Buffer Overflow

Huber Steven Arroyave Rojas
Assandry Enrique Barón Rodríguez

Sistemas Operativos
Universidad de Antioquia
2025

Introducción

- Un buffer overflow ocurre cuando se escriben datos más allá del límite asignado en memoria.
- Puede permitir ejecución de código malicioso, corrupción de datos o caída del sistema.
- Sigue siendo relevante por la existencia de software heredado y errores en nuevos desarrollos.

Problema y justificación

- **Problema:** Fallos en el manejo de memoria comprometen la seguridad del sistema.
- **Justificación:** Comprender y demostrar esta vulnerabilidad ayuda a prevenir los ataques reales.
- Relación directa con conceptos clave del curso de Sistemas Operativos.

Marco teórico

- Tipos de overflow:
  - Stack Overflow
  - Heap-Based Overflow
  - String Format Vulnerability
- Registros clave: EIP (puntero de instrucción), ESP (puntero de pila)
- Técnicas de explotación: NOP sled, shellcode, control de EIP

Objetivos

Objetivo Principal:

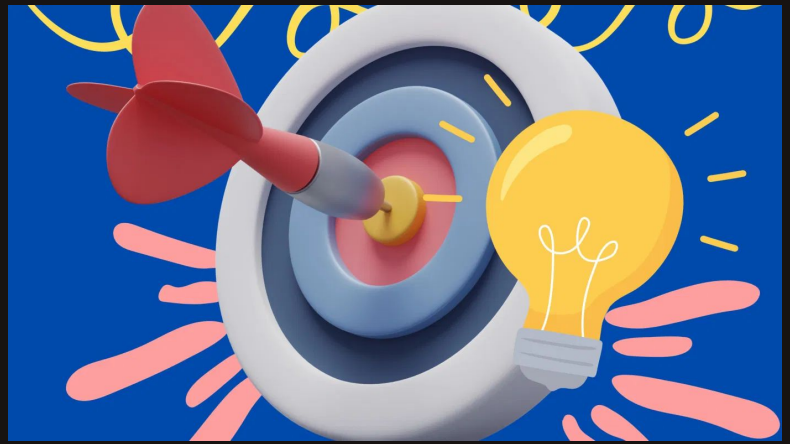
Realizar un ataque de desbordamiento de buffer en un entorno controlado para comprender su funcionamiento y aprender a prevenirlo.

Objetivos Específicos:

- Investigar los fundamentos teoricos y tecnicos del Buffer Overflow.
- Simular el ataque en un entorno controlado (Kali Linux, Immunity Debugger).

Objetivos

- Identificar direcciones de memoria y offsets clave para el ataque.
- Crear y ejecutar un código malicioso (payload).
- Analizar y explorar soluciones de mitigación, evaluando su efectividad.



Metodología

Investigación y revisión:

- Análisis de literatura sobre Buffer Overflow y sus variantes.

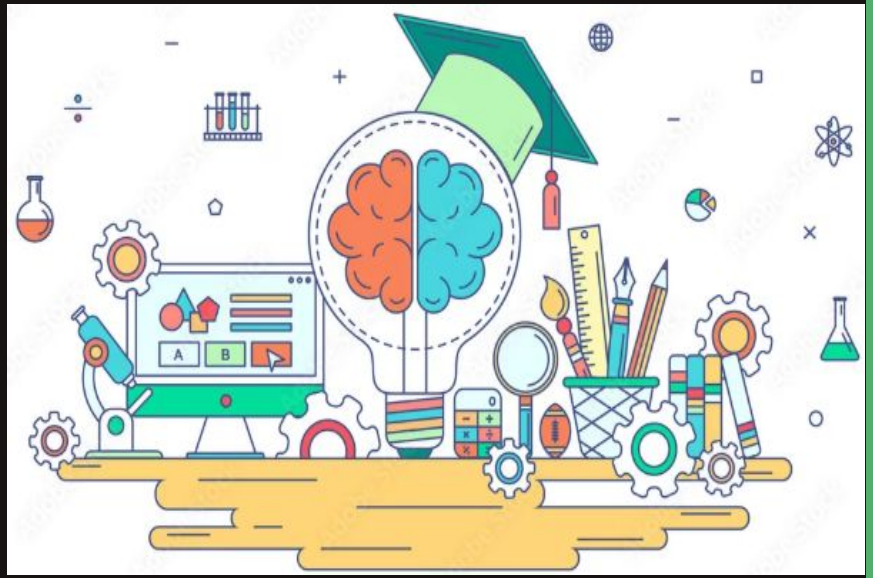
Preparación del Entorno:

