TPC8 e Guião Laboratorial

Dicas para a execução do trabalho

Buffer overflow

Este problema cobre uma gama variada de tópicos: *stack frames*, representação de *strings*, código ASCII, e ordenação de *bytes*. Mostra ainda os perigos de referenciar a memória fora de limites previstos (*out-of-bounds memory references*), e as ideias básicas relativas a *buffer overflow.* As dicas aqui apresentadas referem-se a uma versão antiga do `gcc`, mas aplicam-se de igual modo a uma execução do trabalho com uma versão mais recente: no código gerado, os 3 registos são salvaguardados antes de se reservar espaço para o vector `buf`.

c) **(A/R)** Estado da *stack frame* na linha 7 (a *stack* cresce para cima):

```
+-------------+
| 00 00 00 02 | Valor guardado de %ebx
+-------------+
| 00 00 00 01 | Valor guardado de %esi
+-------------+
|             |
+-------------+
|             | Reservado para *result
+-------------+
|             | Reservado para buf[0-3]
+-------------+
|             | Reservado para buf[4-7]
+-------------+
| bf ff fc 94 | Valor guardado de %ebp <-- %ebp
+-------------+
| 08 04 86 43 | Endereço de retorno
+-------------+
```

d) **(R)** Estado da *stack frame* depois da linha 10 (mostrando apenas as palavras que foram alteradas). De notar que a função `gets` limita-se a ler uma linha do *standard input* até que encontra o caracter `newline` ou uma condição de erro, terminando a seguir a escrita da *string* com o caracter `null`; neste caso, vai ler os 12 caracteres da *string* e acrescentar o `null`; mas como apenas tem reservado para a *string* um *array* de 8 elementos... veja-se o resultado (na sessão laboratorial seria preciso introduzir 24 valores para ter o mesmo efeito):

```
+-------------+
| 33 32 31 30 | buf[0-3]
+-------------+
| 37 36 35 34 | buf[4-7]
+-------------+
| 31 30 39 38 | Valor guardado de %ebp <-- %ebp
+-------------+
| 08 04 86 00 | Endereço de retorno
+-------------+
```

e) **(R)** Este programa está a tentar regressar ao endereço `0x08048600`, uma vez que o *byte* menos significativo foi modificado (*overwritten*) pelo caracter terminador (*null character*).

f) **(R)** O valor guardado de `%ebp` foi modificado para `0x31303938`, e este valor será o "recuperado" para `%ebp` antes do regresso de `getline`. Os valores guardados dos outros registos não são afectados (nesta resolução com versão antiga de `gcc`), uma vez que eles estão guardados na *stack* em endereços mais baixos que `buf`. O mesmo já não se passou na sessão laboratorial, onde todos os registos salvaguardados foram alterados.

g) **(B)** A chamada de `malloc` deveria ter como argumento `strlen(buf)+1`, e deveria também verificar que o valor de retorno é *non-null*.