Reporte técnico: Proyecto final de Sistemas Operativos y Laboratorio

1. Información del Proyecto

- **Título del Proyecto:** Análisis Comparativo de Seguridad en Sistemas Operativos
- **Curso/Materia:** Sistemas Operativos
- **Integrantes:**

Ana María Vega Angarita ana.vega@udea.edu.co
Maritza Tabarez Cárdenas maritza.tabarezc@udea.edu.co
Juan Diego Calderon Bermeo juand.calderon@udea.edu.co
Juan David Vásquez Ospina juan.vasquez21@udea.edu.co

Fecha de Entrega: 16 de julio de 2025

2. Introducción

2.1. Objetivo del Proyecto

Este proyecto tiene como objetivo evaluar y comparar la seguridad de distintos sistemas operativos (Android, macOS, Windows y Linux) frente a ataques cibernéticos comunes. A través de pruebas controladas en entornos virtualizados, se busca identificar vulnerabilidades, mecanismos de defensa, y capacidades de respuesta de cada sistema operativo ante amenazas reales, utilizando herramientas de pentesting y monitoreo. Los resultados obtenidos se presentan de forma visual y técnica en una página web que incluye recomendaciones y buenas prácticas de seguridad.

2.2. Motivación y Justificación

La creciente dependencia de sistemas operativos en contextos personales, empresariales y críticos los convierte en objetivos frecuentes de atacantes. Evaluar sus capacidades de defensa no sólo es relevante, sino esencial para mejorar su configuración y resistencia. Este proyecto es importante para aplicar los conocimientos adquiridos en el curso de Sistemas Operativos y ponerlos en práctica en un escenario realista de ciberseguridad, permitiendo explorar vulnerabilidades y estrategias de mitigación.

2.3. Alcance del Proyecto

El proyecto se enfoca en:

- Simulación de ataques controlados a SOs virtualizados.
- Evaluación de escalada de privilegios, rootkits, denegación de servicio, inyección de código y análisis de servicios.
- Comparación de resultados y propuesta de mitigaciones.

Quedan fuera del alcance:

- Ataques a firmware o hardware.
- Ingeniería social y ataques a redes físicas.
- Análisis en producción o redes reales.

3. Marco Teórico / Conceptos Fundamentales

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.

4. Diseño e Implementación

4.1. Diseño de la Solución

El diseño se basó en una arquitectura de laboratorio con máquinas virtuales aisladas por red, configuradas en VirtualBox. Se tomaron snapshots base por sistema operativo para poder restaurar el estado inicial tras cada prueba. Las pruebas se automatizaron en algunos casos mediante scripts en Bash o PowerShell.

Además, se utilizó una máquina Kali Linux como plataforma de análisis y ataque, desde la cual se realizaron escaneos de puertos, simulaciones de ataques de red (como inyecciones de código o intentos de acceso remoto) y monitoreo del comportamiento del sistema objetivo.

4.2. Tecnologías y Herramientas

Tecnología / Herramienta	Uso
VirtualBox	Virtualización de los sistemas operativos
Kali Linux	Sistema atacante, herramientas de pentesting
ADB (Android Debug Bridge)	Acceso e interacción con Android
Windows Defender / Event Viewer	Monitoreo de alertas y registros en Windows
Process Hacker / System Informer	Observación de procesos críticos en Windows
Extreme Injector	Inyección de DLL en procesos activos (Windows)
WES-NG	Identificación de vulnerabilidades por parches faltantes

