Proyecto de Aplicación No. 01

A lo largo de la historia, han existido vulnerabilidades en los lenguajes de programación, el solo hecho de ingresar un número entero demasiado grande podía ser un problema.

Típicamente ya es un problema que la aplicación deje de funcionar, pero esta clase de situaciones puede ir más allá de solamente detener una aplicación. Muchos ataques a aplicaciones están basados en *exploits*, pero la mayoría solo los utiliza. Para utilizarlos es necesario conocer sobre arquitectura de computadoras, instrucciones del procesador y la pila.

Cuando una aplicación inicia, se crea un proceso y con éste, viene una asignación de memoria, existe una reserva de memoria por proceso.

Una aplicación, por lo menos, tiene los segmentos de código y pila, además del de datos. El segmento de código, almacena las instrucciones del proceso y es de acceso solo lectura; en el segmento de datos, se almacenan las variables; y en la pila, también se encuentra el Heap, además de guardar las variables locales y de control.

Cuando se ejecuta una aplicación y cada vez que se ejecuta una subrutina, se almacena todo lo que estaba en ejecución, en la pila (incluyendo el Instruction Pointer), se ejecuta la subrutina y se recupera el estado anterior de la pila, haciendo operaciones "pop".

Si se logra llenar la pila, hasta donde está el IP, éste puede reemplazarse para apuntar hacia otra instrucción, lo que permite ejecutar código a partir de una aplicación vulnerable. Esto es realmente la base de un *exploit*.

Aplicaciones programadas en C, por ejemplo, son vulnerables, dado que C no comprueba los límites de memoria.

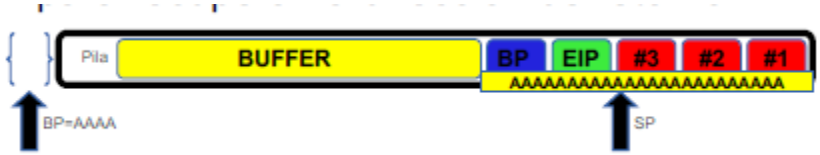
Enunciado

Se le pide a usted como ingeniero, demostrar la vulnerabilidad de una aplicación que corre sobre Windows XP. Los pasos a seguir son:

- Crear un fichero con 1,000 letras "A" (41 en hexadecimal) o alguna otra letra que se desee, con el objetivo de evaluar si es suficiente para desbordar la pila. Posteriormente intentar con 5,000, 10,000, 20,000 y 30,000 letras en el archivo. Documentar los resultados y comentarlos.
- Utilizando un *debugger*, verificar en todo momento qué hay en el Instruction Pointer, en algún momento debe mostrarse 414141.
- El siguiente objetivo será encontrar el desplazamiento exacto dentro de la pila que llegue a sobrescribir el IP. Debe encontrarse un rango y realizar pruebas. Para ello, puede generarse un patrón de elementos irrepetibles. Este patrón puede ser generado con las herramientas de *Metasploit pattern_create.rb*.
- Obtener la dirección exacta donde se encuentra el IP (tomar en cuenta que la notación es Little Endian).

  42356A42 (Little Endian) → 426A3542

- Encontrar el desplazamiento, para ello puede utilizarse la herramienta de *metasploit pattern_offset.rb*.
  Gráficamente se vería:



  Nota: el SP puede que no apunte exactamente a la primera posición después de IP, así que debe comprobarse la posición exacta.

- Finalmente colocar la dirección de memoria a la que apunta el SP en el IP para que se ejecuten las instrucciones codificadas en estas posiciones de memoria.

Requisitos:

- Windows XP (preferiblemente sin *Service Pack*)
- *Immunity Debugger*
- Aplicación Vulnerable: *Easy RM to MP3 Converter* 2.7.3.700 (https://easy-rm-to-mp3-converter.en.softonic.com/)
- *Metasploit Tools*
- *Script* de referencia para hacer pruebas: https://www.exploit-db.com/exploits/40714/

Referencias y Guía:

- Exploit writing tutorial part 1: Stack Based Overflows https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/
- Exploit writing tutorial part 2: Stack Based Overflows – jumping to shellcode https://www.corelan.be/index.php/2009/07/23/writing-buffer-overflow-exploits-aquick-and-basic-tutorial-part-2/
- Understanding buffer overflow exploitation http://www.youtube.com/watch?v=KAr2cjLPufA
- Explotación de Vulnerabilidades mediante Buffer Overflow en Windows. Parte 1 http://www.youtube.com/watch?v=wmnr2KhArZk
- Explotación de Vulnerabilidades mediante Buffer Overflow en Windows. Parte 2 http://www.youtube.com/watch?v=F2p9fD6YiKI
- Explotación de Vulnerabilidades mediante Buffer Overflow en Windows. Parte 3 http://www.youtube.com/watch?v=7vyWVYKoiBE

Entregables:

- El proyecto se entregará vía el Portal Académico, en el espacio especificado.
- Deberá entregarse un documento que contenga capturas de pantalla que contengan el resumen de los pasos que se han seguido y los resultados obtenidos.
- Adjuntar un breve análisis del proyecto, explicando los procesos para realizar los cálculos respectivos.
- La demostración será de forma presencial el día de calificación.