Module 2.11 Assignment

NTU_CE7

30 Aug 2024

CE7 Group 2 Members

1. Chua Lai Chwang
2. Dijay Kumar
3. Tan Yuan
4. Lovell Tan
5. Khai Razali
6. Shashipal Singh
7. Wong Teck Choy

What is Cloud Architecture Design - Security?

Chat generated reference

AWS Cloud Architecture Design for security focuses on ensuring that applications and data hosted on AWS are protected from threats and vulnerabilities. Key principles include:

1. **Shared Responsibility Model**: Understand that AWS secures the cloud infrastructure, while customers secure their applications, data, and configurations within the cloud.
2. **Identity and Access Management (IAM)**: Implement fine-grained access controls using IAM roles and policies to manage who can access resources and what actions they can perform.
3. **Data Protection**: Use encryption for data at rest and in transit. AWS provides services like AWS Key Management Service (KMS) for managing encryption keys.
4. **Network Security**: Leverage security groups, network ACLs, and AWS Shield to protect against unauthorized access and DDoS attacks.
5. **Monitoring and Logging**: Enable AWS CloudTrail and Amazon CloudWatch to monitor and log activity for compliance and threat detection.
6. **Incident Response**: Establish procedures and tools to quickly respond to security incidents, such as AWS Config for resource tracking and AWS Security Hub for centralized security management.

By integrating these practices, you can build a robust security posture in your AWS environment.

Cloud Architecture Design - Security Implementation

How do you securely operate your workload?

1. **Separate Workloads Using Accounts:** Using different AWS accounts for separate workloads (e.g., production, development) ensures that each environment is isolated. This limits the blast radius if an account is compromised.
2. **Secure Account Root User:** Protect the root account with strong passwords and enable Multi-Factor Authentication (MFA). Avoid using the root account for daily tasks. Never create access keys for the root account.
3. **Identify and Validate Control Objectives:** Establish a security baseline that includes control objectives, ensuring they align with regulatory and organizational requirements. Use AWS Config and AWS Security Hub to continuously monitor and validate your control objectives against industry standards like CIS or NIST.
4. **Stay Updated with Security Threats:** Subscribe to AWS security bulletins and use services like AWS GuardDuty and AWS Shield to stay informed and protected. Follow industry forums, blogs, and participate in cybersecurity communities to stay aware of emerging threats.
5. **Follow Security Recommendations:** Use the AWS Well-Architected Tool, like the Security Pillar, to regularly assess your workloads against best practices. Regularly review and update your security policies and configurations as AWS services and best practices evolve.
6. **Automate Security Testing and Validation:** Integrate security tests and checks into your deployment pipelines to catch issues early and continuously.
7. **Use a Threat Model:** Use frameworks like STRIDE or DREAD to systematically identify threats to the workload. Based on the threat model, categorize risks and prioritize actions using risk management frameworks like NIST's Risk Management Framework (RMF).
8. **Regularly Evaluate New Security Features:** Continuously assess and adopt new AWS security services and features to enhance your security measures.

How do you manage identities for people and machines?

1. Using AWS Identity Access Management (IAM) to manage privileges to different AWS services and resources.
2. Various types of policies to define permissions for different objects can be set up at various granularity levels, including:
    a. **Identity-Based Policies** - policies attached to IAM identities (users, groups or roles)
    b. **Resource-Based Policies** - policies attached to specific resources like S3 buckets or EC2
    c. **Permissions Boundaries** - used to apply a restriction on the no./types of identity-based policies that can be granted
    d. **Organizations' Service Control Policies (SCPs)** - used to limit permissions that identity-based policies or resource-based policies grant to users/roles within an organization
    e. **Access Control Lists (ACLs) or Bucket Policies** - for S3; to control access of another account to S3 bucket and objects
    f. **Session Policies** - to restrict resource-based or identity-based policies for temporary role/user sessions
3. **Policies tied to IAM identities** - Apply policies to users, groups, departments, or resources, and manage access actively.
4. Securing Access and Authentication:
    a. **AWS Security Token Service (STS)** - Issue temporary credentials for secure access.
    b. **Multi-Factor Authentication (MFA** - Strengthen user authentication with multi-factor verification.
    c. **Single Sign-On (SSO)** - Consolidate access management with Single Sign-On.
    d. **Amazon Cognito** - Provide robust user authentication for apps.

How do you detect and investigate security events?

