PIN's.
→ Possession - based :- Smart cards, OTP tokens.
→ Inherence - based :- Biometrices.
→ Context - based :- Risk based authentication

Example :-

When you enter your username and password to log into AWS or Gmail.

(2) ~~Authentication~~ :- Authorization :-

Authorization occurs after authentication and determines what actions or resources that authenticated user is allowed to access.

Purpose :-

To enforce access control policies based on roles, privileges, or policies.

Models :-

→ Role - Based Access Control (RBAC)
→ Attribute - Based Access Control (ABAC)
→ Policy - Based Access Control (PBAC)

Example :-

→ A system administrator can create and delete users.
→ A regular user can only view their own data.

③ Accounting :-

Accounting sometimes called auditing or accountability is the process of tracking user actions and maintaining logs for monitoring, compliance, and forensic analysis.

Purpose :-

To provide traceability, usage tracking and non-repudiation.

Data Captured :-

→ Who accessed the system?
→ What actions were performed?
→ When and from where?
→ Was the action successful or failed?

Example :-

Logging details of a user's SSH session or database transactions for auditing purposes.

Tut-3

1.) What are the advantages of cloud simulation tools? Compare CloudSim with other open source simulators like GreenCloud and iCanCloud.

→ Advantages of cloud simulation tools:-

1) Cost-effective Research and Testing.
2) Risk-free Experimentation.
3) Scalability Testing.
4) Resource Optimization Studies.
5) Reproducibility.
6) Time efficiency.
7) Flexibility and Customization.
8) Education and Training.

→ Comparison of CloudSim, GreenCloud, iCanCloud.

(A) CloudSim

Developer:- CLOUDS Laboratory, University of Melbourne.

Language:- Java.

Main focus:- Simulation of resource provisioning, VM scheduling, and cloud infrastructure management

Key features:-

→ Models data centers, hosts, VMs.
→ Supports simulation of IaaS, PaaS, SaaS.
→ Highly modular and extensible for research use.

(B) GreenCloud

Developer:- University of Luxembourg

Language:- C++ with NS2

Main focus:- Energy efficient cloud data center networking and communication modeling.

Key features:-

→ Integrates with NS2 for detailed packet level network simulation.

→ Focuses on energy consumption analysis at server, switch, and communication levels.

→ Model power management policies for data centers.

→ Provides fine-grained visibility into network traffic and energy costs.

(c) iCanCloud

Developer :- University of Barcelona

Language :- C++ (built on OMNeT++ framework)

Main Focus : Modeling of Cloud storage and cost based simulation for public clouds like AWS EC2.

Key features :-

→ Simulates real cloud configurations
→ Supports both private and public cloud modeling.
→ Provides detailed performance and cost evaluation for user-defined scenarios.
→ Highly modular GUI for experiment setup and visualization.
→ Allows integration of cloud pricing data for cost analysis.

2.) What are the best practices for securing APIs in cloud ~~pla~~ applications? Describe common vulnerabilities such as SQL injection, cross-site scripting (XSS), and how to prevent them.

→ Best practices for securing cloud APIs

1) Use strong Authentication and Authorizatio
2) Enforce HTTPS/TLS Encryption.
3) Validate and Sanitize Input data.
4) Implement Rate Limiting and Throttling.
5) Uses API Gateways.
6) Apply Least Privilege principle.
7) Regular Security Testing.
8) Use API Keys and Secrets Securely.
9) Enable Logging and Monitoring.
10) Handle Errors Securely.
11) Protect Against Cross Origin Attacks (CORS).
12) Keep dependencies updated.

→ SQL Injection (SQLi)

SQL injection is a code injection technique in which an attacker manipulates input fields to execute malicious SQL queries directly on the database.

It exploits insecure input validation in API endpoints that interact with databases.

Impact :-

→ Unauthorized data access or modification.
→ Data leakage or deletion.
→ Complete database compromise.

Prevention Techniques :-

→ Parameterized Queries.
→ Input validation and sanitization.
→ Use ORM frameworks.
→ Least privilege databases access.
→ Regular Security Scanning.

→ Cross-site Scripting (XSS)

Cross-site scripting (XSS) occurs when attacker inject malicious javascript code into web pages viewed by other users.

In Cloud APIs, XSS often occurs when API response include unsanitized user input displayed on client interface.

Types of XSS :-

→ Reflected XSS
→ Stored XSS
→ DOM-based XSS

Impact :-

→ Session hijacking & cookie theft.
→ Defacement of web pages.
→ Phishing or malicious redirection.

Prevention Techniques :-

→ Input and Output Encoding.
→ Use Secure frameworks.
→ Content Security Policy (CSP)
→ Avoid Eval and Inner HTML.
→ Sanitize inputs at both ends.