**Proyecto Final**

**Curso de Sistemas Operativos y Laboratorio**

# Análisis Comparativo de Seguridad en Sistemas Operativos

**Miembros del equipo**

Juan David Vásquez Ospina
Juan Diego Calderon Bermeo
Ana María Vega Angarita
Maritza Tabarez Cárdenas

## Resumen

Este proyecto tiene como objetivo evaluar y comparar la seguridad de diferentes sistemas operativos (Windows, Linux, macOS, Android e iOS) frente a ataques cibernéticos comunes. Para ello, se utilizarán entornos virtualizados donde se simularán amenazas. A través del uso de herramientas de pentesting y monitoreo, se observará el comportamiento interno de cada sistema, su capacidad de detección y mitigación de amenazas, y sus vulnerabilidades. Los resultados se documentan en una página web que incluye gráficos comparativos, fortalezas, debilidades y recomendaciones de buenas prácticas de seguridad. El proyecto integra conocimientos clave del curso de Sistemas Operativos, como gestión de procesos, memoria, archivos y control de acceso, aplicados a un enfoque práctico de ciberseguridad.

## Introducción

En el contexto tecnológico actual, el uso de sistemas operativos está presente en prácticamente todos los entornos informáticos: desde servidores en la nube y estaciones de trabajo hasta dispositivos embebidos, móviles e IoT (Internet of Things). A medida que estos sistemas se conectan de forma constante a redes públicas y privadas, aumentan también los vectores de ataque que los ciberdelincuentes pueden aprovechar para comprometer su seguridad.

El desafío seleccionado aborda la necesidad crítica de evaluar y comprender las vulnerabilidades de los sistemas operativos frente a diferentes tipos de ataques. Muchos de estos ataques pueden pasar desapercibidos si los sistemas carecen de mecanismos adecuados de detección, mitigación y registro de incidentes. Esto

representa un riesgo elevado tanto para la confidencialidad como para la integridad y disponibilidad de los sistemas y los datos que gestionan.

Este proyecto es importante porque permite analizar comparativamente el comportamiento de distintos sistemas operativos frente a amenazas reales en un entorno controlado, lo cual proporciona información valiosa para administradores, desarrolladores y responsables de seguridad. Además, fomenta la adopción de buenas prácticas de configuración y selección de sistemas operativos según su nivel de resistencia frente a amenazas específicas. En un momento en que las amenazas evolucionan constantemente y los entornos de TI se vuelven más complejos y distribuidos, la comprensión profunda de las fortalezas y debilidades de los sistemas operativos es fundamental para fortalecer la postura de ciberseguridad de cualquier organización.

**Antecedentes o marco teórico**

Para establecer un fundamento sólido en el desarrollo de este proyecto, es esencial analizar diversos conceptos clave que sustentan el funcionamiento, la arquitectura y la seguridad de estos sistemas operativos. Entre estos conceptos se encuentran:

1. Sistemas operativos: Conjunto de programas que permite manejar la memoria, disco, medios de almacenamiento de información y los diferentes periféricos o recursos de nuestra computadora. [1]
2. Ciberseguridad: Se refiere a todas las tecnologías, prácticas y políticas para prevenir los ciberataques o mitigar su impacto. [2]
3. Pentesting: Conjunto de ataques simulados dirigidos a un sistema informático con una única finalidad: detectar posibles debilidades o vulnerabilidades para que sean corregidas y no puedan ser explotadas. [3]
4. Herramientas de pentesting: Herramientas que facilitan la simulación de un ataque a un sistema software o hardware con el objetivo de encontrar vulnerabilidades para prevenir ataques externos.[4]
5. Virtualización: Proceso que permite una utilización más eficiente del hardware físico de la computadora y es la base de la computación en la nube. [5]
6. Privilegios: Un privilegio es un atributo de proceso que permite que el proceso omita restricciones y limitaciones específicas del sistema.[6]
7. Terminal de comandos: La terminal de comandos es una aplicación con la que podemos interactuar con nuestro sistema operativo a través de texto o bien comandos.[7]
8. Paquete de datos:Estándar que consiste en un conjunto de especificaciones simples pero extensibles para describir conjuntos de datos, archivos de datos y datos tabulares[8]

