AWS Web Application Firewall – WAF

# AWS Web Application Firewall – WAF

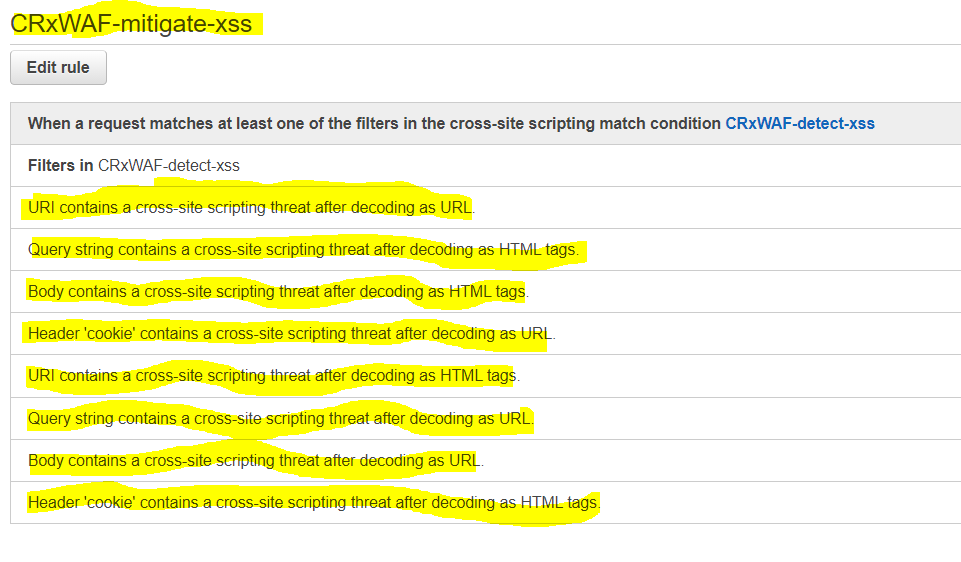
* AWS WAF is a web application firewall that helps protect web applications from attacks by allowing rules configuration that allow, block, or monitor (count) web requests based on defined conditions
* AWS WAF helps protects from common attack techniques like SQL injection and Cross-Site Scripting (XSS), Conditions based include IP addresses, HTTP headers, HTTP body, URI strings
* AWS WAF tightly integrates with Amazon CloudFront and the Application Load Balancer (ALB), services used to deliver content for their websites and applications.
  + AWS WAF with Amazon CloudFront
    - AWS WAF rules run in all AWS Edge Locations, located around the world close to the end users.
    - Blocked requests are stopped before they reach the web servers.
    - Helps support custom origins outside of AWS
  + AWS WAF with Application Load Balancer
    - WAF rules run in region and can be used to protect internet-facing as well as internal load balancers.
  + AWS WAF with API Gateway
    - Can help secure and protect the REST APIs.
* AWS WAF helps protect applications and can inspect web requests transmitted over HTTP or HTTPS.
* AWS WAF provides Managed Rules which are pre-configured rules to protect applications common threats like application vulnerabilities like OWASP, bots, or Common Vulnerabilities and Exposures (CVE).
* AWS WAF allows the following behaviors:
  + **Allow all requests except the ones specified** – blacklisting for e.g all IP addresses except the ones specified
  + **Block all requests except the ones specified** – whitelisting for e.g IP addresses the request originate from
  + **Count the requests that match the specified properties** – allows counting of the requests that match the defined properties, which can be useful when configuring and testing allow or block requests using new properties. After confirming that config did not accidentally block all of the traffic to the website, configuration can be applied to change the behavior to allow or block requests.

## How WAF Works

WAF allows controlling behavior to web requests by creating conditions, rules, and web access control lists (web ACLs).

### Conditions

* Conditions define basic characteristics to watch for in a web request
  + **Malicious script – XSS  (Cross Site Scripting)** – Attackers embed scripts that can exploit vulnerabilities in web applications
  + **IP addresses** or address ranges that requests originate from.
  + Length of specified parts of the request, such as the query string.
  + **Malicious SQL – SQL injection** – Attackers try to extract data from the database by embedding malicious SQL code in a web request
  + **Strings that appear in the request**, for e.g., values that appear in the User-Agent header or text strings that appear in the query string.
  + Some conditions take multiple values.



