Resolución de la Bomba Binaria

Basso, Ignacio Carlos — Universidad de San Andrés
ibasso@udesa.edu.ar

April 16, 2025

Objetivo

Documentar el análisis y resolución de una bomba binaria utilizando ingeniería inversa sobre código ensamblador y herramientas como GDB.

1 Fase 1

Objetivo: Ingresar una cadena que coincida exactamente con una almacenada en memoria, para evitar detonar la bomba.

Análisis: La función `strings_not_equal` compara la entrada con una constante en `0x4c9a58`. Si no coinciden, se detona la bomba.

Resolución: Usando `x/s 0x4c9a58`, se recupera la cadena requerida:

`‘‘Al que es amigo, jamas lo dejen en la estacada Siempre el amigo mas fiel es una conduta honrada”`.

2 Fase 2

Objetivo: Ingresar dos enteros que satisfagan condiciones lógicas, para lograr desactivar el explosivo.

Análisis: A partir de x e y (almacenados en rbx y eax), se computa $z = x + y - 32$. Luego, siendo pasado a la función misterio, que verifica:

- `popcount(z)` $= 11$
- $z \oplus x < 0$

Resolución: Usando $z = 2047$ (11 bits en 1) y $x = -100$ (para que el XOR con z sea negativo), se obtiene $y = 2179$.

Resultado: `-100 2179`

3 Fase 3

Objetivo: Ingresar una palabra y un número que cumplan múltiples restricciones.

Análisis: Se espera una entrada con formato `"%s %d"`, sabiéndose de haber inspeccionado la instrucción:

```
402044: lea rsi,[rip+0xc504e]  -->  dirección: 0x4c7099
```

. Además:

- La función `cuenta` debe devolver un valor > 9999.
- Ese valor debe coincidir con el número ingresado.

Resolución: Se prueba con la palabra `abachar`, que retorna `10767`, al observar el valor acumulado en `ebx` tras ejecutarse la función cuenta. Este valor cumple con todos los requisitos.
Resultado: `abachar 10767`

4 Fase 4

Objetivo: Ingresar una cadena de 6 caracteres tal que la suma de ciertos valores indexados dé 44.
Análisis:

- Se verifica que el largo del input sea 6.
- Cada carácter se reduce a sus 4 bits bajos, al aplicar `AND 0x0F`, generando un índice a un arreglo:

  `[2, 13, 7, 14, 5, 10, 6, 15, 1, 12, 3, 4, 11, 8, 16, 9]`
- Se acumulan los valores y se compara con 44.

Resolución: Se desarrolló un script para generar combinaciones de 6 caracteres cuyos índices sumaran 44. Se identificó una combinación válida que no activa la bomba. La cual es **`000119`**.

5 Fase Oculta

Acceso: Fue descubierta inspeccionando posiciones de memoria en el binario, donde se halló una cadena clave que permitía desbloquear esta fase oculta al ingresarla correctamente. Teniendo que insertar en la fase 3: `abachar 10767 abrete_sesamo`.
Objetivo: Ingresar un número que, al ser pasado como argumento a la función `fun7`, provoque que esta retorne el valor 120.
Análisis:

- La función `fun7` opera sobre un árbol binario.
- La búsqueda se codifica como un recorrido binario, en el que moverse a la izquierda multiplica por 2 y a la derecha multiplica por 2 y suma 1.
- El valor deseado es $120 = 0b1111000$, lo que indica un camino de 4 pasos a la derecha y 3 a la izquierda.

Lógica de la función:

```
def fun7(node, val):
    if node == NULL:
        return -1
    if val < node.value:
        return 2 * fun7(node.left, val)
    elif val == node.value:
        return 0
    else:
        return 2 * fun7(node.right, val) + 1
```

Resolución: Se utilizó `gdb` para recorrer manualmente el árbol binario desde la raíz, ubicada en `0x4f91f0`, siguiendo el patrón binario del número 120:

- 4 pasos hacia la derecha: $36 \rightarrow 50 \rightarrow 107 \rightarrow 1001$.
- 3 pasos hacia la izquierda desde 1001.

Al intentar avanzar una cuarta vez hacia la izquierda, se llegó a un puntero nulo, confirmando que se habían realizado exactamente 7 pasos válidos según el recorrido binario.

Resultado: Se concluye que el valor que hace que `fun7` retorne 120 es:

$$\boxed{1001}$$

Conclusión

Mediante la interpretación cuidadosa del código ensamblador, combinada con herramientas como GDB y scripting auxiliar, se resolvieron todas las fases de la bomba, incluida la fase oculta. Este análisis demuestra la importancia de la comprensión del bajo nivel y la lógica computacional para tareas de ingeniería inversa.