Nadia Arellano González
Actividad Módulo 3- Sesión 3

Caso: Eres el administrador de seguridad de una empresa ficticia llamada SecureCorp, que opera una aplicación web de comercio electrónico. La empresa ha notado un aumento en intentos de acceso no autorizado y ataques de denegación de servicio. La tarea es configurar y proteger la infraestructura de la aplicación para garantizar su disponibilidad y seguridad.

Requerimientos técnicos:

1. **Configuración de Seguridad DNS con Route 53**

   Para proteger el dominio ficticio securecorp.com, lo primero que hice fue ingresar al servicio Route 53 desde la consola de AWS. Allí, seleccioné la opción Hosted Zones y creé una nueva zona hospedada. En el formulario, ingresé el nombre del dominio como securecorp.com, seleccioné el tipo "Public Hosted Zone" y luego hice clic en "Create hosted zone". Esto generó los registros básicos como SOA y NS necesarios para comenzar a administrar los registros DNS del dominio.

   A continuación, habilité DNSSEC para asegurar la integridad de las consultas DNS y prevenir ataques como spoofing. Para ello, dentro de la zona securecorp.com, accedí a la pestaña “DNSSEC signing” y seleccioné la opción para habilitar la firma. Elegí la clave KSK por defecto administrada por AWS, y confirmé la acción. Con esto, el dominio quedó protegido con firmas criptográficas que validan las respuestas DNS.

   Finalmente, creé un registro tipo A que apunta al servidor web principal de la empresa. Para esto, hice clic en “Create record” dentro de la zona hospedada. Ingresé el nombre www para formar www.securecorp.com, seleccioné el tipo de registro como A (IPv4) y en el campo de valor coloqué la IP pública de una instancia EC2 simulada (por ejemplo, 54.218.123.45). Al guardar el registro, el dominio quedó correctamente apuntado al servidor y listo para ser accedido de forma segura.

2. **Protección contra Ataques DDoS y DNS Spoofing**

   Para proteger la aplicación web de SecureCorp frente a ataques de denegación de servicio (DDoS), configuré AWS Shield, una herramienta que ofrece protección automática para los servicios expuestos en internet como CloudFront, Elastic Load Balancer y Route 53. Esta protección se activa automáticamente con Shield Standard, que está incluida sin costo adicional. Con esta configuración, cualquier intento de saturar los recursos de la aplicación será detectado y mitigado sin afectar su disponibilidad.

   Además, configuré restricciones de acceso usando AWS WAF (Web Application Firewall) para limitar el número de solicitudes que una misma IP puede realizar en un periodo corto de tiempo. Dentro del servicio WAF, creé una regla de rate limiting que bloquea temporalmente las IPs que hagan más de 100 solicitudes en 5 minutos. Esto evita que una sola IP genere tráfico excesivo, protegiendo la aplicación de abusos, bots o intentos de ataque de tipo HTTP Flood.

   AWS Shield detecta y mitiga ataques DDoS mediante un sistema de monitoreo continuo que analiza el tráfico entrante en busca de patrones anómalos. Si identifica un comportamiento sospechoso, como un gran volumen de paquetes desde múltiples fuentes, inicia una

mitigación automática para bloquear ese tráfico sin afectar a los usuarios legítimos. Esta defensa se realiza en tiempo real, lo que permite mantener la disponibilidad de la aplicación incluso en medio de un intento de saturación.

3. **Distribución Segura de Contenido con CloudFront**

   Para entregar de forma segura el contenido estático de la aplicación web de SecureCorp, como imágenes, hojas de estilo (CSS) y archivos JavaScript, configuré una distribución en Amazon CloudFront. Desde la consola de AWS, accedí al servicio CloudFront y creé una nueva distribución, especificando como origen un bucket de S3 donde se almacenan los archivos estáticos de la aplicación.

   Durante la configuración, seleccioné la opción de habilitar HTTPS obligatorio, asegurando que todas las solicitudes y respuestas entre los usuarios y los servidores de CloudFront estén cifradas con TLS. Esto garantiza la privacidad de los datos en tránsito y previene ataques del tipo Man-in-the-Middle. También se generó automáticamente un certificado SSL usando AWS Certificate Manager, vinculado al nombre del dominio.

   Finalmente, apliqué políticas de control de acceso para restringir ciertos contenidos solo a usuarios autenticados. Para esto, configuré Signed URLs, lo que permite que solo usuarios con un token válido puedan acceder a archivos privados (como catálogos exclusivos o recursos administrativos). De esta forma, se evita que usuarios no autorizados descarguen o accedan a contenido restringido sin haber iniciado sesión correctamente.

4. **Prevención de Ataques Man-in-the-Middle**

   Para proteger la comunicación entre los usuarios y la aplicación web de SecureCorp contra ataques del tipo Man-in-the-Middle, configuré un certificado SSL/TLS usando el servicio AWS Certificate Manager (ACM). Desde la consola de AWS, accedí a ACM y solicité un certificado para el dominio securecorp.com. Luego de completar el proceso de validación (mediante DNS), el certificado fue emitido con éxito. Posteriormente, lo asocié a la distribución de CloudFront que entrega el contenido web, habilitando el cifrado HTTPS en todas las conexiones.

   Además, activé la política de seguridad HSTS (HTTP Strict Transport Security) dentro de la configuración de CloudFront. Esto se hizo a través de la creación de una política de respuesta personalizada, donde se agregó el encabezado Strict-Transport-Security con el valor recomendado:

   ```
   Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
   ```

   Esta política obliga al navegador del usuario a conectarse siempre por **HTTPS**, incluso si el usuario escribe "http://", evitando así que el tráfico se transmita sin cifrar.

   Gracias a la implementación conjunta de **SSL/TLS** y **HSTS**, la aplicación evita que un atacante pueda interceptar o modificar los datos que viajan entre el usuario y el servidor. El cifrado TLS asegura que el contenido no sea leído ni alterado por terceros, mientras que HSTS impide que el navegador acepte conexiones no cifradas, bloqueando posibles ataques de red que intenten forzar el uso de HTTP inseguro.