Hackeo

Existen diversas formas de hackear, y este tema es bastante interesante. Como un aficionado al hacking ético (White Hat Hacker), encuentro que hay mucho conocimiento por aprender en este campo. Aquí, registraré algunos métodos que he explorado.

Contraseñas Predeterminadas

Algunos sitios web, incluyendo los de agencias gubernamentales, todavía utilizan contraseñas predeterminadas. Aunque muchas empresas o usuarios cambian sus credenciales predeterminadas, otras no lo hacen. Los usuarios suelen ser perezosos, y contraseñas como `12345678` todavía son comunes. Esto es especialmente cierto en sistemas antiguos o de nicho.

nmap o Netcat

Estas herramientas se utilizan para escanear los puertos de un servidor. Presta especial atención a los puertos comúnmente utilizados, como el 80, 22 y 443. Para las instancias de AWS, el nombre de usuario predeterminado es `ec2-user`. Para las instancias de Azure, es `azure-user`. Para las instancias de Google Cloud, el nombre de usuario predeterminado suele ser `ubuntu` o `google-cloud`. Para otras instancias en la nube, generalmente es `root`.

Usando la Consola del Navegador

La consola del navegador es útil para inspeccionar información oculta. A veces, datos críticos están incrustados en el código HTML o JavaScript, pero no son visibles en la página en sí.

Puertas traseras

En la vida, las puertas traseras proporcionan acceso no autorizado a edificios, a menudo sin ser notadas o sin vigilancia, como estacionamientos o puertas laterales. De manera similar, los sistemas pueden tener puertas traseras ocultas que evaden los protocolos de seguridad normales.

Ingeniería Social

Los apodos, cumpleaños y publicaciones en redes sociales de las personas pueden revelar mucha información personal. A menudo, estos detalles se utilizan para crear contraseñas débiles. En el caso de las redes Wi-Fi, conocer el número de casa de alguien u otros detalles identificativos puede ayudar a adivinar su SSID o contraseña.

Inyección SQL

Para cualquier campo de entrada, probar con `? 1=1` es una técnica común para identificar vulnerabilidades y posibles puntos de inyección SQL.

APIs de Actuator o Health

Para servidores de API, aplicaciones como Spring Boot ofrecen un endpoint `/actuator` que proporciona datos sobre el estado de la máquina y la aplicación. Otros frameworks web también tienen funcionalidades similares que pueden exponer detalles sensibles del servidor.

Monitoreo de Tráfico

Para entender cómo el frontend interactúa con el backend, utiliza aplicaciones de proxy como Charles Proxy en macOS para grabar y analizar los registros de solicitudes. Esto puede darte una visión detallada de las rutas y los intercambios de datos entre los componentes.

Límites y Casos Extremos de las APIs

Al trabajar con APIs, es crucial comprender tanto sus límites como los casos extremos que pueden surgir. Estos aspectos son fundamentales para garantizar que tu aplicación maneje adecuadamente todas las situaciones posibles, desde el flujo normal hasta los escenarios inesperados.

Límites de las APIs

1. **Límites de Tasa (Rate Limits):** Muchas APIs imponen límites en la cantidad de solicitudes que puedes hacer en un período de tiempo determinado. Por ejemplo, una API

podría permitir solo 1000 solicitudes por hora. Exceder este límite puede resultar en la suspensión temporal o permanente del acceso.

2. **Límites de Tamaño de Datos:** Algunas APIs tienen restricciones sobre el tamaño de los datos que puedes enviar o recibir en una sola solicitud. Por ejemplo, una API podría limitar el tamaño de un archivo adjunto a 10MB.
3. **Límites de Autenticación:** Las APIs suelen requerir autenticación, y pueden tener límites en cuanto a la cantidad de tokens de acceso que puedes generar o la duración de estos tokens.
4. **Límites de Concurrencia:** Algunas APIs limitan el número de solicitudes simultáneas que puedes hacer. Esto es común en APIs que manejan operaciones intensivas en recursos.

Casos Extremos de las APIs

1. **Respuestas Vacías o Nulas:** A veces, una API puede devolver una respuesta vacía o nula. Es importante manejar estos casos para evitar errores en tu aplicación.
2. **Errores de Red:** Las solicitudes a una API pueden fallar debido a problemas de red. Es esencial implementar un manejo adecuado de errores y reintentos para estos casos.
3. **Cambios en la API:** Las APIs pueden cambiar con el tiempo, ya sea en su estructura de respuesta, en los endpoints disponibles o en los parámetros requeridos. Mantener tu aplicación actualizada con estos cambios es crucial.
4. **Datos Inesperados:** A veces, una API puede devolver datos en un formato inesperado o con valores inusuales. Es importante validar y sanitizar los datos recibidos para evitar problemas.
5. **Límites de Tiempo de Espera:** Las APIs pueden tener un tiempo de espera máximo para las solicitudes. Si una solicitud tarda demasiado, la API podría cancelarla automáticamente.

Ejemplo de Código para Manejar Límites y Casos Extremos

```
import requests
from requests.exceptions import RequestException
```

```python
def fetch_data(api_url):
    try:
        response = requests.get(api_url, timeout=10)
        response.raise_for_status()  # Lanza una excepción para códigos de estado HTTP 4xx/5xx
        data = response.json()

        if not data:
            print("La respuesta está vacía o nula.")
            return None

        return data
    except RequestException as e:
        print(f"Error en la solicitud: {e}")
        return None

# Ejemplo de uso
api_url = "https://api.example.com/data"
data = fetch_data(api_url)

if data:
    print("Datos recibidos:", data)
else:
    print("No se pudieron obtener los datos.")
```

En este ejemplo, se manejan varios casos extremos, como respuestas vacías, errores de red y tiempos de espera. Además, se verifica el código de estado HTTP para asegurarse de que la solicitud fue exitosa.

Conclusión

Comprender y manejar los límites y casos extremos de las APIs es esencial para construir aplicaciones robustas y confiables. Al anticipar estos escenarios y preparar tu aplicación para manejarlos, puedes mejorar significativamente la experiencia del usuario y la estabilidad de tu sistema.

Es importante probar los límites y los casos extremos de una API o servidor. Un ataque de Denegación de Servicio Distribuido (DDoS) intenta superar los límites de solicitudes. Además, los

casos extremos son escenarios en los que las API podrían permitir el acceso a datos restringidos. Probar estos casos puede ayudar a garantizar que los controles de acceso adecuados estén implementados.

Paneles de Administración

A veces, los paneles de administración o internos no están adecuadamente protegidos. Vale la pena intentar acceder a rutas como `/admin` o visitar subdominios como `admin.xx.com` para verificar si estas áreas están correctamente aseguradas.