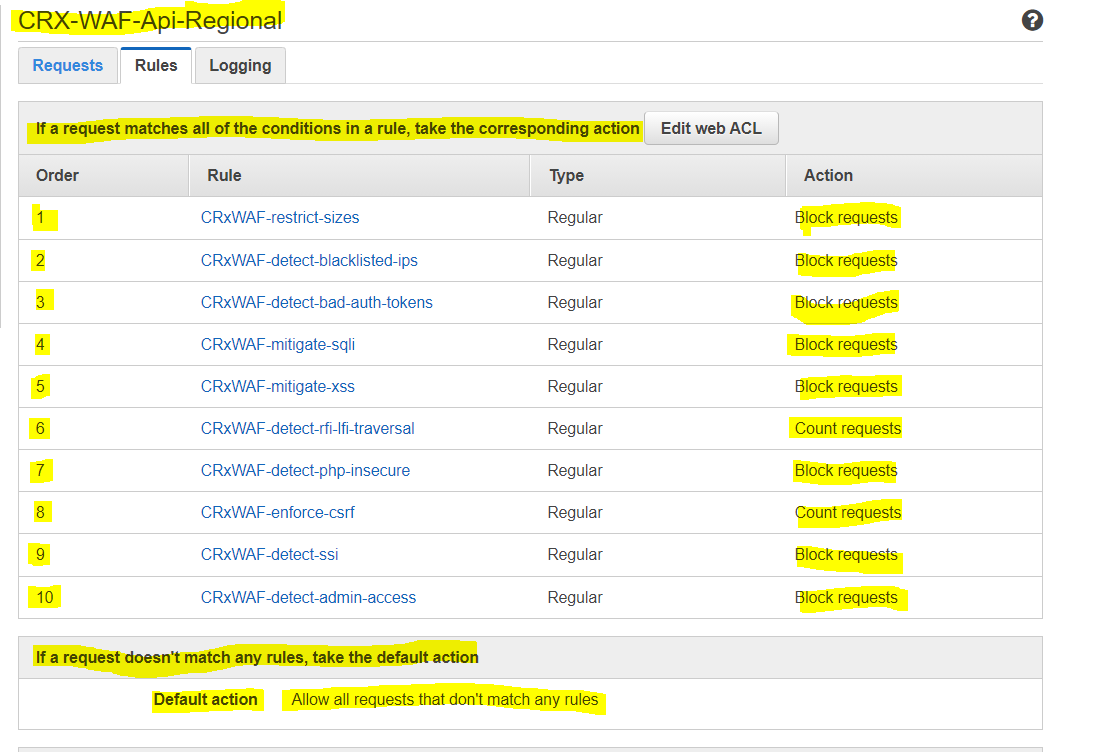
### Rules

* **Rules are basically Combination of Conditions** to precisely target the requests to be allowed or blocked.
* When a rule includes multiple conditions, WAF looks for requests that match all the conditions-it ANDs the conditions together.
* for e.g., based on recent requests that you’ve seen from an attacker, you might create a rule that includes the following conditions:
  + The requests come from 192.0.2.44.
  + They contain the value BadBot in the User-Agent header.
  + They appear to include malicious SQL code in the query string.
* All 3 conditions should be satisfied for the Rule to be passed and the associated action to be taken

### 

### Web ACLs

* Web ACLs provides
  + **Combination of Rules**
  + **Action – allow, block or count**to perform for each rule
    - WAF compares a request with the rules in a web ACL in the order in which its listed and takes the action that is associated with the first rule that the request matches.
    - For multiple rules in a web ACL, WAF evaluates each request against the rules in the order they are listed in the web ACL.
    - When a web request matches all of the conditions in a rule, WAF immediately takes the action—allow or block—and doesn’t evaluate the request against the remaining rules in the web ACL, if any.
  + **Default action**
    - determines whether WAF allows or blocks a request that does not match all of the conditions in any of the rules



## WAF Benefits

* Additional protection against web attacks using specified conditions
* Conditions can be defined by using characteristics of web requests such as the following:
  + IP addresses that the requests originate from
  + Values in request headers
  + Strings that appear in the requests
  + Length of requests
  + Presence of SQL code that is likely to be malicious (this is known as SQL injection)
  + Presence of a script that is likely to be malicious (this is known as cross-site scripting)
* Managed Rules to get you started quickly
* Rules that you can reuse for multiple web applications
* Real-time metrics and sampled web requests
* Automated administration using the WAF API