9. Vulnerabilidad: es una debilidad existente en un sistema que puede ser utilizada por una persona malintencionada para comprometer su seguridad. Las vulnerabilidades pueden ser de varios tipos, pueden ser de tipo hardware, software, procedimentales o humanas y pueden ser explotadas o utilizadas por intrusos o atacantes. [9]
10. Amenaza: En el sentido más sencillo, una amenaza de ciberseguridad es una indicación de que un hacker o actor malicioso está intentando obtener acceso no autorizado a una red para lanzar un ataque cibernético. [10]
11. Mitigación: La mitigación, o mitigación de ataques, es la reducción de la gravedad de un evento. En ciberseguridad, la mitigación se centra en estrategias para limitar el impacto de una amenaza contra los datos bajo custodia. [11]
12. Explotación: Un exploit informático es un tipo de malware que aprovecha errores o vulnerabilidades, utilizadas por los ciberdelincuentes para obtener acceso ilícito a un sistema. [12]

Estos elementos no solo son fundamentales para el correcto desempeño del sistema, sino que también representan puntos críticos que pueden ser explotados por atacantes si no están adecuadamente protegidos.

**Relación con el curso de sistemas operativos**

El análisis de la seguridad en sistemas operativos frente a ciberataques está estrechamente relacionado con los temas clave del curso, ya que los ataques informáticos suelen aprovechar fallas o debilidades en los mismos componentes que se estudian teóricamente y se aplican en laboratorio. A continuación, se detalla esta relación:

**1. Gestión de procesos y planificación**

- **Teoría:** Se estudian los estados de los procesos, la planificación del CPU y la comunicación entre procesos.

- **Relación con la seguridad:** Muchos ataques (como el DoS o la ejecución de código malicioso) afectan directamente el comportamiento de los procesos, por ejemplo, saturando la CPU o inyectando código en procesos legítimos.

**2. Gestión de memoria**

- **Teoría:** Se analizan los esquemas de asignación de memoria, paginación, segmentación y protección.

- **Relación con la seguridad:** Existen ataques como los buffer overflows que explotan errores en la gestión de memoria. Entender cómo el sistema aísla y protege segmentos de memoria es clave para prevenir este tipo de vulnerabilidades.

**3. Sistemas de archivos**

- **Teoría:** Se estudian estructuras de archivos, permisos, y organización jerárquica.

- **Relación con la seguridad:** Los atacantes buscan modificar archivos críticos del sistema, crear puertas traseras o evitar que el sistema registre eventos. Conocer cómo se protege el sistema de archivos es esencial para detectar estas manipulaciones.

**4. Control de acceso y privilegios**

- **Teoría:** Se analizan mecanismos de control de acceso (listas ACL, UID/GID, modos de acceso).

- **Relación con la seguridad:** Los ataques de escalada de privilegios ocurren cuando un atacante obtiene permisos administrativos a partir de una cuenta limitada. Entender cómo el sistema gestiona y restringe los accesos permite evaluar su resistencia a estas amenazas.

**5. Llamadas al sistema y modo kernel/usuario**

- **Teoría:** Se revisa cómo los programas interactúan con el sistema operativo a través de llamadas al sistema y los distintos niveles de privilegio.

- **Relación con la seguridad:** Las llamadas al sistema pueden ser explotadas si el SO no valida correctamente los parámetros. Por ejemplo, los rootkits suelen interceptar syscalls para ocultarse del sistemas

**6. Seguridad del sistema operativo**

- **Teoría:** Aunque puede ser un tema transversal, se aborda la protección frente a amenazas, virus y malware.

- **Relación con el proyecto:** Es el enfoque central. El proyecto busca justamente poner a prueba estas protecciones para evaluar su eficacia ante ataques reales.

**7. Laboratorio del curso**

- En las prácticas de laboratorio se trabajan conceptos como:
    - Manejo de procesos (crear, matar, monitorear).
    - Observación de uso de CPU y RAM.
- Estas habilidades se aplican directamente al realizar pruebas de ataque, monitorear el comportamiento del sistema y detectar signos de vulnerabilidad o compromiso.

**Objetivo General**

Evaluar comparativamente el comportamiento y la seguridad de distintos sistemas operativos frente a ataques cibernéticos comunes, mediante pruebas controladas que permitan identificar vulnerabilidades, mecanismos de defensa y capacidad de respuesta ante amenazas.

