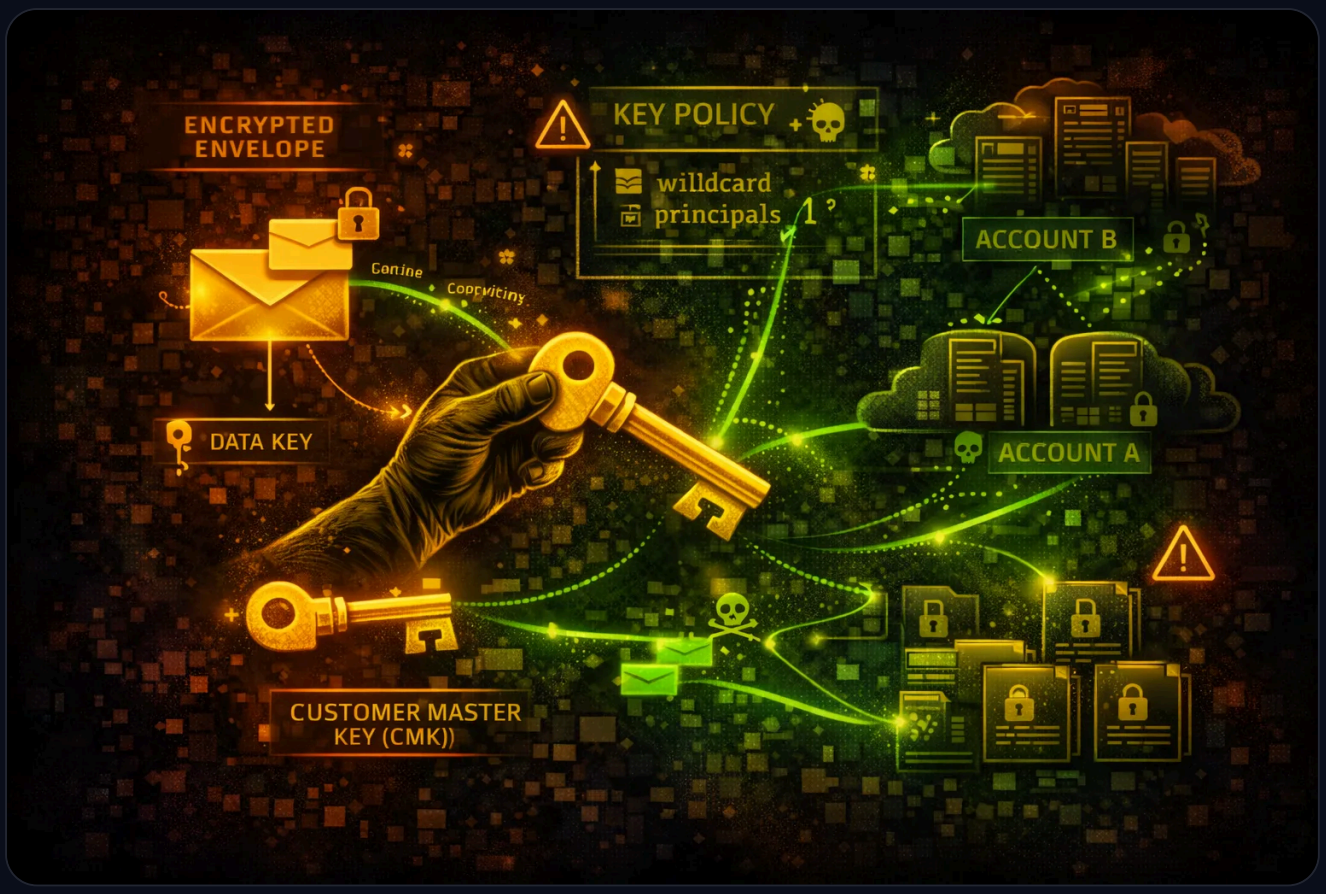

AWS KMS Security

ENCRYPTION

Key Management Service (KMS) manages cryptographic keys for encrypting data across AWS services. Key policies control access. Compromising KMS keys grants access to all encrypted data.

HIGH	Regional	Sym/Asym	CMK/AWS
Risk Level	Scope	Key Types	Management

Service Overview

Customer Managed Keys (CMK)

Keys you create, own, and manage. Full control over key policies, rotation, and deletion. Can be symmetric (AES-256) or asymmetric (RSA, ECC).

Attack note: Key policies often too permissive, allowing privilege escalation through CreateGrant

AWS Managed Keys

Created and managed by AWS services (aws/s3, aws/ebs, etc). Automatic rotation. Limited policy control. Cannot be deleted or shared cross-account.

Attack note: Compromising IAM grants implicit access to AWS managed keys through service permissions

Security Risk Assessment

8.0

Risk Score

KMS key compromise grants access to all encrypted data across S3, EBS, RDS, and other services. Key policies with Principal: * or overly permissive grants are common attack vectors.

