EXCHANGE SECURITY PROTOCOLS

PayBitoPro

Exchange Security Protocols

Application level Security Measures:

1. All private APIs are secure with oauth2 to authorize the request.
2. Users are required to authenticate by username and customizable password security settings with an encryption algorithm and 2-step verification.
3. Data sharing and role-based access specify who can access what data within project management software.
4. The rate limit is integrated with all APIs.
5. Recaptcha is integrated to distinguish between human and automated access to websites.
6. Multi-factor authentication is also implemented at the API level to validate users.
7. Google two-factor authentication is also implemented.
8. Domain restriction is also implemented at the API level to authenticate the request.

Database level Security Measures:

1. Smart Data Protection with Data Guard in Cloud Infrastructure.

   Maintains the standby database by transmitting and applying redo data from the primary database. If the primary database becomes unavailable, then you can use Data Guard to switch or fail over the standby database to the primary role.

2. Using Archive log Mode.

   In ARCHIVELOG mode, the database will make copies of all online redo logs after they are filled. These copies are called archived redo logs.

3. Taking Logical Backup (Expdp) for every day.

   Expdp allows exporting data from the database to another destination. Importing data from the database to another destination is done using impdp. All the files which have been exported can only be imported back using impdp.

4. Taking Physical Backup of RMAN Backup (Incremental) for no Data Loss.

   RMAN can perform backups with minimal effect on the primary database and quickly recover from the loss of individual data files, or the entire database.

5. Flashback Database Mode

   Flashback Database Mode helps rewind the database to a target time, SCN, log sequence number or restore point.

Infra-Level Security Measures

AWS GuardDuty:

AWS GuardDuty is an essential tool that enhances the security of the AWS environment. Enabling GuardDuty gives continuous monitoring and analysis capabilities that help to detect and respond to potential threats. It intelligently analyzes event logs and network traffic data, leveraging machine learning and anomaly detection algorithms to identify unauthorized access, compromised instances, malicious IP addresses, reconnaissance activities, and data exfiltration attempts.

AWS CloudWatch:

AWS CloudWatch is a robust monitoring and observability service to gain valuable insights into the performance, health, and operational state of AWS resources and applications. CloudWatch, can collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in the AWS environment.

AWS CloudTrail:

AWS Cloud Trail is a comprehensive auditing and governance service that allows monitoring and track all API activity across AWS accounts. It provides a detailed history of actions taken by users, roles, and services, allowing me to gain visibility into changes, resource usage, and potential security issues.

AWS VPC Flow logs:

VPC Flow Logs capture detailed information about the network traffic flowing in and out of VPC, allowing us to analyse and monitor the traffic patterns for security, compliance, and troubleshooting purposes.

AWS Secrets Manager:

AWS Secrets Manager is a secure and scalable service that allows storing and managing sensitive credentials such as database passwords, API keys, and other secrets. It eliminates the need to hardcode credentials in applications or configuration files, reducing the risk of exposure.

AWS Macie:

AWS Macie is a powerful security service that helps to discover, classify, and protect sensitive data stored in the AWS environment. Macie uses machine learning algorithms to automatically identify and classify sensitive data such as personally identifiable information (PII), financial data, and intellectual property.

Exchange Security Protocols

AWS WAF & Shield:

AWS WAF (Web Application Firewall) is a managed service that helps protect web applications from common web exploits and security vulnerabilities. WAF allows defining customizable rules to filter and block malicious traffic based on IP addresses, HTTP headers, query strings, and more. It helps to mitigate the risks of common web attacks such as SQL injection, cross-site scripting (XSS) and distributed denial-of-service (DDoS) attacks.

AWS Security Hub:

AWS Security Hub is a comprehensive security service that provides a centralized view of security findings and insights across my AWS accounts. It aggregates and organizes data from various AWS services, including AWS Inspector, Amazon Guard Duty, AWS Macie, and AWS Config, as well as from third-party security tools.

AWS IAM Policies & Roles:

AWS IAM (Identity and Access Management) provides a robust framework for managing user access and permissions within the AWS environment.

AWS EC2 Security Group Rules:

AWS EC2 Security Groups are essential to network security in the AWS environment. They act as virtual firewalls that control inbound and outbound traffic for EC2 instances.

AWS NACL:

AWS Network Access Control Lists (NACLs) are integral to network security in the AWS environment. NACLs act as stateless firewalls that control inbound and outbound traffic at the subnet level.

AWS Private Subnet:

Private subnets allow for the enhancement of the security of backend application servers and databases. By placing these resources in private subnets, we ensure they are not directly accessible from the public internet, reducing the risk of unauthorized access or attacks.

Configuring OpenVPN:

OpenVPN allows creating a private and encrypted network connection, providing a secure tunnel for accessing my instances remotely.

AWS Scheduled Auto Snapshot:

This helps to create consistent backups that can be used for future restores or disaster recovery scenarios.

AWS Auto Backup of Critical Data to S3 Bucket:

It provides an additional layer of protection against data loss, system failures, or other unforeseen events, enabling efficient data recovery and ensuring business continuity.

Server-level Security Measures

Enabling UFW & Iptables:

By combining UFW and iptables, it establishes a robust firewall configuration on the Ubuntu server, protecting it from unauthorized access, network attacks, and other security risks. This ensures that only legitimate and desired network traffic is allowed, while malicious or unwanted traffic is effectively blocked.

Disabling Default SSH port:

It helps to protect against unauthorized access and reduces the risk of automated attacks targeting the default port. It also makes it difficult for unauthorized individuals or automated scripts to find and target the SSH service, thereby reducing the risk of unauthorized access and enhancing the overall security posture of my servers.

Disabling Default Tomcat port:

This helps to protect against unauthorized access and reduces the risk of automated attacks targeting the default port. It makes it difficult for unauthorized individuals or automated scripts to find and target the Tomcat service, thereby reducing the risk of unauthorized access and enhancing the overall security of my Tomcat deployments.

Disabling Default user on Ubuntu server:

To enhance the security of my Ubuntu servers, I disable the default "Ubuntu" user and create a unique random user for each server. This practice helps protect against brute-force attacks and unauthorized access attempts targeting the default username.

By disabling the default "Ubuntu" user and creating a random user for each server, I improve the security of my Ubuntu deployments. This practice mitigates the risk of brute-force attacks and unauthorized login attempts, contributing to a more secure server environment.

Ubuntu Server User Managements & IP-Based SSH authentications:

This involves disabling and enabling users as needed and restricting SSH access based on allowed IP addresses.

By disabling and enabling users as needed and implementing IP-based SSH access control, we can effectively manage user access to the Ubuntu server and restrict SSH connections to specific IP addresses. These measures enhance the security of the server by reducing the attack surface and mitigating the risk of unauthorized access.