Principios de Seguridad en Sistemas Operativos
Tarea 2 Race Condition Vulnerability

Juan Jose Araya Castro
Alvaro Andrei Miranda Muñoz

August 2022

Código curso: MC1005
Due: Agosto 17, 2022 at 23:59 (CST)
Total puntos: 10 puntos

Introducción

El objetivo de este ataque es aprovechar la vulnerabilidad existente de condición de carrera en la que son ejecutados los procesos utilizando un archivo que posee el máximo nivel de permisos (root) al mismo tiempo ejecutado un archivo que posee permisos de un usuario normal. Esto además se logra mediante el seteo de links simbólicos en el sistema, hoy en día esta vulnerabilidad ya se encuentra cubierta, aún así este método de protección puede ser desactivado de manera manual, si el usuario administrador lo decide y deja esta ventana abierta para los diferentes atacantes.

Instrucciones Generales

Para este ataque es necesario un archivo con permisos de root el cual sera vulp, quien puede ser utilizado por cualquier usuario en el equipo sin importar los permisos de cada usuario, luego debemos ubicarnos en un usuario que no posea los permisos de administrador (root) para ejecutar nuestro script y al mismo tiempo utilizar vulp para intentar escribir en un archivo de acceso exclusivo para root.

1. Deshabilitar la bandera del sistema que bloquea este tipos de ataque con el comando: "sudo sysctl -w fs.protected_symlinks=0"

2. Debemos seleccionar cual archivo debemos atacar en este caso seria Shadow o Passwd. (Nosotros realizamos el ataque a ambos para verificar la funcionalidad)

3. Luego debemos crear un archivo de texto donde guardaremos el string que deseamos inyectar en alguno de los archivos atacados. En nuestro caso son: pass_input y shadow_input

4. Luego de definir el nombre del archivo de texto debemos modificarlo en el archivo "target_1.sh"

5. Dentro del archivo de "pass_input" que definimos ingresamos el valor de "hacker:U6aMy0wojraho:0:0::/home/hacker:/bin/bash"

6. Cuando el ataque tiene éxito, puede ser prácticamente de inmediato o tomar algunos minutos pero este tendrá éxito en este escenario.

7. Ingresamos al usuario inyectado por el ataque y una vez dentro del perfil inyectado dados los parámetros definidos por el ataque este usuario poseerá control total del equipo como un usuario "root"

8. Teniendo control de la máquina con este usuario que se agregó, podemos modificar otros archivos manualmente y ejecutar cualquier optro script o modificar otro archivo que pertenezca a root como el shadow

Descripción del Ataque

Según Wikipedia la condición de carrera es algo que afecta electrónicos, software y otros sistemas que dependen de una secuencia de tiempo sobre eventos incontrolables esto se vuelve un error cuando uno o mas comportamientos se vuelven indeseables.

La idea del ataque es hacerle pensar al equipo que el usuario root con el mayor nivel de permisos en el sistema desea realizar cambios sobre un archivo en específico. Mediante iteraciones infinitas se puede correr un script de manera constante y que solamente este usuario root pueda modificar el archivo cuando realmente es un usuario sin privilegios, esto se logra creando symbolic links que apunten a un archivo de sistema que pertenezca a root.

Lo que se busca es obtener un nivel de permisos de root para modificar cualquier archivo o configuración del equipo, una vez logrado este paso se puede tomar como un ataque satisfactorio.

Documentación del Ataque

Vídeos de las ejecuciones y los ataques:

1. Actividad 1:
   Alvaro Miranda - Juan Jose Araya - Actividad 1.1

Alvaro Miranda - Juan Jose Araya - Actividad 1.2

2. Actividad 2:
   Alvaro Miranda - Juan Jose Araya - Actividad 2.1

   Alvaro Miranda - Juan Jose Araya - Actividad 2.2

   Alvaro Miranda - Juan Jose Araya - Actividad 2.3

3. Actividad 3:
   Alvaro Miranda - Juan Jose Araya - Actividad 3

Autoevaluación

1. Estado Final: Completado
2. Problemas Encontrados: Existencia de una bandera la cual bloquea estos tipos de ataques además de que al recompilar el archivo .c perdía los permisos de root, lo cual nos retrasaba en las pruebas y confirmación de los cambios realizados.
3. Limitaciones Adicionales: Falsos positivos, inicialmente ejecutamos los scripts con un usuario el cual poseía permisos de administrador o root por lo que el ataque se ejecutaba en la primera o segunda ejecución, además para realizar el ataque en las nuevas versiones debemos deshabilitar una bandera de seguridad descubrimos que luego de reiniciar el equipo virtual se reiniciaba la bandera por lo que sufrimos varios días sin saber que dado el reinicio el ataque no funcionaba.
4. Reporte de Commits en Git: Commits History
5. Evaluación: 100%

Rubro	Puntaje Total	Puntaje Obtenido
Actividad 1:	25	25
Actividad 2:	25	25
Actividad 3:	25	25
Documentación del Ataque:	25	25

Lecciones Aprendidas

Este proyecto de Vulnerabilidad de Condición de Carrera nos muestra lo vulnerable que eran los sistemas aun cuando teníamos diferentes capaz de seguridad, como por ejemplo discos encriptados, antivirus, puertos usb deshabilitados etc.

1. En nuestro caso nos fue dado un archivo con código en C llamado vulp, que teníamos que compilar. Parte de la vulnerabilidad radica en ignorar algunos puntos de buenas prácticas que hubieran impedido el ataque.

2. Con respecto al escenario 2 logramos entender de que es necesario comprobar los archivos, su contenido, su integridad, en este caso fue especificamente con los i-nodes, sin embargo existen otras formas de comprobar archivos.

3. En el caso del Least Privilege, entendemos la importancia de impedir que un usuario que no realiza la llamada logre ejecutar una operación, parte de esto representa el core de la vulnerabilidad de condición de carrera.

4. Ningún archivo con acceso de root o información delicada o sensible deberia estar a la mano de un usuario que no tenga permisos de root o administrador, ya que puede sentar las bases para abrir una ventana para un posible ataque.

5. Mantener los sistemas actualizados con las últimas actualizaciones de seguridad del sistema operativo, dado que este tipo de ataques se soluciona con una de las versiones posteriores al SO que utilizamos.

Bibliografía

- Laboratory for Computer Security Education. UTC EDU Race Condition
- Lecture Notes (Syracuse University). Set-UID Privileged Programs
- SEED Labs – Race Condition Vulnerability Lab. Race Condition Vulnerability Lab
- Race Condition Vulnerability Race Condition Vulnerability
- Race condition Race condition