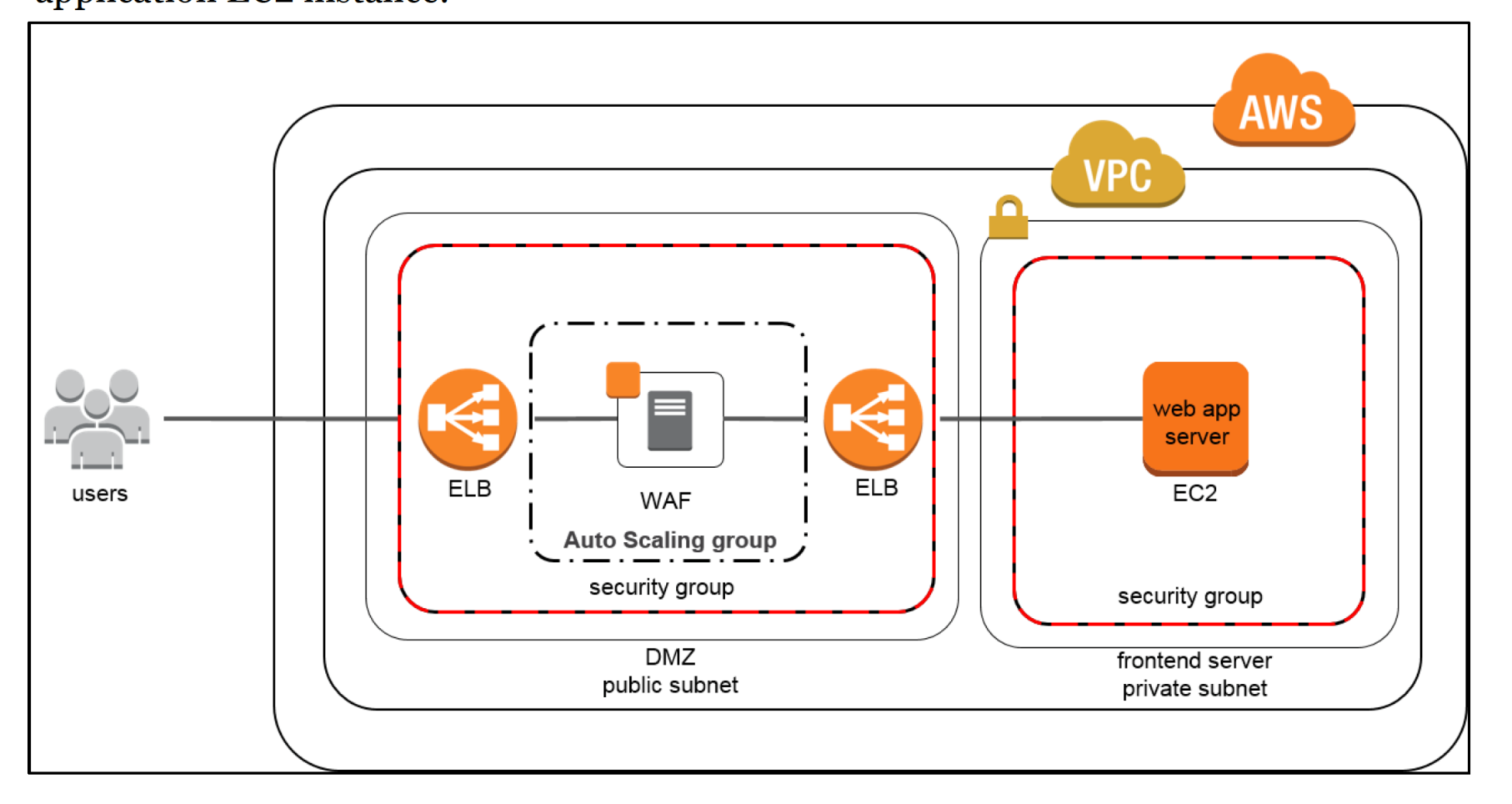
## AWS WAF based Architectures


          Solution Overview: Blocking IP Addresses that Submit Bad Requests
        

1. Above architecture shows AWS WAF integration with CloudFront and Lambda to dynamically update WAF rules
2. CloudFront receives requests on behalf of the web application, it sends access logs to an S3 bucket that contains detailed information about the requests.
3. For every new access log stored in the S3 bucket, a Lambda function is triggered. The Lambda function parses the log files and looks for requests that resulted in error codes 400, 403, 404, and 405.
4. Lambda function then counts the number of bad requests and temporarily stores results in the S3 bucket
5. Lambda function updates AWS WAF rules to block the IP addresses for a period of time that you specify.
6. After this blocking period has expired, AWS WAF allows those IP addresses to access your application again, but continues to monitor the requests from those IP addresses.
7. Lambda function publishes execution metrics in CloudWatch, such as the number of requests analyzed and IP addresses blocked.
8. CloudWatch metrics can be integrated with SNS for notification

### Web Application Firewall Sandwich Architecture

NOTE :- From DDOS Resiliency Whitepaper and doesn’t use the AWS WAF and not valid anymore



* DDoS attacks at the application layer commonly target web applications with lower volumes of traffic compared to infrastructure attacks.
* WAF can be included as part of the infrastructure to mitigate these types of attacks
* WAFs act as filters that apply a set of rules to web traffic, which cover exploits like XSS and SQL injection but can also help build resiliency against DDoS by mitigating HTTP GET or POST floods.
* HTTP works as a request-response protocol between end users and applications where end users request data (GET) and submit data to be processed (POST). GET floods work by requesting the same URL at a high rate or requesting all objects from your application. POST floods work by finding expensive application processes, i.e., logins or database searches, and triggering those process to overwhelm your application.
* WAFs have several features that may prevent these types of attacks from affecting the application availability for e.g. HTTP rate limiting which limits the number of requests per end user within a certain time period. Once the threshold is exceeded, WAFs can block or buffer new requests to ensure other end users have access to the application.
* WAFs can also inspect HTTP requests and identify those that don’t confirm to normal patterns
* In the “**WAF sandwich**,” the EC2 instance running the WAF software (not the AWS WAF) is included in an Auto Scaling group and placed in between two ELB load balancers. Basic load balancer in the default VPC will be the frontend, public facing load balancer that will distribute all incoming traffic to the WAF EC2 instance.