	(Windows)
log show (macOS)	Revisión de logs del sistema operativo
BusyBox (Android)	Acceso a herramientas Unix básicas
Shell Scripts	Automatización de pruebas de DoS y persistencia
gcc (MinGW / MSYS2)	Compilación de DLL personalizada para inyección
launchctl (MacOS)	Para gestionar servicios de inicio
lldb (MacOS)	Herramienta de depuración para intentos de inyección de código.
top y Activity Monitor (MacOS)	Para visualizar procesos, uso de CPU y memoria.
arp-scan	Descubrimiento de dispositivos en la red mediante ARP
Nmap	Escaneo de puertos, detección de servicios y sistemas operativos
Metasploit	Framework para explotación de vulnerabilidades
Hashcat	Cracking de contraseñas mediante fuerza bruta o ataques por diccionario
MySQL	Base de datos, objetivo común en pruebas de inyección SQL
Meterpreter	Shell avanzada de Metasploit para post-explotación
Dirb	Fuerza bruta para descubrimiento de directorios web
WPScan	Escaneo de vulnerabilidades en sitios WordPress
John the Ripper	Cracking de contraseñas locales
FTP Client	Acceso a servidores FTP para exploración y transferencia
Telnet	Conexión a servicios remotos sin cifrado
VNC Viewer	Acceso remoto a escritorios gráficos
hping3	Generación de tráfico personalizado para pruebas de red y DoS

4.3. Detalles de Implementación

- Cada sistema se configuró con servicios mínimos activos y se creó una cuenta estándar sin privilegios administrativos para realizar las pruebas.
- Se tomaron snapshots del sistema tras la instalación base para permitir la reversión y repetición de pruebas de forma controlada. Las pruebas se realizaron sin credenciales

de administrador, simulando el accionar de un atacante con acceso limitado. Solo se utilizaron privilegios elevados para la recolección de logs o verificación de efectos post-ataque desde un usuario con permisos.
- Se evitó el uso de credenciales de administrador, salvo en los sistemas que ya estaban comprometidos por diseño (ej. Metasploitable).

5. Pruebas y Evaluación

5.1. Metodología de Pruebas

Para cada sistema operativo se aplicaron las siguientes pruebas:

- Escalada de privilegios desde usuario estándar
- Instalación y ejecución de rootkits
- Inyección de código (DLL o scripts)
- Ataques de denegación de servicio (DoS)
- Monitoreo de logs del sistema y alertas de seguridad

5.2. Casos de Prueba y Resultados

Presente los casos de prueba más importantes que ejecutó y los resultados obtenidos. Puede usar para ello una tabla como la siguiente:

ID	Descripción	Resultado Esperado	Resultado Obtenido	Éxito/ Fallo
CP-001	Escalada de privilegios en Android	Acceso root sin restricciones	su otorgó acceso root	Éxito
CP-002	Rootkit en Windows con r77	Ocultar procesos del Administrador	Proceso oculto tras permitir en Defender	Éxito
CP-003	Inyección de DLL en notepad.exe	Mostrar mensaje y/o crear archivo txt	Archivo dll_inyectada.txt creado correctamente	Éxito
CP-004	DoS en macOS con yes	Aumento uso CPU, sistema lento	Saturación moderada, sistema responde lento	Éxito
CP-005	Rootkit persistente en macOS con plist	Ejecución automática al iniciar sesión	Ejecutado desde LaunchAgents	Éxito
CP-00	Escalada de privilegios	Fallo del intento,	Intento bloqueado,	Éxito

