CWE-400: Uncontrolled Resource Consumption

CWE-400 Consumo de Recursos no Controlado (Agotamiento de recursos)

JUAN ANDRES COLLI CUPUL

Descripción

El software no restringe adecuadamente el tamaño o la cantidad de recursos solicitados o influenciados por un actor, que puede usarse para consumir más recursos de los previstos.

Los recursos limitados incluyen memoria, almacenamiento del sistema de archivos, entradas del grupo de conexiones de bases de datos o CPU.

Descripción

Si un atacante puede activar la asignación de estos recursos limitados, pero el número o el tamaño de los recursos no está controlado, entonces el atacante podría causar una denegación de servicio que consume todos los recursos disponibles.

Esto evitaría que usuarios válidos accedan al software y podría tener un impacto potencial en el entorno. Por ejemplo, un ataque de agotamiento de memoria contra una aplicación podría ralentizar la aplicación y su sistema operativo host.

Consecuencias comunes

Alcance	Impacto
Disponibilidad	Impacto técnico: DoS: bloqueo, salida o reinicio; DoS: consumo de recursos (CPU); DoS: Consumo de recursos (memoria); DoS: Consumo de recursos (otro) El resultado más común del agotamiento de los recursos es la denegación de servicio. El software puede ralentizarse, bloquearse debido a errores no manejados o bloquear usuarios legítimos.
Control de acceso	Impacto técnico: mecanismo de protección de derivación; Otro En algunos casos, es posible obligar al software a "abrirse con error" en caso de agotamiento de los recursos. El estado del software, y posiblemente la funcionalidad de seguridad, puede verse comprometido.

Ejemplo

Este código asigna un socket y bifurcaciones cada vez que recibe una nueva conexión.

```
sock=socket(AF_INET, SOCK_STREAM, 0);
while (1) {
 newsock=accept(sock, ...);
 printf("A connection has been accepted\n");
 pid = fork();
}
```

Ejemplo

El programa no realiza un seguimiento de cuántas conexiones se han realizado, y no limita el número de conexiones.

Debido a que forking es una operación relativamente costosa, un atacante podría hacer que el sistema se quede sin CPU, procesos o memoria al realizar una gran cantidad de conexiones.

Alternativamente, un atacante podría consumir todas las conexiones disponibles, evitando que otros accedan al sistema de forma remota.

Posibles mitigaciones

- Diseñar mecanismos de regulación en la arquitectura del sistema.
- La mejor protección es limitar la cantidad de recursos que un usuario no autorizado puede hacer que se gaste.
- La aplicación de inicio de sesión debe estar protegida contra ataques DoS tanto como sea posible.
- Limitar el acceso a la base de datos, quizás almacenando en caché los conjuntos de resultados, puede ayudar a minimizar los recursos gastados.
- Para limitar aún más el potencial de un ataque DoS, considere rastrear la tasa de solicitudes recibidas de los usuarios y bloquear las solicitudes que excedan un umbral de tasa definido.