SafeKeep

# Amazon Web Services (AWS) Test Cases

## Attack Surfaces

Access Level Controls
Misconfigured ports, firewall rules, network policies etc.
Over Permissive Roles and Access to Servers, Storage etc.
Logging, Monitoring, Alerts and Back-up

## ✓ Identity & Access Management ✓

There should be no active keys for the root account
Root account should not be utilized for day to day tasks
Multi-Factor Authentication should be enabled for each user, including root
Service users (for example, for continuous integration and continuous deployment) should have only programmatic access
All users should have only one active access key and it should be rotated at least every 180 days
There should be no unused security groups
Password policies should be enhanced for each user with access to AWS console
Change all SSH and PGP keys regularly
Remove all unused security accounts
Use Access Analyzer to look for unintended access. You might see roles that have the unwanted ability to be assumed by outside entities

## ✓ S3 Storage ✓

Permission to list, get, put ,delete, and manage data should be enabled only for specific users
Bucket versioning and access logging should be enabled
Granted permissions should be configured for specific user(s), not for everyone (such as “*” for identity or action)
Verify if the bucket is publicly accessable
Verify if principles of least privilege access is enabled, read and write only are well defined for access control
Verify if the data is encrypted at rest and there is enforced encryption of data in transit
Verify if there is a backup and restore configuration put in place
Verify if there are no unused buckets adding up the cost
Verify if the S3 do not have very common names to prevent enumeration from outside sources

## ✓ Logging, Monitoring, and Backup ✓

Identifying and validating tasks part of logging and monitoring
Logs are fed to a secure location and are backed up
Different label of logs are present depending on the service triggering log generation
Automated filtering of logs and generation of alerts followed by sending them as notifications to concerned security team members
Verification of the log integrity safeguards

# Amazon Web Services (AWS) Test Cases

## ✓ EC2 Service (Elastic Cloud Computing) ✓

There should be no default security groups in use with default configuration
Verify if only least permissive rights are implement for the security group(s)
Only allowed ports should be opened to everyone or set of users
Perform a Pen Test on the deployed application such as web app and its APIs
Verify the third party components (if used) for any of the apps such as web app
There should be a description for the usage of each opened port/port range
All white listed Ips should be known and have a description
Verify the access configuration, IAM users, IAM roles
Verify if regular updates, patches are put in place for the OS and the used applications
Verify if there are backup and restore configurations in place

## ✓ API Management ✓

Verify the APIs which are enabled are indeed being utilized
Verify the configuration of the enabled APIs
Verify if proper logging is put in-place for API activities

## ✓ Creating, Storing, and Utilization of Confidential Elements (Certificates, Keys, Passwords etc.) ✓

Verify if the client is aware of all the keys that are being used, on AWS, and hybrid
Verify the IAM for such files
Verify what mechanisms/algorithms/encryption methods are used to create these files and if the entropy is high
Verify if these files are being rotated after a fixed number of days and properly replaced with new ones
Verify the level of security where all the private keys, certificates are being stored and used
Verify that upon deletion of these files, there is no residual data present
Analyze with what process the public keys, certificates are being circulated
Verify that the integrity of confidential files are intact
Verify if proper alerts are generated when these files are being used/called and upon editing/replacing

## ✓ VPC (Virtual Private Cloud) ✓

Network access control lists (ACL) should be configured according to your framework type
Unused network ACL should be removed
Flow logs should be enabled for all subnets in use
Verify the IAM policies to control access
Classification of data stored in VPC, followed by level of security the data needs

# Amazon Web Services (AWS) Test Cases

## ✓ Cloud Load Balancing ✓

Review the IAM policies used to access the load balancer and API actions
Review if only secure protocols are being used such as HTTPS, TLS (nothing less than 1.2)
Verify user and groups policies
Security of data at rest and in transit through ELB
Review the API configuration to control the ELB panel

## ✓ Information Leakage ✓

Review the configuration (security groups) for accepting clients and replying back to them
Code leakage
Files/code containing passwords, credentials in plain text and/or hard coded

## ✓ Validation of Third Party Components ✓

Old repos, backup files
Code leakage

## ✓ Inter-Security policies of launched Vms, servers etc. ✓

Password complexity, expiry, multi factor authentication etc.
Logging, auditing, alerts on high priority tasks such as login, tampering with system files etc.
Level of security on executables, nx, PIE on linux and Compile flags on Windows OS

## ✓ Relational Database Service ✓

Data backup should be enabled
The backup retention  period should be more than 7 days and according to the security policies
A multi-AZ deployment should be used
Instance storage should be encrypted
The security group should allow access only to specified IP addresses
Create a separate IAM to manage the RDS and rotate the IAM credentials regularly
Database snapshots should not be publicly accessible

## ✓ Connection to AWS ✓

Open ports on the launched instances, servers, etc.
SSH, Web GUI login limit, password complexity, keys, certificates

## ✓ Simple Notification Service ✓

Permission to add, delete, publish, receive, remove topics, set topic attributes, and subscribe should not be granted to all principals
A separate IAM user with programmatic access should be used only for working with SNS service and implement least privilege access

SafeKeep

# Amazon Web Services (AWS) Test Cases

## ✓ CloudTrail ✓

CouldTrail should be turned on and configured correctly, not by default
Global services logging should be enabled
Write access to S3 buckets (if possible dedicated and centralized) with logs should be allowed only for the CloudTrail service
Data events for trails must be enabled