6	en Windows con runas	UAC activa	evento 4625 en logs	
CP-007	WES-NG sobre Windows 10	Detectar parches faltantes	Se identificaron múltiples CVEs	Éxito
CP-008	Acceso a archivos críticos en Android	Lectura exitosa de archivos de sistema	Acceso a /data/system/... sin restricción	Éxito
CP-009	Explotación de backdoor en FTP vsftpd 2.3.4	Shell remota como root	Shell root obtenida con Metasploit	Éxito
CP-010	Acceso a escritorio remoto VNC sin autenticación	Acceso denegado o autenticación solicitada	Acceso completo sin autenticación	Éxito
CP-011	Explotación de PostgreSQL mal configurado	Acceso limitado a la BD	Sesión Meterpreter abierta	Éxito
CP-012	Fuerza bruta a Apache Tomcat con credenciales por defecto	Acceso denegado	Shell interactiva obtenida	Éxito
CP-013	Acceso vía Telnet con credenciales débiles	Acceso remoto no permitido	Inicio de sesión exitoso	Éxito
CP-014	Ejecución de exploit Drupalgeddon2	Acceso al sistema vía shell	Sesión Meterpreter establecida	Éxito
CP-015	Extracción de credenciales de settings.php	Obtención parcial o denegada	Usuario y contraseña revelados	Éxito
CP-016	Acceso a MySQL con credenciales extraídas	Acceso exitoso	Conexión establecida y base listada	Éxito
CP-017	Ataque de diccionario con Hashcat a hashes de Drupal	Crackeo parcial	Contraseña de admin descubierta	Éxito
CP-018	WPScan para detectar vulnerabilidades y usuarios	Detección parcial	Vulnerabilidades y usuarios detectados	Éxito
CP-019	Fuerza bruta contra usuario admin con RockYou	Acceso denegado	Acceso exitoso con admin:admin	Éxito
CP-020	Subida de shell con Metasploit vía	Shell remota obtenida	Sesión Meterpreter activa	Éxito

	WordPress			
CP-021	Acceso a /etc/shadow y extracción de hashes	Acceso permitido desde shell	Hashes obtenidos y crackeados	Éxito
CP-022	Escaneo con Nmap a red local	Identificación de hosts y puertos	Hosts identificados correctamente	Éxito
CP-023	Simulación de DDoS con hping3 contra un puerto abierto	Degradación del servicio	Saturación del servicio en el puerto 80	Éxito

5.3. Evaluación del Rendimiento

Durante las pruebas de DoS, el uso de CPU se elevó por encima del 90% en todos los casos. En Windows y Android, el sistema se volvió inestable con múltiples procesos activos. En macOS, se mantuvo cierto control del sistema aún bajo estrés.

5.4. Problemas Encontrados y Soluciones

- **Antivirus bloqueando pruebas en Windows**: Solucionado desactivando temporalmente Defender para pruebas controladas.
- **Problemas de compatibilidad con Extreme Injector**: Ejecutado siempre con privilegios de administrador.
- **ADB no reconocía Android x86 al inicio**: Solucionado activando “Depuración USB” y puente de red en VirtualBox.
- **MacOS Catalina lento en VirtualBox**: Se optimizó asignando más RAM y recursos de CPU.

6. Conclusiones

El proyecto cumplió con éxito su objetivo principal: analizar, ejecutar y comparar técnicas de ataque y defensa en diferentes sistemas operativos. Se identificaron vulnerabilidades críticas en sistemas como Android y versiones antiguas de Linux (Metasploitable), mientras que otros como Windows y macOS demostraron mecanismos de defensa robustos, aunque no infalibles.

A través de este trabajo, se afianzaron conocimientos en:

- Gestión de usuarios y privilegios
- Administración de procesos y servicios del sistema
- Herramientas reales de ciberseguridad
- Monitoreo de logs y respuestas a incidentes

Este trabajo evidenció la importancia de mantener los sistemas actualizados, utilizar cuentas con privilegios mínimos y monitorear de forma continua la actividad del sistema para detectar comportamientos sospechosos.

7. Trabajo Futuro

- Evaluar la efectividad de antivirus y firewalls frente a exploits en tiempo real.
- Automatizar pruebas con Ansible, PowerShell o scripts Bash.
- Incluir pruebas con sistemas embebidos como Raspberry Pi OS.
- Comparar diferentes versiones del mismo sistema operativo (ej. Android 12 vs 14, windows 10 vs 11, MacOS Catalina vs MacOS Sonoma).
- Realizar pruebas en un entorno híbrido (cloud + local) para simular entornos empresariales reales.

8. 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

9. Anexos

Incluya cualquier material adicional que sea relevante pero que no encaje en el cuerpo principal del reporte, como:

- GitHub: https://github.com/MaritzaTC/operating_system_security.git
- Página desplegada: https://operating-system-security.vercel.app/general