Transaction Analysts

Secure Configuration Baseline Document

Version 1.0

Transaction Analysts India Private Limited

#4, Sathyam Arcade 1st Floor, 1st Phase, BTM Layout,
2nd Stage, Bengaluru, Karnataka 560076

Document Revision History

Date	Version	Description	Created by	Approver
06/06/2025	1.0	Initial Revision (Baselined)	Karmendra Suthar, Director, Omniware Technologies	Vinay Chandrakanth, Director Omniware Technologies

Table of Contents

1 Introduction

This document defines the Secure Configuration Baseline for the Payment Aggregator & Payment Gateway System deployed on Amazon Web Services (AWS) infrastructure across the Mumbai (ap-south-1) and Hyderabad (ap-south-2) regions. The architecture supports a high-availability, scalable, and secure payment processing platform that interfaces with merchants, end customers, and banking partners.

The purpose of this baseline document is to establish a standardized security posture for all AWS resources deployed in this environment. It ensures that configurations align with:

- Industry security best practices (NIST, PCI-DSS v4.0)
- Regulatory and compliance requirements
- Organization's internal security and governance policies

Objectives:

- Minimize attack surfaces through secure configuration and hardening
- Ensure confidentiality, integrity, and availability of payment and customer data
- Provide consistent security controls across production and DR regions
- Establish a foundation for ongoing security auditing and compliance

2 EC2 Nginx Hardening

This Nginx hardening outlines the points taken to secure the Nginx web server within an AWS environment, ensuring that it meets industry best practices for security and compliance.

Biz Server	TAP-PgBiz-One	pgbiz.tapay.in
MRM Server	TAP-PgMRM-One	pgmrm.tapay.in

2.1 TAP-PgBiz-One

Hardening Settings	Remarks
add_header X-Frame-Options "SAMEORIGIN" always;	
#add_header X-XSS-Protection "1; mode=block" always;	
add_header X-Content-Type-Options "nosniff" always;	
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;	

add_header Content-Security-Policy-Report-Only "default-src 'self'; script-src 'self' 'unsafe-inline'; 'unsafe-eval'; style-src 'self' 'unsafe-inline'; object-src 'self' blob: https://*.s3.ap-south-1.amazonaws.com/; base-uri 'self'; connect-src 'self'; font-src 'self' https://fonts.gstatic.com; frame-src 'self' blob: https://www.google.com/ https://*.s3.ap-south-1.amazonaws.com/; img-src 'self' data: https://*.s3.ap-south-1.amazonaws.com/; manifest-src 'self'; media-src 'self'; form-action 'self'; frame-ancestors 'self'; worker-src 'none'; upgrade-insecure-requests; block-all-mixed-content" always;	
add_header Permissions-Policy "microphone=(self), geolocation=(self), camera=(self)" always;	
add_header Referrer-Policy "strict-origin-when-cross-origin" always;	
# Add headers to prevent caching	
add_header Cache-Control "no-cache, no-store";	
add_header Expires 0;	
add_header Pragma "no-cache";	

2.2 TAP-PgBiz-One

Hardening Settings	Remarks
add_header X-Frame-Options "SAMEORIGIN" always;	
#add_header X-XSS-Protection "1; mode=block" always;	
add_header X-Content-Type-Options "nosniff" always;	
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;	
add_header Content-Security-Policy-Report-Only "default-src 'self'; script-src 'self' 'unsafe-inline'; 'unsafe-eval'; style-src 'self' 'unsafe-inline'; object-src 'self' blob: https://*.s3.ap-south-1.amazonaws.com/; base-uri 'self'; connect-src 'self'; font-src 'self' https://fonts.gstatic.com; frame-src 'self' blob: https://www.google.com/ https://*.s3.ap-south-1.amazonaws.com/; img-src 'self' data: https://*.s3.ap-south-1.amazonaws.com/; manifest-src 'self'; media-src 'self'; form-action 'self'; frame-ancestors 'self'; worker-src 'none'; upgrade-insecure-requests; block-all-mixed-content" always;	

add_header Permissions-Policy "microphone=(self), geolocation=(self), camera=(self)" always;	
add_header Referrer-Policy "strict-origin-when-cross-origin" always;	
# Add headers to prevent caching	
add_header Cache-Control "no-cache, no-store";	
add_header Expires 0;	
add_header Pragma "no-cache";	

3 EC2 Configuration

EC2 Instances:

EC2 Instance: Bastion (Jump) server	TAP-Bastion
EC2 Instance: API/Transaction server	TAP-PgBiz-One
EC2 Instance: Batch processing server	TAP-PgBiz-Batch
EC2 Instance: Default/Low priority queue server	TAP-PgBiz-Queue-Default
EC2 Instance: MRM (UI) server Original	TAP-PgMRM-One

Aspect	**Configuration Baseline**
Instance Type	All EC2 instances are provisioned with appropriate instance types based on workload requirements.
Operating System	All instances run secure and supported versions of Ubuntu 20.04 or above.
Security Groups	Security groups are configured to allow only required traffic (e.g., HTTP, HTTPS) from trusted IP ranges.
Recommendation	Review and adjust instance types based on evolving resource needs. Ensure all non-production instances are tested for security compliance.

3.1 Compute Services (EC2 / Application Servers)

Control Area	Configuration Baseline
AMI Selection	Only hardened and approved AMIs.
SSH Access	Key-based access only via Bastion. SSH ports closed from internet.
Monitoring	CloudWatch Agent, Wazuh Agent installed.
Auto Updates	OS security patches managed via SSM Patch Manager.
Anti-Malware	Wazuh/HIDS agent installed and active.

3.2 Access Control and Authentication

Control Area	Configuration Baseline
IAM Roles and Policies	IAM roles assigned to EC2 instances follow the principle of least privilege. Access to AWS services is restricted to necessary services only.
SSH Key Pairs	SSH key pairs are used for authentication. and no passwords are used for SSH access.
Access Logging	SSH access is logged in Authlog saved in AWS CloudWatch for auditing.
Multi-Factor Authentication (MFA)	MFA is enforced for AWS Console access to EC2 management and for IAM roles associated with sensitive instances.
Recommendation	Perform regular reviews of IAM roles and permissions.

3.3 Identity and Access Management (IAM)

Control Area	Configuration Standard
Root User	Root account is not used for day-to-day operations. MFA is enabled.
IAM Users & Groups	Roles are used for service access. User permissions follow least privilege principle.
MFA	Enabled for all admin users, especially Bastion access (2FA).
IAM Roles	Assigned per component (e.g., EC2, Lambda). No sharing across services.
Access Keys	No hardcoded keys. Rotated every 90 days.
Policy Enforcement	Deny-all-by-default and allow only explicitly required actions.

3.4 Encryption

Control Area	**Configuration Baseline**
Encryption in Transit	TLS encryption is enabled for all connections to EC2 instances, and SSL/TLS certificates are regularly updated to connect with outside world.
Recommendation	Regularly review SSL/TLS certificate expiration and implement automated alerts for certificate renewal.

3.5 Network Security

Control Area	**Configuration Baseline**
Virtual Private Cloud (VPC)	All EC2 instances are deployed within a dedicated VPC, ensuring isolation from other network traffic and public access.
Network ACLs	Network ACLs are configured to restrict inbound and outbound traffic for EC2 instances to only trusted IPs and subnets.

Security Groups	Security groups are set up to restrict traffic to necessary ports (e.g., HTTP, HTTPS, SSH) from specific IP ranges or trusted services.
Private IPs for Communication	EC2 instances communicate over private IP addresses within the VPC to avoid exposure to public networks unless necessary. NAT Gateway is used for any external communication from private IPs.
Recommendation	Regularly audit security group and network ACL rules to ensure compliance with the least privilege model.

3.6 Networking and VPC Configuration

Control Area	Configuration Standard
VPC Design	Separate VPCs for production and DR (Hyderabad), with subnet segmentation (App, DB, Public).
NAT Gateway	Used for outbound traffic from private subnets.
Internet Gateway	Attached only to public subnets.
VPC Flow Logs	Enabled for all VPCs, sent to CloudWatch.
Subnets	App, Web, and DB tiers logically separated.
Peering / TGW	Disabled unless necessary. Requires approval and monitoring.

3.7 Security Groups and Network ACLs

Control Area	Configuration Standard
Ingress/Egress Rules	Whitelisted IPs only. Admin access restricted via security group and Bastion.
Open Ports	No port 22 or 3389 open to 0.0.0.0/0.
Application Access	API access allowed via specific IP ranges.
Network ACLs	Explicit deny rules for unused ports and services.
TLS Enforcement	TLS 1.2 or 1.3 enforced on all external communications (Admin UI, Merchant APIs, End Customers).

