Escalada de Privilegios en el Kernel Linux mediante Dirty COW (CVE-2016-5195)

1. Introducción

Dirty COW (Copy-On-Write Dirty) es una vulnerabilidad de condición de carrera en el kernel de Linux, descubierta en octubre de 2016 y asignada al identificador CVE-2016-5195. Este fallo permite a un atacante con acceso local a un sistema explotar una debilidad en el mecanismo de Copy-On-Write para modificar archivos de solo lectura, incluyendo binarios privilegiados, logrando así escalada de privilegios (de usuario normal a root).

2. Contexto Técnico

Mecanismo Copy-On-Write (COW)

- COW es una técnica de optimización en sistemas operativos donde múltiples procesos comparten la misma copia de datos en memoria hasta que uno de ellos intenta modificarla.
- En Linux, esto se aplica a la gestión de memoria y archivos mapeados en memoria (mmap).

Origen de la Vulnerabilidad

El fallo ocurre en la implementación del manejo de páginas de memoria cuando un proceso intenta escribir en una región mapeada como solo lectura. El kernel permite temporalmente que un hilo modifique una página COW mientras otro hilo verifica permisos, creando una condición de carrera que puede ser explotada para sobrescribir archivos protegidos.

3. Explotación de Dirty COW

Condiciones Necesarias

- Acceso a un sistema Linux (local).
- Kernel afectado (versiones anteriores a 4.8.3, 4.7.9, 4.4.26, etc.).
- Permisos de lectura en un archivo objetivo (ej: /etc/passwd o binarios SUID).

Pasos del Ataque

1. Mapear un archivo protegido (ej: /usr/bin/passwd) en memoria con mmap.
2. Forzar una condición de carrera:
    - Un hilo intenta escribir en la página mapeada (modificando su contenido).
    - Otro hilo invoca madvise() para descartar la caché de la página, haciendo que el kernel recargue el contenido desde el disco.
3. Ganar la carrera: Repetir el proceso hasta que se sobrescriba el archivo en disco.

Ejemplo Práctico

Un exploit clásico modifica /etc/passwd para agregar un usuario con UID 0 (root):
sh
.

4. Impacto

- Escalada de privilegios: Un usuario común puede convertirse en root.
- Persistencia: Modificación de binarios críticos o configuraciones.
- Afectación global: Cualquier sistema Linux sin parches (incluyendo Android y dispositivos embebidos).

5. Mitigación y Parches

- Actualizar el kernel a una versión corregida (parches aplicados en 2016).
- Restringir permisos: Limitar el acceso a /proc/self/mem y archivos sensibles.
- Herramientas de detección: Monitorear intentos de escritura en archivos inmutables.

6. Conclusión

Dirty COW es un ejemplo clásico de cómo un error en la gestión de memoria del kernel puede comprometer la seguridad de todo el sistema. Su explotación demostró la importancia de:

- Revisar código del kernel para condiciones de carrera.
- Aplicar parches de seguridad de forma oportuna.
- Implementar mecanismos como SELinux/AppArmor para contener exploits.

Referencias:

- CVE-2016-5195
- Documentación del parche en el kernel Linux
- Ejemplo de exploit en GitHub

Este informe resume el riesgo crítico que supuso Dirty COW y subraya la necesidad de mantener sistemas actualizados ante vulnerabilidades de bajo nivel.