⚔️ Attack Vectors

Key Policy Attacks

- Overly permissive key policies
- Principal: * without conditions
- Missing kms:ViaService restrictions
- Cross-account key sharing abuse
- Disabled key rotation exploitation

Grant Abuse

- Create grants for attacker principal
- Retiring grants for persistence
- Grant token theft from metadata
- Cross-account grant creation
- Delegating grant permissions

⚠️ Misconfigurations

Key Policy Issues

- kms:* allowing all actions
- Principal: * in key policy
- Missing MFA conditions
- No kms:ViaService condition
- Cross-account without restrictions

Operational Issues

- Key rotation disabled
- No deletion protection
- Grants without constraints
- Exposed key aliases
- Missing CloudTrail logging

🔍 Enumeration

List All Keys

```
aws kms list-keys --region us-east-1
```

List Key Aliases

```
aws kms list-aliases
```

Get Key Policy

```
aws kms get-key-policy \\
  --key-id KEY_ID \\
  --policy-name default
```

List Grants

```
aws kms list-grants --key-id KEY_ID
```

Describe Key

```
aws kms describe-key --key-id KEY_ID
```

📈 Privilege Escalation

Grant-Based Escalation

- kms:CreateGrant to grant self Decrypt
- kms:PutKeyPolicy to modify policy
- Create retiring grants for persistence
- Leverage existing grants tokens
- Cross-account role assumption

Escalation Paths

- IAM user → CreateGrant → Decrypt all data
- Lambda role → Key access → S3 objects
- EC2 role → EBS volume decryption
- Cross-account → Trust relationship
- Service role → AWS managed key access

Key insight: CreateGrant permission is often overlooked but allows full privilege escalation to Decrypt.

🔗 Persistence

Persistence Mechanisms

- Create retiring grants (survive IAM changes)
- Add backup key administrator
- Enable cross-account access
- Create alias to attacker-controlled key
- Import key material (control encryption)

Ransomware Scenarios

- Schedule key deletion (7-30 day ransom)
- Disable key (immediate data lockout)
- Re-encrypt with attacker key
- Delete all grants (lockout legitimate users)
- Modify key policy (remove access)

Detection

CloudTrail Events

- CreateGrant - grant created
- PutKeyPolicy - policy modified
- ScheduleKeyDeletion - key deletion
- DisableKey - key disabled
- Decrypt - data decryption

Indicators of Compromise

- Unusual CreateGrant from new principals
- Cross-account Decrypt operations
- Key policy modifications
- Bulk decryption events
- GuardDuty KMS findings

💻 Exploitation Commands

Create Grant (Privilege Escalation)

```
aws kms create-grant \\
  --key-id KEY_ID \\
  --grantee-principal arn:aws:iam::ACCOUNT:user/attacker \\
  --operations Decrypt Encrypt
```

Decrypt Data

```
aws kms decrypt \\
  --ciphertext-blob fileb://encrypted.dat \\
  --output text --query Plaintext | base64 -d
```

Schedule Key Deletion (Ransom)

```
aws kms schedule-key-deletion \\
  --key-id KEY_ID \\
  --pending-window-in-days 7
```

Modify Key Policy

```
aws kms put-key-policy \\
  --key-id KEY_ID \\
  --policy-name default \\
  --policy file://malicious-policy.json
```

Create Retiring Grant (Persistence)

```
aws kms create-grant \\
  --key-id KEY_ID \\
  --grantee-principal ATTACKER_ARN \\
  --retiring-principal ATTACKER_ARN \\
  --operations Decrypt
```

Disable Key (DoS)

```
aws kms disable-key --key-id KEY_ID
```

📜 Policy Examples

✗ Dangerous - Overly Permissive

```
{
  "Effect": "Allow",
  "Principal": "*",
  "Action": "kms:*",
  "Resource": "*"
}
```

Anyone can perform any KMS action - full key compromise

✓ Secure - Least Privilege

```
{
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::123456789012:role/AppRole"},
  "Action": ["kms:Decrypt", "kms:GenerateDataKey"],
  "Resource": "*",
  "Condition": {
    "StringEquals": {"kms:ViaService": "s3.us-east-1.amazonaws.com"}
  }
}
```

Only specific role, limited actions, service-restricted

✗ Risky - No ViaService Condition

```
{
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::123456789012:role/AppRole"},
  "Action": ["kms:Decrypt", "kms:Encrypt"],
  "Resource": "*"
}
```

Missing ViaService allows direct KMS API calls

✓ Secure - MFA Required

```
{
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::123456789012:user/admin"},
  "Action": ["kms:ScheduleKeyDeletion", "kms:PutKeyPolicy"],
  "Resource": "*",
  "Condition": {
    "Bool": {"aws:MultiFactorAuthPresent": "true"}
  }
}
```

Sensitive operations require MFA

Defense Recommendations

Least Privilege Key Policies

Restrict kms:* - only allow specific actions needed (Decrypt, GenerateDataKey).

```
"Action": ["kms:Decrypt", "kms:GenerateDataKey"]
```

Require MFA for Admin Actions

Add MFA condition to sensitive operations like key deletion and policy changes.

```
"Condition": {"Bool": {"aws:MultiFactorAuthPresent": "true"}}
```

Enable Key Rotation

Enable automatic annual key rotation for symmetric CMKs.

```
aws kms enable-key-rotation --key-id KEY_ID
```

Deletion Protection

Deny ScheduleKeyDeletion in SCP or key policy to prevent ransomware.

```
"Effect": "Deny", "Action": "kms:ScheduleKeyDeletion"
```

Monitor with CloudTrail

Set up CloudWatch alarms for CreateGrant, PutKeyPolicy, and Decrypt events.

Regular Grant Audits

Regularly review and revoke unnecessary grants.

```
aws kms list-grants --key-id KEY_ID | jq '.Grants[]'
```

Always obtain proper authorization before testing