4 AWS RDS – MySQL

4.1 Introduction

The secure configuration for an Amazon RDS for MySQL database, where:

RDS Instances

RDS Instance 1: prod-oltp-instance	TAP-Oltp-RDS
RDS Instance 2: prod-oltp-replica-instance	TAP-Oltp-Replica-RDS
RDS Instance 3: prod-warehouse-instance	TAP-Warehouse-RDS

4.2 RDS Configuration

Control Area	Configuration Standard
Database Engine Version	Database version: MySQL 8.0
Multi-AZ Deployment	Multi-AZ configuration enabled for prod-oltp-instance, prod-oltp-replica-instance and prod-warehouse-instance for high availability and failover protection

4.3 Access Control and Authentication

Control Area	Configuration Standard
IAM Roles and Policies	IAM roles configured with least privilege for application and backup access to RDS instances.
Database Authentication	Authentication enabled for prod-db-instances, and strong password policies enforced for database users."
User Privileges	Minimum required privileges granted to all database users.

4.4 Encryption

Control Area	Configuration Standard
Encryption at Rest	AWS KMS encryption enabled for data at rest, including EBS volumes and automated backups.
Encryption in Transit	TLS encryption enforced for all connections to RDS instances, and SSL certificates are up to date.

4.5 Network Security

Control Area	Configuration Standard
VPC and Security Groups	RDS instances deployed in a VPC with private subnets; security groups restrict access to trusted IPs and AWS services only.
Network ACLs	Network ACLs configured to control inbound and outbound traffic to RDS instances.

4.6 Backup and Recovery

Control Area	Configuration Standard
Automated Backups	Automated backups enabled with 7-day retention for production and 14-day retention for development instances.
Manual Snapshots	Manual snapshots taken before major changes and updates to databases.

4.7 Monitoring and Logging

Control Area	Configuration Standard
CloudWatch Monitoring	CloudWatch enabled for key RDS metrics (CPU, disk I/O, storage space), with alarms set for critical thresholds.
CloudTrail Logging	CloudTrail logging enabled for API calls related to RDS, and logs stored securely in S3 with a 90-day retention policy.
Enhanced Monitoring	Enhanced Monitoring enabled for prod-db-instance to provide deeper insights into RDS performance.

4.8 Patch Management

Control Area	Configuration Standard
Automatic Minor Version Upgrades	Automatic minor version upgrades enabled for all RDS instances, ensuring timely security patch application.
Patch Validation	Patches validated in a non-production environment before deployment to production instances.

5 SIEM Configuration

5.1 Introduction

Security Information and Event Management (SIEM) system is vital for centralized logging, monitoring, and analysis of security events across an environment. Hardening SIEM solution within PAPG system is an essential step to secure PAPG data and its infrastructure and ensure that the system is protected from potential attacks, complies with security best practices, and operates with minimal vulnerabilities.
This document outlines security hardening steps for implementing SIEM, these hardening steps include configurations, access control, encryption, monitoring, and logging to improve the security of PAPG SIEM solution.

5.2 Security Best Practices for SIEM

1. **Use Multi-Factor Authentication (MFA)**
   - Enable MFA for all user accounts, especially for administrative roles.
   - Use MFA for accessing the AWS Management Console, AWS CLI, and other remote systems.
2. **Principle of Least Privilege (PoLP)**
   - Ensure that users and systems have only the minimum permissions necessary to perform their tasks.
   - Review and adjust IAM roles and policies frequently.
3. **Role-Based Access Control (RBAC)**

- Implement role-based access control to define user roles and assign them the appropriate permissions.
- Define clear roles for security analysts, system administrators, and other users to limit unnecessary access.

4. **Security Group Configuration**
   - Define strict inbound and outbound security group rules.
   - Restrict access to the SIEM system to trusted sources (IP addresses, networks).
5. **Instance Hardening**
   - Secure EC2 instances used for SIEM by disabling unnecessary services.
   - Ensure that the EC2 instances are configured with firewalls, intrusion detection/prevention systems, and logging tools.
6. **Patch Management**
   - Regularly patch all components of the SIEM, including OS, applications, and SIEM software, to reduce vulnerabilities.
7. **Secure API Access**
   - Use API Gateway and IAM roles to manage access to your SIEM solution.
   - Implement rate limiting to prevent brute force attacks and misuse.

5.3 AWS IAM Configuration

1. **Create Least Privilege IAM Roles**
- Assign minimal IAM roles to your SIEM system for AWS resource access.
- Use AWS-managed policies whenever possible to avoid excessive permissions.