* With WAF sandwich pattern, the instance can scale and add additional WAF EC2 instances should the traffic spike to elevated levels.
* Once the traffic has been inspected and filtered, the WAF EC2 instance forwards traffic to the internal, backend load balancer which then distributes traffic across the application EC2 instance.
* This configuration allows the WAF EC2 instances to scale and meet capacity demands without affecting the availability of your application EC2 instance.

## AWS Certification Exam Practice Questions

1. The Web Application Development team is worried about malicious activity from 200 random IP addresses. Which action will ensure security and scalability from this type of threat?
   1. Use inbound security group rules to block the IP addresses.
   2. Use inbound network ACL rules to block the IP addresses.
   3. **Use AWS WAF to block the IP addresses.**
   4. Write iptables rules on the instance to block the IP addresses.
2. You’ve been hired to enhance the overall security posture for a very large e-commerce site. They have a well architected multi-tier application running in a VPC that uses ELBs in front of both the web and the app tier with static assets served directly from S3. They are using a combination of RDS and DynamoDB for their dynamic data and then archiving nightly into S3 for further processing with EMR. They are concerned because they found questionable log entries and suspect someone is attempting to gain unauthorized access. Which approach provides a cost effective scalable mitigation to this kind of attack? [Old Exam Question]
   1. Recommend mat they lease space at a DirectConnect partner location and establish a 1G DirectConnect connection to their VPC they would then establish Internet connectivity into their space, filter the traffic in hardware Web Application Firewall (WAF). And then pass the traffic through the DirectConnect connection into their application running in their VPC. (Not cost effective)
   2. Add previously identified hostile source IPs as an explicit INBOUND DENY NACL to the web tier subnet. (does not protect against new source)
   3. **Add a WAF tier by creating a new ELB and an AutoScaling group of EC2 Instances running a host-based WAF. They would redirect Route 53 to resolve to the new WAF tier ELB. The WAF tier would then pass the traffic to the current web tier. Web tier Security Groups would be updated to only allow traffic from the WAF tier Security Group**
   4. Remove all but TLS 1.2 from the web tier ELB and enable Advanced Protocol Filtering This will enable the ELB itself to perform WAF functionality. (No advanced protocol filtering in ELB)