How do you secure content delivery with CloudFront?

CloudFront secures content delivery using multiple layers:
🔷 **SSL/TLS** – Ensures all traffic is encrypted (HTTPS). Example: force HTTPS-only access to your app.
🔷 **AWS WAF** – Protects against attacks (SQL injection, XSS, DDoS). Example: block malicious IPs. (Amazon Web Application Firewall)
🔷 **Signed URLs / Signed Cookies** – Restrict access to paid or premium content. Example: OTT app video streaming.

https://d111111abcdef8.cloudfront.net/protected/movie.mp4
?Expires=1725100000
&Key-Pair-Id=K123EXAMPLE
&Signature=Qwerty1234...Base64Signature...

- **Signed URL** — use when you want to allow access to one file (e.g., a single MP4). Simple and safe for one-off downloads/streams.
- **Signed Cookie** — use when a playback session needs many requests (HLS playlist + many TS/segment files). Cookie avoids signing each segment individually.

1. **Protect your origin** (S3, origin server) so only your CDN can fetch it. Use CDN-origin access (S3 OAC/OAI or origin access control).
2. **Create keys**: produce an RSA key pair (private key stays on your backend). Upload the public key to the CDN (CloudFront: Public Key → Key Group → attach to distribution). This tells CDN which keys are trusted to sign URLs.

 User requests a premium video (app/web).
 Your backend verifies the user has paid/has subscription.
 Backend **generates a signed token** (URL or cookie) that includes: resource(s), expiry time, and optionally IP/device restrictions. The token is cryptographically signed with your private key.

 Backend returns the signed URL (or sets signed cookie via response) to the client player.
 Client requests content from the CDN (e.g., CloudFront) using the signed URL or cookie.
 CDN checks signature and expiry (and policy like IP if used). If valid, CDN serves the content; otherwise it returns 403.

🔷 ***origin access control*** **(OAC) or Origin Access Identity (OAI)** – Prevents direct access to S3, so files can only be served via CloudFront. CloudFront provides two ways to send authenticated requests to an Amazon S3 origin: ***origin access control*** **(OAC) a**nd *origin access identity* (OAI). OAC helps you secure your origins, such as Amazon S3.

🔷 **Geo-Restrictions** – Allow/block access based on country. Example: restrict movie streaming to India only.
Example:
You host software downloads in S3 → configure CloudFront with OAI + signed URLs. Users get files only through CloudFront with limited-time access, not directly from S3.
DevOps Pipeline Use-case:
When deploying an app through CI/CD, part of the pipeline applies CloudFront security configs (SSL certs, WAF rules, OAI policies) via Terraform/CloudFormation, ensuring **security is automated and consistent**.

Which CloudFront feature ensures all traffic between users and the distribution is encrypted?
A) AWS WAF
B) SSL/TLS certificates
C) Origin Access Identity (OAI)
D) Signed URLs

Answer: B) SSL/TLS certificates *Explanation: SSL/TLS certificates enable HTTPS encryption for all traffic, ensuring secure communication between users and CloudFront.*

What is the primary purpose of Origin Access Identity (OAI) or OAC in CloudFront?
A) To encrypt data in transit
B) To prevent direct access to S3 buckets, forcing traffic through CloudFront
C) To block malicious IP addresses
D) To restrict access based on geographic location
Answer: B) To prevent direct access to S3 buckets, forcing traffic through CloudFront *Explanation: OAI or OAC ensures that S3 content can only be accessed through CloudFront, not directly from S3, providing better security and control.*

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-restricting-access-to-s3.html

---

Which security feature would you use to provide time-limited access to premium content in an OTT streaming application?
A) Geo-restrictions
B) AWS WAF
C) Signed URLs or Signed Cookies
D) SSL/TLS
Answer: C) Signed URLs or Signed Cookies *Explanation: Signed URLs and Signed Cookies allow you to control access to content with time limitations, perfect for premium or paid content.*

AWS WAF integrated with CloudFront protects against which types of attacks?
A) Only DDoS attacks
B) Only SQL injection attacks
C) SQL injection, XSS, and DDoS attacks
D) Only geographic-based attacks
Answer: C) SQL injection, XSS, and DDoS attacks *Explanation: AWS WAF provides comprehensive web application protection including SQL injection, cross-site scripting (XSS), and DDoS attack mitigation.*

In a DevOps pipeline, which tools are commonly used to automate CloudFront security configurations?
A) Jenkins and Docker only
B) Terraform and CloudFormation
C) Kubernetes and Helm
D) Ansible and Puppet
Answer: B) Terraform and CloudFormation *Explanation: Terraform and CloudFormation are Infrastructure as Code tools that can automate the deployment and configuration of CloudFront security settings.*