**Objetivos Específicos**

1. Seleccionar y configurar entornos virtualizados con diferentes sistemas operativos en un laboratorio de pruebas seguro.
2. Simular ataques de seguridad comunes, como escalada de privilegios, utilizando herramientas especializadas de análisis y pentesting.
3. Observar, registrar y comparar el comportamiento de los sistemas operativos ante cada tipo de ataque, evaluando aspectos como detección, respuesta, impacto y recuperación.
4. Analizar las vulnerabilidades específicas que afectan a cada sistema operativo y los mecanismos de seguridad que implementan.
5. Desarrollar una página web intuitiva y visualmente atractiva que presente los tipos de ataques realizados, los sistemas operativos evaluados, sus fortalezas y debilidades frente a cada ataque, así como el comportamiento interno de los sistemas operativos según el tipo de ataque recibido, y que además proponga recomendaciones de buenas prácticas para mejorar la seguridad de los sistemas evaluados.Elaborar una página web, intuitiva y visualmente agradable que muestre los hallazgos, fortalezas y debilidades observadas, y proponga

recomendaciones de buenas prácticas para mejorar la seguridad de los sistemas evaluados.

**Metodología**

La metodología Kanban será el marco de trabajo utilizado para organizar y optimizar el flujo de tareas a lo largo del desarrollo del proyecto, permitiendo gestionar de manera más eficiente las tareas pendientes, las que están en proceso y las completadas, asegurando que no haya sobrecarga y que el trabajo fluya de manera continua. El tablero Kanban será la herramienta principal para visualizar el progreso de las tareas, dividido en las columnas Pendiente (tareas que aún no han comenzado), En Proceso (tareas que están siendo trabajadas actualmente) y Finalizado (tareas completadas), moviendo las tareas de una columna a la siguiente según avancen, lo que permitirá un seguimiento claro y visual del estado del proyecto. Se establecerán límites en el número de tareas que pueden estar en la columna En Proceso al mismo tiempo, evitando la sobrecarga del equipo y garantizando que las tareas se finalicen de manera eficiente antes de comenzar nuevas actividades. Se realizarán reuniones periódicas diarias o semanales para analizar el avance de las tareas, identificar posibles cuellos de botella y hacer ajustes en el flujo de trabajo, así como para priorizar tareas según los objetivos del proyecto. Finalmente, se llevará a cabo un proceso continuo de ajustes en el flujo de trabajo, mejorando la eficiencia del proceso mediante la retroalimentación del equipo y la asignación de tareas y tiempos de desarrollo.

**Principales herramientas que podrían utilizarse para implementar la solución**

| Categoría | Herramientas / Tecnologías Sugeridas | Propósito |
|---|---|---|
| Virtualización / Laboratorio | VirtualBox, VMware Workstation | Crear entornos de prueba controlados |
| Sistemas Operativos | Windows(versiones),Linux(distribuciones), macOS, Android e IOS | Plataformas a analizar |
| Pentesting / Ataques | Metasploit, Hydra, stress-ng, exploit-db, Netcat, Nmap | Simular ataques y explotar vulnerabilidades |

| | | |
|---|---|---|
| Monitoreo y análisis | Wireshark, htop, dmesg, journalctl, Sysinternals (Windows) | Observar comportamiento, monitorear procesos, analizar logs |
| Detección de rootkits | chkrootkit, rkhunter | Verificar persistencia y manipulación del sistema |
| Control de integridad | AIDE, Tripwire | Detectar cambios en archivos críticos |

**2. Actividades necesarias para cumplir los objetivos**

A continuación, se detallan las actividades en orden lógico:

**Fase 1: Preparación del entorno**

1. **Definir los sistemas operativos a evaluar**
   (Windows(versiones),Linux(distribuciones), macOS, Android e IOS).

2. **Configurar el entorno de laboratorio virtualizado**, asegurando aislamiento total de la red y almacenamiento de snapshots.

3. **Instalar y documentar la configuración base** de cada SO (servicios activos, puertos abiertos, usuarios, etc.).

**Fase 2: Ejecución de pruebas**

4. **Realizar ataques controlados** para cada tipo de amenaza:
   - Escalada de privilegios
   - Rootkits
   - Ataques de denegación de servicio (DoS)
   - Inyección de código o manipulación de procesos
   - Ataques a servicios de red (ej. SSH, SMB)

5. **Monitorear y registrar el comportamiento del sistema**:
    - Logs del sistema
    - Cambios en procesos y uso de recursos
    - Persistencia del ataque (¿el sistema se recupera o sigue comprometido?)