2. **Manage Permissions and Policies**
- Use IAM policies that restrict access to only required actions.
- Avoid the use of overly permissive wildcard permissions.

3. **Audit IAM Roles and User Activities**
- Enable AWS CloudTrail to track IAM user activities, including changes to IAM roles and permissions.
- Use AWS IAM Access Analyzer to check for unintended access to resources.

5.4 Data Encryption and Protection

1. **Encryption at Rest**
   - Enable encryption for all logs stored in Amazon S3, Amazon RDS, and any other storage used by your SIEM system.
   - Use **AWS Key Management Service (KMS)** to manage encryption keys.
2. **Encryption in Transit**
   - Ensure all communication to and from your SIEM system uses HTTPS or other secure protocols.
   - Enable TLS/SSL encryption for all API endpoints, log transmission, and data streams.
3. **Encryption for Logs**
   - Encrypt logs both in transit (during transmission) and at rest (stored in S3 or other repositories).
   - Use AWS CloudTrail and CloudWatch Logs with built-in encryption support.
4. **Key Management Service (KMS)**
   - Use **AWS KMS** to manage encryption keys securely.
   - Implement key rotation to regularly update encryption keys.

5.5 Networking and VPC Configuration

1. **VPC Security Best Practices**

- Deploy your SIEM solution within a Virtual Private Cloud (VPC) to isolate it from public internet access.
- Enable VPC Flow Logs to monitor network traffic to/from the SIEM instances.

2. **Use of Security Groups and NACLs**

- Create strict security group and network access control list (NACL) rules for your SIEM resources.
- Allow only trusted IP ranges for management access.

3. **Private and Public Subnet Configuration**

- Place critical components of your SIEM system (like log collectors) in private subnets.
- Use NAT gateways to allow instances in private subnets to access the internet if needed, but ensure no direct internet access.

4. **VPN and Direct Connect Security**

- Use a VPN or AWS Direct Connect for secure communication between on-premises environments and AWS resources.
- Ensure VPN encryption and IPsec configurations are secure.

5.6 Logging and Monitoring

1. **Enable AWS CloudTrail and CloudWatch**

- Enable **AWS CloudTrail** to log API calls made within the AWS environment. This is critical for auditing activities.
- Use **CloudWatch Logs** to capture application and infrastructure logs, and set up CloudWatch Alarms for critical events.

2. **Enable Amazon S3 Access Logs**

- Enable **Amazon S3 Access Logs** to capture access logs of objects within your S3 buckets storing SIEM logs.
- Set up logging at the bucket level to capture detailed information about access requests.

3. **Log Forwarding to SIEM System**

- Ensure that logs from AWS services (e.g., CloudTrail, CloudWatch Logs) are forwarded to your SIEM system for centralized monitoring and analysis.

4. **Secure Log Storage**

- Store logs in an encrypted S3 bucket with limited access. Use lifecycle policies to manage log retention.

5. **AWS Config and CloudWatch Alarms**

- Enable **AWS Config** to monitor the configuration state of your AWS resources.
- Set up **CloudWatch Alarms** to trigger notifications or remediation actions for suspicious activities.

5.7 Incident Response

1. **Define Incident Response Process**

- Develop an incident response plan that includes predefined workflows for security incidents and breaches.
- Train staff on incident handling using AWS tools like CloudWatch, and CloudTrail.

2. **Custom Alerts for Suspicious Activities**

- Create custom alerts for specific patterns of malicious activities such as brute-force attacks, unusual network traffic, and unauthorized access attempts.

5.8 Backup and Recovery

1. **Regular SIEM Backup**

- Ensure that SIEM configurations, logs, and policies are backed up regularly to S3.
- Implement **AWS Backup** to automate backup processes.

2. **Backup Configurations for Log Sources**

- Ensure that all log sources (e.g., EC2 instances, VPCs, security devices) have backup configurations that ensure logs are recoverable.

3. **Testing Recovery Procedures**

- Regularly test your backup and recovery processes to ensure they work in case of a disaster.

5.9 Auditing and Compliance

1. **Regular SIEM Security Audits**

- Regularly audit your SIEM configuration, AWS resources, and IAM policies for security and compliance.

2. **Automate Compliance Checks**

- Use **AWS Config** rules to automatically check for compliance with security best practices and regulatory requirements.

3. **Compliance with Regulatory Frameworks**

- Ensure that your SIEM and AWS environment comply with relevant regulations like PCI-DSS, and RBI by implementing specific configurations and regular audits.