- Configuración de VirtualBox con Kali Linux, Windows y Brainpan.
- Compilación de código vulnerable (si aplica, con protecciones deshabilitadas).

Metodología

Analisis de Explotación:

- Uso de GBD, Immunity Debugger, mona.py y scripts de Metasploit.
- Identificación de direcciones de memoria, offsets y badchars.
- Desarrollo y ejecución del payload (Shellcode).
-



Metodología

Documentación y Mitigación:

- Registro de resultados y observaciones.
- Análisis de soluciones de mitigación y mejores prácticas.

Cronograma

ID	Tarea	Duración	Fecha de Inicio	Fecha de Fin
1	Fase 1: Investigación y Configuración			
1.1	Revisión Bibliográfica y Marco Teórico	7	20/05/2025	26/05/2025
1.2	Preparación del Entorno Virtual (VirtualBox, OS)	4	27/05/2025	30/05/2025
1.3	Instalación y Configuración de Herramientas de Ataque	5	31/05/2025	04/06/2025
2	Fase 2: Análisis y Explotación del Binario Vulnerable			
2.1	Fuzzing y Detección de Crash	6	05/06/2025	10/06/2025
2.2	Depuración con Immunity Debugger/GDB	7	11/06/2025	17/06/2025
2.3	Identificación de Offsets y Badchars	4	18/06/2025	21/06/2025
2.4	Generación de Shellcode (msfvenom)	3	22/06/2025	24/06/2025
2.5	Desarrollo del Exploit Script (Python)	5	25/06/2025	29/06/2025
2.6	Ejecución y Validación del Exploit	4	30/06/2025	03/07/2025
3	Fase 3: Análisis de Mitigaciones y Documentación			
3.1	Investigación y Análisis de Técnicas de Mitigación	2	04/07/2025	05/07/2025
3.2	Pruebas con Mitigaciones Activas (si aplica)	1	06/07/2025	06/07/2025
3.3	Redacción de Resultados y Conclusiones	2	07/07/2025	08/07/2025
3.4	Revisión y Edición Final del Documento	2	09/07/2025	10/07/2025

Resultados esperados

- Ejecución exitosa de un ataque de Buffer Overflow en un entorno controlado
- Identificación clara de los offsets, los badchars y la dirección de salto
- Desarrollo y prueba de un exploit funcional en Python

Conclusión

El Buffer Overflow es una vulnerabilidad crítica y explotable. Donde, a pesar de que se ha tratado de solucionar dicha problemática, todavía sigue siendo una preocupación en cuanto a la seguridad informática.

Referencias

- Cloudflare. (s.f.). *Buffer overflow*. Cloudflare. https://www.cloudflare.com/es-es/learning/security/threats/buffer-overflow/
- Foster, J. , Osipov, V., Bhalla, N., Heinen, N., & Liu, Y. (2005). *Buffer overflow attacks: Detect, exploit, prevent*. Syngress Publishing. https://repo.zenk-security.com/Techniques%20d.attaques%20%20.%20%20Failles/Buffer%20Overflow%20Attacks%20-%20Detect%20Exploit%20Prevent.pdf
- Rapid7. (s.f.). *Metasploit Framework*. Rapid7. https://docs.rapid7.com/metasploit/
- Immunity Inc. (s.f.). *Immunity Debugger*. https://www.immunityinc.com/products/debugger/
- GNU Project. (s.f.). *GDB: The GNU Project Debugger*. Sourceware. https://www.gnu.org/software/gdb/