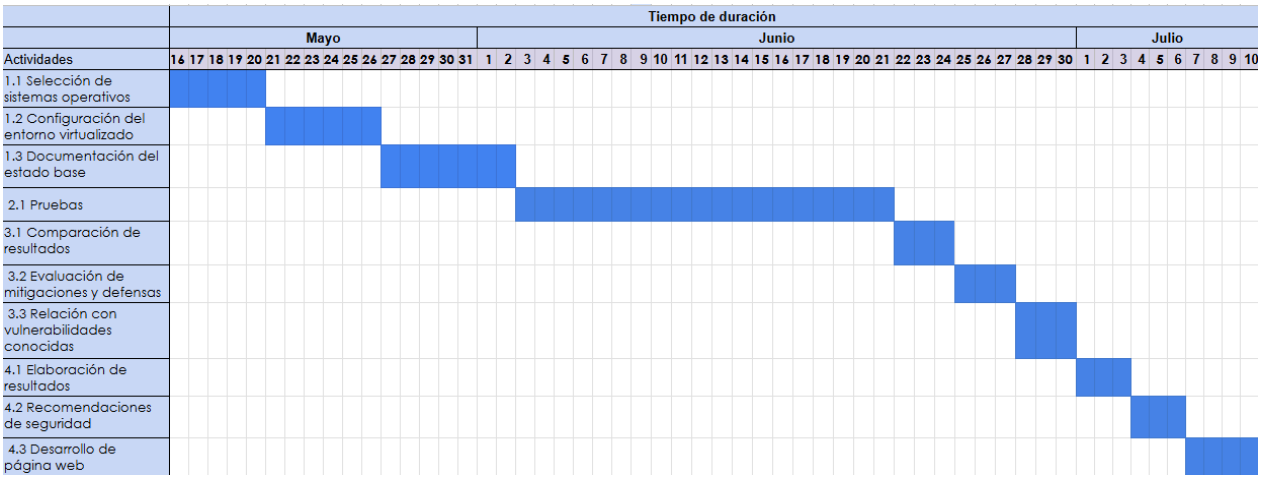
**Fase 3: Análisis y comparación**

6. **Comparar la respuesta de cada sistema operativo** ante los mismos ataques, evaluando:
    - Tiempo de detección o reacción
    - Gravedad del compromiso
    - Dificultad para ejecutar el ataque
    - Herramientas de mitigación activas
7. **Analizar vulnerabilidades detectadas** y correlacionarlas con CVEs o configuraciones por defecto.

**Fase 4: Conclusiones y documentación**

8. **Elaborar tablas y gráficos comparativos** con los resultados obtenidos.
9. **Elaborar la página web**, incluyendo:
    - Fortalezas y debilidades de cada SO
    - Recomendaciones de seguridad
    - Buenas prácticas de configuración y mitigación

## Cronograma




## Referencias


[1]https://desarrollarinclusion.cilsa.org/tecnologia-inclusiva/que-es-un-sistema-operativo/#:~:text=Un%20sistema%20operativo%20es%20un,placa%20de%20red%2C%20entre%20otros.

[2] https://www.ibm.com/es-es/topics/cybersecurity

[3] https://www.incibe.es/empresas/blog/el-pentesting-auditando-seguridad-tus-sistemas

[5] https://www.ibm.com/mx-es/topics/virtualization

[4]https://www.incibe.es/aprendeciberseguridad/pentesting#:~:text=El%20Concepto,vulnerabilidades%20para%20prevenir%20ataques%20externos.

[6]https://www.ibm.com/docs/es/aix/7.3.0?topic=rbac-privileges

[7]https://holamundo.io/2023/05/03/que-es-una-terminal-de-comandos-y-cual-utilizar-para-programar/

[8]https://datapackage.org/#:~:text=Data%20Package%20is%20a%20standard,reusability%20(FAIR)%20of%20data.

[9] https://www.bancosantander.es/glosario/vulnerabilidad-informatica#:~:text=Cross%20Site%20Scripting:%20se%20basa,que%20roba%20as%C3%AD%20sus%20datos.

[10] https://www.ibm.com/mx-es/think/topics/cyberthreats-types#:~:text=IBM%20Cloud%20Team&text=En%20el%20sentido%20m%C3%A1s%20sencillo,para%20lanzar%20un%20ataque%20cibern%C3%A9tico.
[11]https://www.hypr.com/security-encyclopedia/mitigation#:~:text=En%20ciberseguridad%2C%20la%20mitigaci%C3%B3n%20se,la%20venganza%20o%20la%20malicia.
[12] https://www.malwarebytes.com/es/exploits