1. Use **Monitoring and Detection Tools**

- **AWS CloudTrail** - Track API calls and user activities.
- **Amazon CloudWatch** - Monitor resource performance and set alerts.
- **AWS GuardDuty** - Detect malicious activity and unauthorized behavior.
- **AWS Security Hub** - Aggregate and prioritize security findings.
- **Amazon Inspector** - Assess application vulnerabilities.

2. Conduct **Investigation and Analysis**

- **Analyze Logs** - Review CloudTrail and CloudWatch logs for suspicious activity.
- **Security Hub Findings** - Check and investigate alerts.
- **AWS Athena** - Query and analyze logs stored in S3.
- **Network Traffic** - Examine VPC flow logs and network activity.
- **EC2 Instances** - Inspect and analyze compromised instances.

3. Plan **Response and Mitigation**

- **Incident Response Plan** - Follow a predefined plan for handling incidents.
- **Automate Responses** - Use automated tools for quick action.
- **Remediation** - Address and fix identified issues.
- **Post-Incident Review** - Analyze the incident to improve future responses.

How do you protect your data at rest?

To protect data at rest in AWS:

1. **Data Encryption**
Encrypt Data: Use AWS services like S3, EBS, and RDS for encryption. Both AWS-managed and customer-managed keys are supported through AWS Key Management Service (KMS).

2. **Key Management**
Manage Keys: Use AWS KMS for encryption key management or AWS CloudHSM for dedicated hardware security modules (HSMs) for high-security needs.

3. **Access Control**
Control Access: Implement IAM policies, S3 bucket policies, and encryption contexts to restrict access to encrypted data. Ensure least privilege and proper monitoring.

4. **Monitoring & Auditing**
Monitor & Audit: Leverage AWS CloudTrail for API call logs, Amazon Macie for data classification, and AWS Config for configuration tracking and compliance monitoring.

5. **Backup**
Encrypt Backups: Ensure all backups are encrypted with features like EBS snapshots and S3 cross-region replication. AWS Backup can also be used for centralized encrypted backups.

6. **Compliance**
Adhere to Standards: AWS complies with global standards like GDPR and HIPAA, offering resources and tools to help meet regulatory requirements.

7. **Additional Measures**
Data Obfuscation: Use encryption, tokenization, or data masking to protect sensitive data during testing or analysis.
Document Watermarking: Apply visible or invisible marks to documents to protect intellectual property and track document distribution.

This combination of encryption, key management, access controls, and monitoring provides strong protection for data at rest in AWS.

How do you protect your data in transit?

1. **End-to-End Encryption**
   Client-to-Server Encryption: Encrypt data on the client side and decrypt on the server side, ensuring security throughout transmission (e.g., PCI/DSS compliance for online payments).

2. **TLS/SSL Encryption**
   Transport Layer Security (TLS): Encrypt all communications between clients and AWS resources using TLS/SSL to secure data at the transport layer.

3. **Secure File Transfer**
   SFTP (Secure File Transfer Protocol): Use SFTP over SSH for secure file transfers, providing encrypted channels at the application layer.

4. **Secure API Communications**
   OAuth & PKI Tokens: Implement secure API access over HTTPS using OAuth and Public Key Infrastructure (PKI) tokens to encrypt data in API messages.

5. **Network Security Controls**
   Firewalls & NAC: Use firewalls to monitor and control network traffic and Network Access Control (NAC) to restrict access based on identity and compliance.

6. **Additional Security Measures**
   Multi-Factor Authentication (MFA): Ensure only authorized individuals access sensitive data during transmission.
   Data Loss Prevention (DLP): Monitor and control data transfers to prevent unauthorized leakage.
   Regular Audits & Compliance: Conduct regular security audits and maintain compliance (e.g., SOC 2, ISO 27001).
   Employee Education: Train employees on best practices for secure data handling and awareness of threats like phishing.

How do you anticipate, respond to, and recover from incidents?

1. **Anticipate**: Design for high availability, implement security best practices, automate compliance, and create robust backup and disaster recovery plans.
2. **Respond**: Use real-time monitoring tools like CloudWatch, CloudTrail, and GuardDuty. Have an incident response plan in place and automate responses where possible.
3. **Recover**: Conduct root cause analysis, restore services using backups, and update incident response plans based on lessons learned.

AWS References :

- How do you securely operate your workload
- How do you manage authentication for people and machines
- How do you detect and investigate security events
- How do you protect your data at rest
- How do you protect your data in transit
- How do you anticipate, respond to, and recover from incidents