TEC | Tecnológico de Costa Rica

Tarea 3

G-Root

Junio, 2025

Maestría en Ciberseguridad

Principios de Seguridad en Sistemas Operativos

Prof. Kevin Moraga

Objetivo

Implementar un rootkit a nivel de kernel para acercar al estudiante a la interacción con sistema operativo.

Datos Generales

- **Fecha de Entrega:** Viernes 20 de Junio de 2025 antes de las 23:59:59 GMT-6.
- **Fecha de Revisión:** Asíncrona.
- **Lenguaje:** Electivo
- **Recurso Humano:** Grupos de 3
- **Valor de la asignación:** 20 %

Profesor

Kevin Moraga
kmoragas@ic-itcr.ac.cr
Escuela de Computación

Introducción

Un rootkit se define como un conjunto de software que permite al usuario un acceso de "privilegio" a un ordenador, pero mantiene su presencia inicialmente oculta al control de los administradores al descomponer el funcionamiento normal del sistema operativo.

Definición

Introducción

Los rootkits son una gran amenaza para la seguiridad de sistemas operativos, haciendo que el sistema deje de ser confiable y a su vez se oculte de tal manera que es difícil determinar esa pérdida de confianza.

Los rootkits que se alojan en el kernel típicamente pueden modificar u ocultar:

- Archivos
- Procesos
- Conexiones
- Módulos
- Keyloggers

Además los rootkits normalmente se aprovechan de mecanismos como la instalación de drivers o bien módulos del kernels para correr en modo privilegiado.

Ejemplos

A continuación algunos ejemplos de Rootkits:

- Brootus: https://github.com/dsmatter/brootus
- Linux Rootkit: https://github.com/nurupo/rootkit
- Windows Rootkit: https://github.com/memN0ps/eagle-rs/
- Otros: https://gist.github.com/inso-/d9798bd91685ddd00433#file-rootkit-c

Lecturas complementarias

Se recomienda que el estudiante revise las siguientes lecturas:

1. https://xcellerator.github.io/posts/linux_rootkits_01/
2. https://github.com/dsmatter/brootus/raw/master/docs/bROOTus_writeup.pdf
3. https://people.cse.nitc.ac.in/jeena/files/presentation_os_0.pdf

Tareas

Entregable 1: Rootkit - Modificación

1. El estudiante deberá modificar un rootkit de Kernel ya existente.
2. La modificación deberá ser sobre el módulo de ocultamiento de archivos. En lugar de ocultar el archivo se deberá de mostrar el archivo con el nombre de "Oculto". También es posible realizar lo mismo con un proceso, una conexión o un módulo de kernel.
3. Ejecución satisfactoria en un sistema operativo.

Entregable 2: Video

- Se debe crear un video a modo de explicación de la implementación del rootkit en un Sistema Operativo.

Opcional

1. Ejecución de un rootkit en un kernel de linux superior al 5.0
2. Implementación de la creación de un shell reverso.

Aspectos Administrativos

Evaluación

- Entregable 1: 50 %
- Entregable 2: 25 %
- Documentación del Ataque: 25 %
- Extra: 10 %

Documentación

Las siguientes son las instrucciones para la documentación:

1. **Introducción**: Presentar el problema.
2. **Instrucciones para ejecutar el programa**: Presentar las consultas concretas usadas para correr el programa para el problema planteado en el enunciado de la tarea y para los casos planteados al final de esta documentación.
3. **Descripción del Ataque**: En esta sección se debe describir el ataque de Evil Maid en su profundidad y incluyendo referencias hacia variantes del mismo.
4. **Documentación del Ataque**: Este es un resumen de como funciona el ataque Evil Maid ejecutado por el estudiante.

5. **Autoevaluación**: Indicar el estado final en que quedó el programa, problemas encontrados y limitaciones adicionales. Por otro lado, también debe incluir una calificación con la rúbrica de la sección "Evaluación".
6. **Lecciones Aprendidas**: Orientados a un estudiante que curse el presente curso en un futuro.
7. **Video**: Enlace al video de demostración del ataque.
8. **Bibliografía** utilizada en la elaboración de la presente asignación.
9. Es necesario documentar el código fuente.

Aspectos Adicionales

Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones:

1. Sí no se entrega documentación, automáticamente se obtiene una nota de 0.
2. Sí el código no compila se obtendrá una nota de 0, por lo cuál se recomienda realizar la defensa con un código funcional.
3. El código debe ser desarrollado en el lenguaje definido previamente, en caso contrario se obtendrá una nota de 0.
4. Sí no se se firma el archivo de la entrega se obtendrá una nota de 0.
5. La revisión de la documentación será realizada por parte del profesor, no durante la defensa del proyecto.
6. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final de la presente asignación.
7. Durante la revisión podrán participar asistentes, otros profesores y el coordinador del área.
8. Cualquier indicio de copia será calificado con una nota de 0 y será procesado de acuerdo al reglamento.

Licencia