AWS WAF(WEB Application FIREWALL)

WAF protects application from cyber attacks be defining certain rules.

It contains 4 parts:

Use Case: I live in a place A in Bangalore and want to meet my friend living in place B in Bangalore.

Rule Statement: If traffic is less on the roads? Are there any Uber / OLA available?

Rule Statements define basic characteristics that would be analyzed within a web request.

These can be custom-defined or you can use ready-made ones from AWS and marketplace.

● Block all the requests which are coming from out of India.

● Block request which has a URI Path of /admin

You can even build a custom condition based on: Headers, HTTP Method, Query Strings, URI Path, Geo-Location, Body.

Rules: If traffic is less AND uber ola available then yes or no

We can combine multiple statements into rules to precisely target requests.

WAF provides two primary rule types: Regular Rule & Rate-Based rule

Let’s look into sample regular rule:

● If a request comes from 172.30.0.50

● Request has SQL-like code

Rules with multiple statements can be AND, OR, NOT

Rate-Based rule = Regular Rule + Rate limiting feature

1 If a request comes from 172.30.0.50

2. If requests exceed 1000 request in 10 minutes

WebACL: Container for all the things + default action. Web ACL is a centralized place that contains the rules, rule statements and associated

configuration.

It is used to define the protection strategy.

Association: Zeal

Association defines to which entity WAF is associated to.

WAF cannot be associated with EC2 instances directly.

Support Association: ALB and CloudFront, API Gateway

IAM

Permission Boundaries

ex: permission boundary set for s3 full access, policy attach for administrator access. But since boundary is for s3 only, the user won't have administrator access

**permission boundary just sets the boundary. It does not give access.**

**for access you will have to define permission policy.**

ex: if only permission boundary for s3 full access defined but no policy attached, then will get access denied, as not permission POLICY was attached for access.

The **effective permissions is the inersection (AND) of all the policies**, like Permission boundary, identity based policy and organization SCP ( Service control policies (SCPs) are a type of organization policy that you can use to manage permissions in your organization. SCPs offer central control over the maximum available permissions for all accounts in your organization.)

109. Overview of AWS Organizations

In aws organization we have a main  account which can have 100s of account.

AWS organization offers 2 features:**policies and consolidated billing**

1.When policies is applied via aws organizations (SCP), then even the root user of the accounts under aws org. can not change those permissions.

2. In consolidated billing, a single bill is generated for all the aws accounts under aws org.

**Cryptography Kung Fu and Security**

120. Challenge with storing encryption keys & Intro to

CloudHSM

AWS CloudHSM is a cloud-based hardware security module (HSM) that enables you to easily generate and use your own encryption keys on the AWS Cloud.

AWS uses **Safenet Luna SA HSM**appliance for cloud hsm.

AWS Key Management Service

* after decryption, KMS returns data in Base64 format, so we have to decode that too to see a password as plain text.
* If you enable encryption to an s3 bucket at a later stage, then the objects before that stay in plain text without encryption
* Even if you have server side encryption enable(SSE-S3), but you upload a file using  CLI command for SSE-C type of encryption, i.e. key generated by client. Then the encryption method would be overridden and final method shown would be SSE-C